Real-World Applications of Genetic Programming in Financial Trading

Luke Burdette
burde041@morris.umn.edu
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA

Abstract

Genetic programming is a branch of artificial intelligence that has been used since the 1990's for financial analysis. As computing has evolved, so has its usage in financial trading, with an estimated 50% of trading volume in US equity markets now involving high frequency computer trading. This has led to changes being required in the models that best predict the future values of stocks to secure the best returns for investors. This paper explains how the usage of genetic programming in finance has evolved over time, as well as analyzing three recent applications of genetic programming to develop models for predicting return of investments, and their results.

Keywords: financial datasets, strongly typed genetic programming, stock market, financial trading, syntax tree, genetic programming, return forecasting

1 Introduction

In recent years, the applications of artificial intelligence (AI) in many different fields has been rapidly expanding due to the discovery of benefits over other mainstream computing solutions. In the case of finance, trading algorithms may be as simple as sending a buy order when a stock's 50-day moving average passes its 200-day moving average, with an estimated greater than 50% of trades today used computers. Genetic Programming (GP) is a branch of AI used to generate functions that evolve over time using a given set of data and a goal metric to measure effectiveness. Ideally, over time the model will improve and can be used on data outside of the given set. The term evolution in this context is derived from how it is similar to the biological process, with the original functions being referred to as parents, and the new generated functions called children. GP has existed in theory for over 50 years, yet it was only in the late 1980's when it became a reality [1]. At first, it was only used for optimizing financial models, but as its potential was recognized, GP was additionally applied to portfolio optimisation and discovering new market patterns and confirming existing trading rules and patterns [1].

GP has historically been successful at solving problems that other types of analysis cannot [5]. Despite this, many established financial organizations are not aware of possible benefits of applying GP to financial problems, in addition to being extremely wary of using any new computer model due to possible risk introduced. This has resulted in scant evidence of its mainstream usage [1].

In this paper, the benefits of using GP in real-world financial trading scenarios are shown through exploration of two studies that used different types of GP to create models that achieve better financial returns than existing mainstream standard financial models. In Section 2, the necessary background information on GP and financial terms are discussed. In Section 3 a recent GP model used for financial returns and its results are discussed. In Section 4, a different type of GP modeling called Strongly Typed GP is used to generate trading rules. In Section 5, a final third type of analysis using GP known as sentiment analysis is used in combination with technical analysis to predict return. In Section 6, the conclusion, the overall takeaways for why this is important and what the future may hold for the usage of GP in finance are discussed.

2 Background

To fully understand the topic, which has both financial and computer science related components, background information on both is important. Relevant GP background is in subsection 2.1, and finance background is in subsection 2.2.

2.1 Genetic Programming

Before going further, it is important to explain the basics of how GP works. GP typically starts with an unfit population of functions, then it uses various methods to generate new similar functions which ideally perform better, and repeats this process until a given criteria is met. Two common operations used in GP are crossover and mutation. Crossover takes two functions and swaps them at a randomly selected point to generate a new function. Mutation selects a random point on the function and swaps the value. Standard GP does not require that all inputs and outputs have a single type, but there is a version called Strongly Typed Genetic Programming (STGP) which requires each function to have a specified type for each argument and the value it returns [6].

A typical way that GP functions are visualised is through a syntax tree. A syntax tree is a tree-structure that captures the order in which the function components within a program

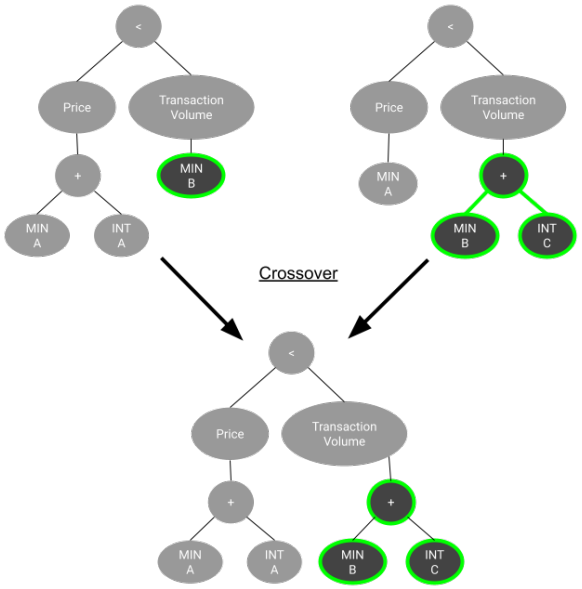

Figure 1. An example of crossover operation with syntax trees [6]

execute. The program output is the root node, functions are internal nodes, and terminal arguments are leaf nodes. Syntax trees were chosen to visualise GP programs because they allow for the nodes that change between generations to be easily shown to show how a function changes over time (see Figure 1).

