Machine Learning Engineer Nanodegree

Capstone Proposal (2nd submission

Lin Yu, Maria

May 2^{nd}, 2019

1 Domain Background

Stock investment can be one of the ways to manage one's asset. Technical analysis is sometimes used in financial markets to assist traders make buying and selling decisions [1]. Many technical analysis trading rules are deterministic trading policies. [2] uses genetic algorithm to find technical trading rule. [3] studies evolutionary algorithms in optimization of technical rules for automated stock trading. [4] proposed a stock trading system based on optimized technical analysis parameters for creating buy-sell points using genetic algorithms. [5] studies the selection of the optimal trading model for stock investment in different industries. [6] describes the optimization of trading strategies.

The optimization of trading rule using genetic algorithm or evolutionary algorithms belongs to policy-based method, which is a branch of Machine Learning. Policy-based methods try to directly optimize for the optimal policy which is an important branch for domains with continuous action spaces [7]. There are studies focus on how to find a trading strategy via Reinforcement Learning (RL) [8] or using Deep-Q learning for automatic trading algorithm [9]. But in this study we will focus on the policy-based method using Generic Algorithm that directly search for the optimal parameters of a deterministic policy.

Yahoo Finance's stock history data [10] will be used in this study. The reason to choose Yahoo Finance data is because it is free and available for public to assess. The performance of algorithm will be evaluated using different stocks.

The purpose of the study is to see the difference between using an agent with optimized policy to manage one's asset with buy-and-hold strategy, or manage one's asset with an agent with unoptimized policy.

2 Problem Statement

The goal is to create a usable tool with iPython Notebook on Macbook Pro; the tasks involved are the following:

1. Download and preprocess the Yahoo Finance data.
2. Choose and train an agent that is able to manage the trading such that it maximize the initial investment.
    - Choose a trading policy
    - Choose a trading agent

- Define how to split training and testing data. Here in this study, I use past one year data (April 2017- April 2018) for training and the current one year data (April 2018 - April 2019) for testing.
- Optimize the policy hyper parameters using the training data
- Use the trading agent with the trading policy with optimized hyper parameter to conduct trading for the testing data.

3. Compare the Agent's performance on trading with the optimized hyper parameters for the select stock (AAPL) with respect to the buy-and-hold stratety and see whether optimizing hyper parameter of a trading policy is able to help achieving better gain for the testing data.

The intention of the solution is to create a tool to help one make decision on managing his/her own asset and to show that with optimized hyper parameters, the trading agent use the same trading policy, it may be able to perform better than buy-and-hold.

3 Datasets and Inputs

I decided to get data from Yahoo Finance for stock prices. It provides historical data that can be downloaded as csv files or read into a Python DataFrame. I will choose AAPL as the first example to study because it has high transaction volume and good long term prospect. It is suitable to be used as an investment tool to combat with inflation.

The historical data fetch from Yahoo Finance contains: **Date**, **Open**, **High**, **Low**, **Close**, **Adjusted Close** and **Volume**. The reason that the **Adjusted Close** is chosen is because it accounts for all corporate actions such as stock splits, dividends/distributions and rights offerings.

4 Solution Statement

The goal is to create a usable tool with iPython Notebook on Macbook Pro. The tasks involved are the following:

1. Download and preprocess the Yahoo Finance data.
2. Choose and train an agent that is able to manage the trading such that it maximize the initial investment.
    - I will choose abcd trading policy [18]
    - I will choose buy_stock trading agent [18]
    - Define how to split training and testing data. Here in this study, I use past one year data (April 2017- April 2018) for training and the current one year data (April 2018 - April 2019) for testing. I will repeat this study for using April 2014-April 2015 data as training data and April 2015- April 2016 as testing and so on for 4 years. I will also repeat the study for another stock for 4 years.
    - I will choose Generic Algorithm or alternative optimization algorithm that is suitable for this study [17].

- Optimize the policy hyper parameters using the training data.
- Use the trading agent with the trading policy that uses the optimized hyper parameters to conduct trading for the testing data.

3. Compare the Agent's performance on trading with the optimized hyper parameters for the select stock (AAPL) with respect to the buy-and-hold trading strategy and see whether optimizing hyper parameter of a trading policy is able to help achieving better gain for the testing data.

5 Benchmark Model

I choose the investment return with buy-and-hold strategy as benchmark strategy. Investment return for buy-and-hold will calculated as buying the maximum number of share with the initial investment of $10,000$ at the price of the beginning ten day's average price and selling the number of share bought at the price of the ending ten day's average price. The ROI, Sharpe Ratio, Sortino Ratio of the benchmark buy-and-hold strategy is tabulated below for training data and testing data. The detailed definition and description of these ratios are given in the next Section.

