Using Genetic Algorithm to Optimize Agent Parameters in Stock Market

David Wang, Vienne Tan, Neeraj Narayanan and Aryan Pramanick
Department of Computing Science
University of Alberta
Edmonton, Alberta, T6G 2E8, Canada
davidhan@ualberta.ca and jiayee@ualberta.ca
nnarayan@ualberta.ca and apramani@ualberta.ca

December 6, 2023

Abstract

In the dynamic landscape of financial markets, the process of stock trading represents a nuanced intersection of multifaceted variables, where agents navigate complexities driven by information, market dynamics, and individual strategies. Over the past 50 years, multiple attempts have been made to create a mathematical model (formulation) that accurately captures these complexities and attempts to explain how an investor makes decision regarding stock purchases. Traditional models assume that the factors that affect an individual's decision to trade stocks will remain constant over time. While this is certainly simpler, it will be unable to account for the aforementioned nuances, thus providing an oversimplified version of the financial market. Such models also cannot take confounding variables into account, such as the interaction between risk and how diverse an agent's portfolio is. To overcome these difficulties, we use a Genetic Algorithm. A Genetic Algorithm is a method of solving global numerical optimization problems. The aim is to use a Genetic Algorithm to optimize an objective function, which represents the portfolio value of an agent (an individual trading stocks). The agents can mutate and generate progeny that inherit traits (parameters) from their parents, with certain stochastic mutations. This allows the system to incrementally build up changes and account for the complexities, while allowing us to gauge the relationship between confounding variables.

1 Introduction

In the intricate landscape of financial markets, the act of trading stocks is a multifaceted process that involves a myriad of variables, uncertainties, and decision-making mechanisms. Studies in this field have led to the formulation of multiple mathematical and computational models that attempt to explain how an individual makes decisions about purchasing stocks. One such model is the *Capital Asset Pricing Model (CAPM)* (Perold, 2004), which describes the relationship between systematic risk (the general perils of investing) and the expected returns. This model is still widely used in the industry due to its simplicity and ease of use. However, there are some limitations to such models. CAPM makes unreliable assumptions based on its assumption of a linear relationship between risk and return. Therefore, understanding the decision-making process of stock trading goes beyond the conventional models that assume perfect rationality and market efficiency. This is where we use Genetic Algorithms to circumvent these issues.

A Genetic Algorithm is an advanced optimization technique that draws inspiration from the principles of biological evolution. In this paradigm, an agent embodies a potential solution to the given optimization problem, interacts with the environment. Through iterative generations, the agent evolves, refines the solutions by combining genetic operators such as natural selection and stochastic mutation. In this paper, we use a GA system to model a financial market. An agent (investor) in this system is allowed to trade stocks, mutate, and generate progeny. The environment includes randomly chosen stocks and their changes in stock price. The goal of the agent is to maximize an objective function, which represents an agent's portfolio value. The agent has two parameter values that define their behaviour - *risk* and *diversity*. In each successive generations, small stochastic mutations are made to these parameter values inherited from the parent in order to maximize the portfolio value. The policy used by the agent to determine the amount of money invested in each stock depends on the values of these parameters. An agent with a higher risk is defined to pick stocks that have lower market capitalization, and vice versa. An agent with high diversity will invest money in a larger number of stocks.

Two comparison metric are also used to understand and interpret the results of this study. The first metric uses *Stochastic Sampling*. In this system, the environment that the agent participates in remains the same as the GA system, which includes the same parameters, policy and initial amount of money. In this system, the parameter values are randomly sampled from a pool of values, thus making the system completely random. The second comparison metric used is the NASDAQ-100, which is a stock market index representing the performance of 100 of the largest non-financial companies listed on the NASDAQ stock exchange.

2 Problem Formulation

We formulate our problem in terms of a Markov Decision Process : MDP (State, Action, Return)