There are two types of data used for testing performance of a GP algorithm. In-sample data are the existing data used to evolve the model, and out-of-sample data are data that exist but are not used at the time of the models creation. A model is usually more effective at predicting in-sample data because that is what it is trained on.

A GP algorithm typically uses the following series of steps in order to create a model [3]:

1. Generate a random population of functions and evaluate the quality of each function using the return formula. The quality metric is called fitness, which is the difference between the return value of the function and the desired return.
2. Multiple functions are then selected based on fitness as the parent functions which breed to create new child functions. It is done through both the crossover operation and the mutation operation. The crossover operation uses two parent functions and chooses a random point to split the parent models' variables and then swaps the split part of one model with the corresponding split part of the other other to create new children. The mutation operation chooses variables at random and alters them to a different value, then the new altered function becomes a child function.
3. The fitness of the new child functions is then calculated and the best ones are used as the parent functions of the next generation. Eventually after a given number of generations, the best individual is returned and becomes the new return forecasting model.

2.2 Financial Models

An important factor in financial model development is market efficiency. This is how the market reacts to available information, with higher market efficiency meaning that more actionable information about the value of a stock or company is available. The US market has high market efficiency and as a result, models have to be efficient at utilizing new information. An index fund is an asset with a portfolio that is meant to mimic a financial market. One of the most widely followed indexes is the Dow Jones Industrial Average (DJA) which tracks 30 major US stocks with a history of great returns. Another important index is the Standard and Poor's 500 (S&P500) which tracks 500 large publicly traded US stocks. One example where GP has had beneficial results is predicting stock market futures [3]. Futures are contracts that obligate the buyer to purchase an asset for a set price at a future time. The buyer will make or lose money depending on how the price has changed relative to the original price when the contract executes. During times of financial crisis where many assets lose most of their value, GP generated models have been shown to still make accurate predictions. The models also can do later analysis on causes and effects of the crisis [5].

To understand how effective a model is, the concept of return is used with the following formula:

$$\text{return} = \ln\left(\frac{P_t}{P_{t-1}}\right)$$

where P_t is the closing price of a stock on a given date t, and P_{t-1} is the previous closing price. Terms that are before others in time such as P_{t-1} are also referred to as lagged terms. The log is taken because it helps to scale how returns compound [3]. The price and return of a stock also is usually normally distributed because extreme changes become less likely as the price approaches zero. Since return has a normal distribution, and in this case is a log variable, return is considered to be a log-normal distribution [3].

In addition to the base return value, the Sharpe ratio is also used to measure the return of an investment relative to its risk. The formula for Sharpe Ratio is:

$$\text{Sharpe Ratio} = \frac{R_p - R_f}{\sigma_p}$$

where R_p is the return of an asset, R_f is a risk free asset rate, and σ_p is the standard deviation of the asset's excess return or different between the return and the return of a comparable benchmark. If a higher investing risk does not

lead to a higher return compared to a risk-free asset, then it is considered a bad investment [4].

The usual model used for return forecasting is an autoregressive (AR) model, meaning that it predicts future values based on past performance. AR models work under the assumption that past values have an effect on current values. This is often used because returns over shorter periods of time usually have a normal distribution, falling within three standard deviations of the mean return unless market disrupting events occur. Because of the assumption of a normal distribution, while AR models are accurate for general whole market trends and stock values, there are exceptions in widely fluctuating individual stocks where the models can be inaccurate.