- state (s) : A state $s \in$ S represents the current configuration of the system at a particular point in time. It holds the price of all stocks being traded.
- Agent (Ag) : An agent ag represents an entity that is participating in the stock market. In this case, ag is a tuple such that
  ag = (P, W, M)
  - P = Parameter Values,
  - W = Worth of agent's portfolio at time t,
  - M = Number of shares in each stock.
- Time(t) : Each epoch of time t represents a generation.
- Action space (A) : a set of all actions an agent can take
- Action ($\overrightarrow{a}$) : An action $\overrightarrow{a} \in$ A represents the current action that an agent takes. In this case, an action is a vector that defines the distribution of stocks in an agent's portfolio. It is determined by a parameterized policy.
- Return (worth): The current worth (in dollars) of portfolio at the end of each generation:
  $Return = \overrightarrow{S} \; . \; \overrightarrow{M}$
  - $\overrightarrow{S}$ = Vector that holds the stock prices
  - $\overrightarrow{M}$ = Vector that holds the number of shares held by the agent in each stock (distribution of stocks in agent's portfolio).
- Policy : The agent determines how much money they invest in each stock on the basis of a policy. Here, we define the policy using a random Gaussian distribution. G is normally distributed with mean μ and standard deviation σ:

$$G \sim \mathcal{N}(\mu, \sigma)$$

  In the formulation here:
  - μ = risk
  - σ = diversification

  The parameters vector includes two values - risk and diversification. Risk is the mean of the Gaussian distribution, and diversification is the standard deviation.

  The Gaussian distribution will determine the distribution of stocks in the agent's portfolio, that is, it will determine $\overrightarrow{M}$. The agent will invest more money into the stock that represents the mean of the distribution, with the diversity of the portfolio dependent on the variance of the distribution.

3 Related Work

Several approaches have been used to forecast stock markets, including artificial neural network (ANN) and support vector machine (SVM) (Vui et al., 2013; Kim and Shin, 2007). Deep learning is a generic term for an ANN with multiple hidden layers between the input and output layers. However, these neural network models are not able to provide specific explanations for their prediction results, in which this approach developed a stock market prediction model using the available financial data, such as real-time data, including transaction records. This create useful insights for us to analyze the ideal parameters from gaining the maximum amount of money by the end of the experiment. The related paper we referred to propose to overcome this limitation by applying a Recurrent Neural Network (RNN) to stock market predictions by implementing a Long short-term memory unit (LSTM) for sequence learning of financial time series (Chung and Shin, 2018). A hybrid model that integrates the LSTM network with a GA to search for a suitable model is used for prediction of the next-day index of the stock market (Chung and Shin, 2018).

4 Proposed Approach

The initial agent (Ag_0) will begin with a given amount of money in the investment portfolio, and they will begin with randomized parameters. Every year represents one generation. After the first generation, an offspring will be formed with the mutated parameters that it inherits from the initial agent (parent).

Algorithm 1: Genetic Algorithm (Risk Aversion, Diversification)

$Ag_0[P].risk\_aversion$= risk_aversion
$Ag_0[P].diversification$ = diversification
$Ag_0[M]$ = Gaussian(NAS_COMP, mean = risk aversion, variance = diversification)
$Ag_0[W] = S_0 * Ag_0[M]$
Ag_0 = (P, W, M)
$best\_agent = Ag_0$
for i = *time from 1:final generation* **do**
    $Ag_i[P].risk\_aversion = best\_agent[P].risk\_aversion + random.uniform(-0.1, 0.1)$
    $Ag_i[P].diversification = best\_agent[P].diversification + random.uniform(-0.1, 0.1)$
    $Ag_i[M] = Gaussian(NASCOMP, mean = Ag_i[P].risk\_aversion, variance = Ag_i[P].diversification)$
    $Ag_i[W] = S_i * Ag_i[M]$
    if $Ag_i[W]_i > best\_agent[W]_i$: **then**
        $best\_agent = Ag_i$
Return *best_agent*

In our pseudocode, our input is a Pandas dataframe $NASCOMP$ containing the complete 25 year history of all stocks in the NASDAQ Composite, and our output is the agent with the best portfolio value. The initial agent will receive a random value for the parameters (risk and diversification), a value between 0 and 1.

Initially, we have an array of stocks and their market capitalization values. These values are sorted from highest to lowest. We then use a normal Gaussian distribution parameterized using risk (μ) and diversification (σ) to determine the distribution of stocks in the agent's portfolio. The agent will invest more money into the stock described by the mean in our array of stocks, with the distribution of money into the remaining stocks before and after being defined by the diversification parameter. We use a Gaussian distribution as we know that the mode of the distribution will occur at the mean, with an equal distribution before and after the mean.

The agent's portfolio worth is determined by the dot product between state space (S) which has the price of the stock and the stock distribution (Ag[M]). Then for each generation the best agent generates an offspring with mutated parameters. Between the 2 agents in the environment the agent with a better portfolio value gets to survive and move to the next generation where it becomes the best agent. This cycle keeps going until the end of time where at the end we are left with the agent with the best portfolio value and hence we get the parameter values that maximizes the objective function.