One of the studies discussed below [3] also contains jump-diffusion, which is a stochastic process composed of two components, a jump component and a diffusion component. The jump component models the probability of a jump based on the time between jumps and number of jumps. The diffusion component then uses the logarithm of the randomly varying jump component to model the expected stock price more accurately. Jump-diffusion is useful to account for increased market volatility.

All of the studies used in this paper used training data that included the 2008 financial crisis, which was a period of global extreme economic uncertainty due to low lending standards causing a house pricing bubble to burst leading to many banks failing. While this is a complex issue, it is important to note that the unusual market fluctuations during this period make for useful testing data to show how the models work in different financial circumstances.

The Bayesian Information Criterion (BIC) was used in order to help find the best starting model. BIC is used when there are many parameters and a dependent variable, in this case financial return. The likelihood of improving the return, and the number of variables in the model are both used to return a BIC value where a lower value indicates a lower chance of error. BIC is mainly used as a heuristic to confirm that the model is continuing to improve over time, and not the primary measure for model selection. In addition, mean square error (MSE) is used to measure model accuracy, with lower MSE meaning higher accuracy. The definition of periodic averaged MSE is defined as:

$$MSE_T = \frac{1}{T}\sum_{t=1}^{T}(\text{Observed}_t - \text{Predicted}_t)^2$$

where T represents the number of observations, Observed_t presents the observed market variance, and Predicted_t presents the predicted market variance from the models [3]. Variance is used because it shows how good the model is at predicting what the next price will be.

3 Forecasting market return with genetic programming

In the study by Ding, Cui, Xiong, and Bai, [3] the goal was to see how predictable the stock market is by using a forecasting model developed using genetic programming (GP). The reason why the GP approach was chosen was because of the non-linear nature of return forecasting; meaning that returns have a large range of possible results, and tend to follow a logarithmic scale rather than a linear scale. The GP model also will develop its own rules which may be different than what other existing models use.

The authors used data from four countries' daily stock index database from January 1, 2006 to December 31, 2017 as the base for the model (These are summarized in Table 1). They used the developed economies of the United States and Japan, and the emerging economies of India and China to see how the model would differ across the two setups. In the case of the US market for example, since there are already many computers performing trading operations with complex models, it may change how predictable the market is compared to another market with less competition and computer trading involved, such as China. In addition, China and India have caps on the percentage the price of a stock can fluctuate in a given day, and if they reach this cap, the trading of the stock halts for the day. On the other hand, the values of stocks in developed economies often fluctuate rapidly over short periods, which makes model predictions harder. To account for this, they smoothed the data to reduce the noise.

Multiple types of autoregressive models were used for the linear and non-linear benchmark models. For the linear model, the standard autoregressive (ARMA) model was used. Linear models assume a linear relationship between past and future returns. For the non-linear model, the self-exciting threshold auto-regression (SETAR) and linear smooth transition autoregressive (LSTAR) models were used [3]. Nonlinear models assume a non-linear relationship between past and future returns. The specifics of these models are not necessary information other than they are commonly used for financial analysis, and provide diversity for the new models being generated. Multiple types of autoregressive models were used in order to find the best one. The goal was to maximize the likelihood, or how well the model fits the data based on the estimated variable values in the model. The authors used BIC as the fitness metric, and started with the model: $f(r_{t-1}, r_{t-2}, r_{t-3}) = r_t$ where r_{t-1}, r_{t-2} and r_{t-3} are lagged terms. They then ran the algorithm (see Algorithm 1) for the chosen number of generations.

The samples were divided by developing and emerging economies. Both of these sub-samples ran the GP algorithm 50 times, and they extracted the best function for both subsamples. The resulting models were different with the developed model having three natural log terms and the emerging

Country	Obs	Mean	Min	Max
US	3020	0.00025	-0.094	0.109
Japan	2957	0.00012	-0.121	0.132
China	2916	0.0005	-0.118	0.134
India	2928	0.0004	-0.116	0.159

Table 1. Shows by country the number of daily return observations as well as then mean, minimum, and maximum daily returns [3]

Algorithm 1: GP for Stock Market Return Forecasting Model

```
Initialisation
    Initialise the population of the first generation
while not find the 'good enough' forecasted model or
  not reach the maximum number of generations; do
    for each individual forecasted model in the
      generation do
        Evaluation
            Evaluate each forecasted model's fitness
    end
    Select Parents
        Select the individual forecasted models from
          the population of the current generation to
          breed
    Crossover
        Pair the selected parents up to produce
          offspring forecasted models
    Mutate
        Randomly alter the forecasted model with a
          given probability
    Elitism
        Select the best forecasted model from the
          population of the current generation and
          insert it into the next new generation
    Update Population
        Update the population of the current
          generation
end
```

model having zero. The developed model is:

$$r_t = r_{t-2} + r_{t-3} + r_{t-1}^2 + \ln(r_{t-2} * r_{t-3})I(r_{t-2} * r_{t-3} > 0) + \ln(r_{t-3}^2) + \ln(r_{t-3}^3)I(r_{t-3} > 0)$$

where r_{t-q} is the lagged term and I is an indicator function: $I = 1$ if the condition in the parenthesis holds and $I = 0$ otherwise. The emerging model is:

$$r_t = r_{t-2} + r_{t-3} + r_{t-1}^2 + r_{t-2}^2 + +r_{t-3}^2 + r_{t-2}r_{t-3}$$

where r_{t-q} is the lagged term. The authors speculated the log terms were caused by jump-diffusion. The natural log items are most likely the jump ingredient in the developed markets [3]. The reason this did not show in the emerging market model is because in China and India, the daily change in price is bounded within ±10%, which reduces the jump probability. These differences show that the two models have distinguishing features for each market type.

Model:	ARMA	SETAR	LSTAR	NRFM1	NRFM2
US	6.00e-05	5.78e-05	5.77e-05	4.06e-05	
Japan	3.62e-04	1.80e-04	1.79e-04	1.73e-04	
China	2.53e-04	2.54e-04	2.52e-04		1.74e-05
India	8.33e-05	8.28e-05	8.28e-05		4.63e-05

Table 2. MSE values of the models for out-of-sample data [3]

3.1 Results of models

After the final models evolved, the mean squared error (MSE) value was calculated to measure the model performance for in-sample and out-of-sample tests [3]. The model that was used for developed economies is referred to as NRFM1, and the model for emerging economies is NRFM2. For out-of-sample forecasting, the GP generated models had an average improvement rate of 32% compared to the existing commonly used models of ARMA, SETAR, and LSTAR (see Table 2). The developed country model also made better predictions which is assumed to be due to developed market prices having greater market efficiency, meaning it accurately reflects available information in its prices. For out-of-sample forecasting, the inverse was true, with the developed model being less accurate than the developing model due to the future being easier to predict due to lower market efficiency and more market constraints.

The authors proposed using the GP models for futures trading and through testing found that it increased returns by 10% or more compared to the ARMA, SETAR, and LSTAR models [3]. These statistics show that GP models can be used to forecast stock returns more effectively than existing standard models in the given scenarios. The two GP models effectively show how the future of forecast modeling may use GP for the best results.

4 Generating Trading Rules from Strongly Typed Genetic Programming

Beyond regular GP, there is a form referred to as Strongly Typed Genetic Programming (STGP). STGP is an enhanced version of GP that was first proposed in 1993 and sought to allow functions to take and return arguments of any data type. STGP requires that for each function the data types of every argument and every type returned are specified beforehand [6]. The mutation and crossover operations are different due to type requirements. In a mutation, the variable that is changed has to be the same type as the original, and with crossover, the point in the second parent must be selected so that it returns the same type as from the first

parent. If there is no such node, then the crossover operator returns either the parents or nothing. This whole process is most easily represented as a syntax tree, which is a hierarchical representation of a function (see Figure 1). Because the data has to match a type-correct syntax tree, the space that the STGP algorithm searches is smaller than regular GP where all combinations are valid.

In a paper published in 2020, Michell and Kristjanpoller proposed the idea of generating stock market trading rules using STGP [5]. The authors theorized that because STGP requires the root node has a set return value type, this method could result in generating unique trading rules. The rules they generate have an output of buy, sell, or hold. Then the output is compared to the actual best decision based on the value of the return to see if the decision was good or not. The percentage of correct signals sent is then used to measure the fitness of the model. Data used for the STGP model came from from the US market, and used the US stock market indexes as a performance benchmark to beat. The generation and population metrics chosen were small, with 40 generations and a population size of 60 [5].