5 Benchmark

A. Stochastic Sampling

In this algorithm, each agent will begin with a constant amount of money in the investment portfolio and receive a random value for the parameters (risk and diversification), a value between 0 and 1. The agent's portfolio worth is determined by the dot product between state space (S) which has the price of the stock and the stock distribution (Ag[M]) till the end of time. Hence every agent survives for the whole time period. Then we compare the portfolio value between all the agents and the one with the best portfolio value is returned as the output. In this approach we use stochastic sampling to generate the

parameters for all the agents.
The difference between using a GA and a stochastic approach is that we want to see how much better and faster, retaining information of the best agent and passing it to the next generation is compared to randomly generating parameters. The financial market is ever evolving just like GA and we use the stochastic approach as a benchmark to show how effective GA is in our model.

B. NASDAQ-100

The NASDAQ-100 is a composite index. This means that it is a collection of stocks - in this case, all from the American Tech Sector. The reason this exists is because it allows investors to easily diversify their investments over many different assets, dramatically lowering the overall risk of their investment.

6 Empirical Evaluation

In the proposed approach, the agent begins with 10,000 USD and invests a certain amount of money in each of the 500 stocks (defined by the policy).
The results of the experiment are graphically presented below. The agent's portfolio value are measured on the y-axis, with generation represented on the x-axis. As seen, the profit tends to fluctuate over time, reaching a highest value of approximately 208 times the starting value S. All parameter values obtained are the average of ten runs of the experiment.

The parameters obtained are:

- risk (μ) = 0.51
- diversification (σ) = 0.34

The overall trend of the graph shows that there is an increase in the portfolio value across the generation. We can see that the GA reaches its maximum portfolio value in approximately the seventh generation.

In the stochastic sampling approach, the agent also begins in the same environment as the first system. The agent in each subsequent generation receives a new random set of parameter values.

The results of this experiment are presented graphically. As we can see, the profit reaches a maximum value of about 30 times starting value. This profit, however, is lesser than the profit generated by the agent in the first system.

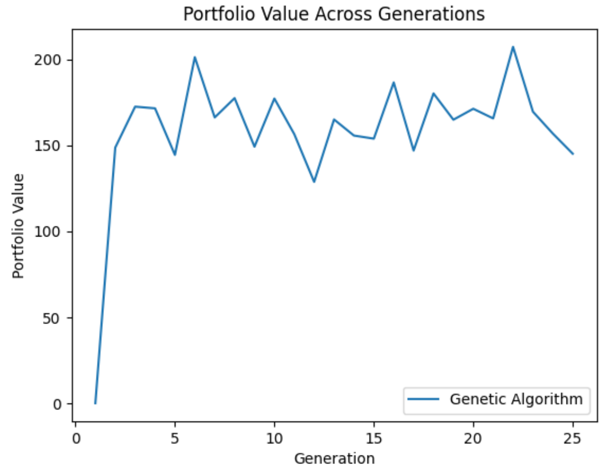

Figure 1: Performance with first approach

The parameters obtained are:

- risk (μ) = 0.87
- diversification (σ) = 0.72

The overall trend of the graph shows that the maximum portfolio value is obtained in the twelfth generation. However, this value is not maintained across generations and by the final generation, the portfolio value is lower than the peak.

The information for the NASDAQ's history was obtained online.

7 Theoretical Analysis

We expect our learning algorithm to run in $O(n)$ time, where n refers to the total number of generations. This is because there is only one for loop, which performs a constant number of computations per iteration. The maximum amount of time an agent in a given generation can interact with the environment is constant. At the end of all time steps, we analyze the total profit generated by the agent with respect to its parameters, which will be taken as a solution to the problem. Since we're

Algorithm 2: Stochastic Sampling (Risk Aversion, Diversification)

NAS_COMP is the data matrix containing the stock prices over the years.
Sort the elements of NAS_COMP according to the initial market capitalization.
$Ag_0[P].risk\_aversion$ = risk_aversion
$Ag_0[P]$.diversification = diversification
$Ag_0[M]$ = Gaussian(NASCOMP, mean = risk aversion, variance = diversification)
$Ag_0[W]$ = S * $Ag_0[M]$
Ag_0 = <P, W, M>
$best\_agent = Ag_0$
for *every iteration 1 to n* **do**
  $Ag_i[P].risk\_aversion$ = random value for risk aversion
  $Ag_i[P].diversification$ = random value for diversification
  $Ag_i[M] = Gaussian(NASCOMP, mean = Ag_i[P].risk\_aversion, variance = Ag_i[P].diversification)$
  $Ag_i[W] = S * Ag_i[M]$
  if $Ag_i[W]_i > best\_agent[W]_i$: **then**
    $best\_agent = Ag_i$
Return *best_agent*