4.1 Proposed Model

The model aims to minimize risk by using the daily value of the US Federal Reserve fund rate as the minimum risk-free return, designated as r_f. Treasury bonds are considered risk-free since they are backed by the US government. The strategy also did not account for other regular types of market activity that were restricted during the analysis period due to the 2008 financial crisis [5]. The model using a rolling window approach where it uses a set number of days for modeling, which was 252 days, or one financial year in this study. After this the model predicts a chosen number of days into the future, ten in this study or two financial weeks.

The authors decided to also consider additional windows to see how it influenced model behavior, using the additional windows of 126 (half a year) and 504 (2 years) days for training, and 5 (1 week) and 22 (a month) days for predicting. After the predicted days have passed, the modeling process is repeated, but moves forward the chosen number of days (days 11-262 instead) [5]. This is repeated until the whole chosen period is covered. The authors settled on a period from January, 2003 to November, 2015 because it contained pre-crisis, crisis, and post-crisis periods.

When the period is complete the daily average return is calculated and compared to other models to measure performance. The 90 most traded stocks over the period were used for the STGP model. The model chooses which signal to send based on a three part rule which aims to minimize risk in the portfolio and maximize return. The rule takes a stock at a specific time and calculates the return, then compares it using the following steps [5]:

1. If the estimated return is greater than r_f, buy the stock.
2. If the estimated return is less than 0, sell the stock.
3. For other cases do not trade the stock, since in this case the estimated return is lower than r_f meaning the risk is too great for the possible return.

When using the training data, the model could see if it was making predictions that improved returns, but with the testing data the model had to make its own predictions based on the trends it had observed without knowing the return.

4.2 Model Results

The STGP model was compared to various benchmarks to measure its effectiveness. When comparing the model to existing benchmarks, a transaction cost of 0.1% was set for buying or selling due to this being a standard value for real-world transactions at the time of the paper. This means that if the STGP model chooses to perform more transactions, there is a higher cost incurred, but there may also be a higher return. In addition, the Sharpe ratio was used to compare the risk in different investment strategies compared to the STGP model. The benchmarks used were the S&P500 and the DJA, which are index funds meant to represent general US market behavior. The model outperformed both benchmarks, improving the return over the DJA by 65.08%, and the S&P500 by 51.46% [5]. An additional GP model from 1999 was also used and achieve a 407.32% improvement, although it is worth noting that this model is not recent enough to represent modern GP return modeling.

They also compared the results to the common buy-and-hold strategy (B&H) for long-term investing, which consists of holding the stocks for the entire period based on the common market trend that the value of assets tend to always apprecite given a long enough time period. The 90 most traded stocks over the model period were used with the B&H strategy, and it was found that the STGP model still outperformed the benchmark by 17.74% [5]. In addition, 61.11% of the stocks in the STGP portfolio outperform their B&H counterpart. This is significant because a B&H strategy only incurs the 0.1% transaction cost when initially buying and finally selling, while the STGP model incurs the transaction cost many more times, yet still ultimately has a higher return showing it truly generated a smarter investment strategy (see Table 3).

After all the simulations were completed, the best prediction window was found to be 22 days ahead. This seems to show that the model does better at short-term forecasting than long term, still allowing for increased return. The model also chose what was considered the best rule (buy, hold, or sell) 50% of the time, which is significantly higher than the general case of 33.33% [5]. The overall conclusion reached is that once again, the STGP model outperforms existing

Benchmark	STGP Improvement (%)
US Fed Rate	435.16
DJA	65.08
S&P500	51.46
B&H	17.74
Standard GP	407.32

Table 3. STGP model performance compared to benchmarks.

benchmarks, showing the value of its usage in mainstream finance.

5 Sentiment Analysis

Beyond the purely math and computer driven patterns of investing, there also lies an element of human interest and motivation in investment strategies. Sentiment analysis takes this into account by analyzing details such as positive or negative word choice in financial articles in order to predict what the public opinion of a stock value is, and where it may head in the future. In their 2022 paper, Christodoulaki, Kampouridis, and Kanellopoulos [2] use GP along with both technical and sentiment analysis in order to measure the effectiveness of using them to predict stock prices. In this context, technical analysis means using statistical trends such as price movement and volume to predict future price movements. Once again the Sharpe ratio and risk were used as measuring metrics.

The text they used was from an article scraper that searched the company name, went into 20 pages for each company, then obtained 12 features for analysis. The text was then fed through the AFINN wordlist which contains more than 3300 words, each with an associated polarity score. These word polarity scores range from -5 to +5, indicating respectively how negative or positive the sentiment is, and together they add to a overall sentiment score for the given text. In addition, the python library TextBlob was used to measure the subjectivity of words and how they modify adjacent words, such as 'very good' [2]. If the TextBlob polarity was less than 0.5, meaning that the majority of the text was the same sentiment, and the AFINN sentiment was greater than 0.75, meaning that 75% of the sentiment words have a positive polarity, then a buy signal was sent for the stock. The AFINN and TextBlob polarity scores were used as an input in the GP system. The data used was from 26 major companies across different industries from 2015-2020 in order to use a diverse stock portfolio for testing.

Initially, the results found statistically significant evidence that for the 26 companies the sentiment analysis model held lower risk than technical analysis, but also a lower return ratio. When accounting for the Sharpe ratio though, sentiment achieved a better risk ratio than technical analysis, meaning that the overall return for the amount of introduced risk was better. The authors concluded that this is evidence that through GP sentiment analysis can be effectively used alongside traditional technical analysis methods in order to achieve better returns with lower risk. The authors also ultimately suggested they would like to perform further research to confirm which sentiment and technical indicators had the largest impacts on performance and they were not ready to officially promote sentiment analysis usage in a real trading environment, but their results were still promising [2].

6 Conclusion

In the near future, it seems highly likely that the interconnected relationship between technology and the financial market will continue to strengthen. While 50% of equity trades are performed using computers today, this only recently became possible, and the amount of trades using computers in the future will likely increase. The GP model developed by Ding, Cui, Xiong, and Bai, and the STGP model by Michell and Kristjanpoller may both be an indicator of where future financial analysis may be headed. Both models show how GP can be effectively applied to different areas of the financial trading in order to secure better returns for investors. In addition, the model by Christodoulaki, Kampouridis, and Kanellopoulos shows how technical and other forms of analysis can be effectively used simultaneously.

7 Acknowledgements

I would like to thank my advisor Nic McPhee, Elena Machkasova, and Kristin Lamberty for their insight and advice over the course of the research process. In addition, I would like to thank Maggie Luetmer for reviewing my paper.

References

[1] Anthony Brabazon, Micheal Kampouridis, and Michael O'Neill. 2020. Applications of genetic programming to finance and economics: past, present, future. *Genetic Programming and Evolvable Machines* 21 (June 2020), 33–53. https://doi.org/10.1007/s10710-019-09359-z

[2] Eva Christodoulaki, Micheal Kampouridis, and Panagiotis Kanellopoulos. 2022. Technical and Sentiment Analysis in Financial Forecasting with Genetic Programming. In *Technical and Sentiment Analysis in Financial Forecasting with Genetic Programming*. IEEE, 8 pages.

[3] Shusheng Ding, Tianxiang Cui, Xihan Xiong, and Ruibin Bai. 2020. Forecasting stock market return with nonlinearity: a genetic programming approach. *Journal of Ambient Intelligence and Humanized Computing* 11 (February 2020), 4927–4939. https://doi.org/10.1007/s12652-020-01762-0

[4] Yang Liu, Guofu Zhou, and Yingzi Zhu. 2020. Maximizing the Sharpe Ratio: A Genetic Programming Approach. SSRN, 55 pages. https://doi.org/10.2139/ssrn.3726609

[5] Kevin Michell and Werner Kristjanpoller. 2020. Generating trading rules on US Stock Market using strongly typed genetic programming. *Soft Computing* 24 (March 2020), 3257–3274. https://doi.org/10.1007/s00500-019-04085-1

[6] David J. Montana. 1995. Strongly Typed Genetic Programming. *Evolutionary Computation* 3, 2 (1995), 199–230. https://doi.org/10.1162/evco.1995.3.2.199