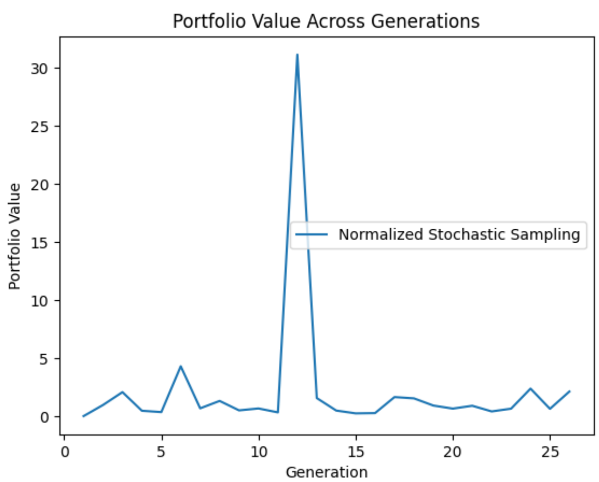

Figure 2: Performance with second approach

Figure 3: Real-life NASDAQ-100

GA	Stochastic Sampling	Benchmark
~ 208 * S	~ 32 * S	~ 30 * S

Figure 4: Comparison of approaches

	Genetic Algorithm	Stochastic Sampling
Average Risk	0.508304	0.873072
Average Diversity	0.338368	0.7153883

Figure 5: Comparison of approach parameters

using a greedy approach to evaluate the function, we predict the agent will increase its profits over time.

8 Discussion

From the results, we can see that a low average diversification and a relatively low average risk (compare to Stochastic Sampling), give us a higher portfolio value. Some amount of risk is required. The high risk and high diversification in Stochastic Sampling leads to a very irregular pattern in portfolio value. Making riskier investments means that the portfolio value is not guaranteed to be stable over time. There will always be periods of decline in the value when considering high risk and high diversity portfolios.

Comparing the two graphs, we can see that the peak in the GA system, is larger than the peak in the Stochastic Sampling system. This is because in GA, the minor mutations that are made add up over generations, and the better mutations survive. So the algorithm guarantees an increase over time. In SS, since the parameters are chosen randomly every iteration, there is no guarantee that a high portfolio value in one iteration will lead to a high value in the next. This trend is clear from the graph.

Some limitations of our approach are that we only have one agent at any one given time, and that we don't allow for the selling of stocks. Currently it is possible to lose a good solution in algorithm one in the early stages since we compare the objective function value from the start to the early stage of the model.

9 Future Work

- Increase the number of parameters. A real-time modeling of the financial markets requires the addition of multiple parameters, all of which change across generations and over time.

- Expand this system to a multi-agent system. One pitfall this experiment faces is that only one agent interacts with the stock market in a given generation. One possible avenue of research is to increase the number of agents, with the actions and decisions of one agent influencing the others.

- Add the action of selling and buying stocks midway when the agent thinks it's appropriate, to maximize the objective function.

- Combine algorithm one and algorithm two. This means allowing every agent in algorithm one to hold the stock for the whole time period.

10 Conclusions

The goal of our experiment is to determine the best average parameters of an agent, which yield the maximum portfolio value. From the empirical analysis and discussion above, we can conclude that an ideal combination of parameters would be one where both risk and diversification are relatively low (as compared to Stochastic Sampling).

By this, we mean that the risk can be between 0.4 and 0.5, and diversification between 0.3 and 0.4. This agrees with both our hypothesis and our general behaviour in financial markets, where an agent tends to concentrate their investment capital in companies with high market capitalization, which are more likely to perform well in the long run.

References

Chung, H. and Shin, K. S. (2018). Genetic algorithm-optimized long short-term memory network for stock market prediction. *Sustainability*, 10(10):3765.

Kim, H.-j. and Shin, K.-s. (2007). A hybrid approach based on neural networks and genetic algorithms for detecting temporal patterns in stock markets. *Applied Soft Computing*, 7(2):569–576.

Perold, A. F. (2004). The capital asset pricing model. *The Journal of Economic Perspectives*, 18(3):3–24.

Vui, C. S. et al. (2013). A review of stock market prediction with artificial neural network (ann). In *2013 IEEE international conference on control system, computing and engineering*. IEEE.