Training data: Benchmark buy-and hold strategy performance

	ROI	Sharpe Ratio	Sortino Ratio
Buy and Hold	20.69%	6.68%	5.97%

Testing data: Benchmark buy-and hold strategy performance

	ROI	Sharpe Ratio	Sortino Ratio
Buy and Hold	20.96%	5.87%	7.78%

6 Evaluation Metrics

6.0.1 Return of Investment

A very simple investment return is defined similar to interest that banks give per year. The investment return in this project is defined as gain in percentage with respect to his/her initial investment.

$$Return\ of\ Investment = \frac{Assets\ Value}{Initial\ Investment} \tag{1}$$

6.0.2 Sharpe Ratio

Definition of the Sharpe Ratio is given by [11]. The Sharpe ratio was developed by Nobel laureate William F. Sharpe and is used to help investors understand the return of an investment compared to its risk. The ratio is the average return earned in excess of the risk-free rate per unit of volatility or total risk.

Subtracting the risk-free rate from the mean return allows an investor to better isolate the profits associated with risk-taking activities. Generally, the greater the value of the Sharpe ratio, the more attractive the risk-adjusted return. Sharpe Ratio is defined as:

$$Sharpe\ Ratio = \frac{R_p - R_f}{\sigma_p} \tag{2}$$

where, R_p is the return of portfolio, R_f is risk-free rate, σ_p is standard deviation of the portfolio. Here we use $R_f = 2.5\%$ as risk-free rate through out this study as this is the highest risk free deposit rate in Singapore. We didn't use the benchmark as the risk free rate as most common people don't regard stock. buy-and-hold strategy as risk-free.

6.0.3 Sortino Ratio

The Sortino ratio was designed to replace Sharpe ratio as a measure of risk-adjusted return [12]. It is defined as:

$$Sortino\ Ratio = \frac{R_p - R_f}{\sigma_d} \tag{3}$$

where, R_p is the return of portfolio, R_f is risk-free rate, σ_d is standard deviation of the down side of the portfolio.

In this study, I will use Return of Investment (ROI) as the cost function to optimize the trading policy parameter and we will evaluate the result with both Sharpe Ratio and Sortino Ratio. The reason for choosing purely ROI as the cost function this is that if ROI is optimized with a buying low and selling high strategy, then the Sharpe Ratio and Sortino Ratio shall be automatically taken care of. We will evaluate both metrics to verify this hypothesis.

When I choose the trading policy and trading agent, I did consider the processing time, where the processing time includes the time for the policy to give a suggestion and the trading agent to give ROI. This is because if it takes very long time for the trading agent to give a ROI for each buying for selling step, then the optimization process will be very long as optimization always use the performance metric ROI as a guidance to optimize the hyper parameter of a trading policy.

The time for a policy to give a suggestion is also critical as as asset management can be a time consuming task if the program takes very long time to give a suggestion. This may cause user to lose patient on using it.

7 Project Design

7.0.1 Problem Formulation

We model the problem as an investment episode. Each episode last for one year. The trading agent is given an initial amount of cash to invest. At the end of the episode the agent may hold cash or shares or both. Figure 1 shows the state space and action space. The state space include the cash holding, the number of share and the asset value (cash plus share value). The action space includes three possible actions: buy x number of shares, sell x number of shares and do nothing.

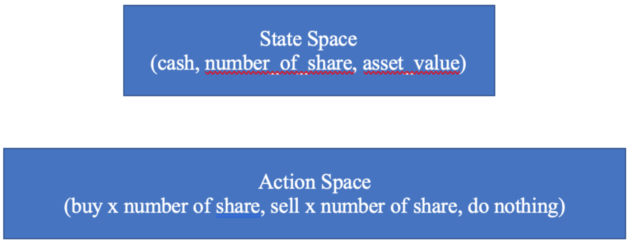

Figure 1: The state space and action space.

7.0.2 Objective Function

The objective function can be a state value, can be average state value, average action value or a function of reward. The best true objective function is a function of reward.

$$J(\theta) = E_{\pi}[R(\tau)] \tag{4}$$

Here, $\tau = S_0, A_0, R_1, S_1 ....$ in which S is state, A is action, and R is reward and π, the trading policy $\pi(S, A, \theta)$, is a function of state, action and policy parameters, θ. Therefore, J is a function of θ.

7.1 Policy and Optimization Pseudo Code

Here in this study we focus on deterministic policy:

$$\pi : s \to a \tag{5}$$

The pseudo code for the optimization of the the $J(\theta)$ is given below:

Algorithm

Generate initial population of θs arbitrarily

for the episode for each θ in the generation:

    evaluate $J(\theta)$

    θ = best θ of the generation

    While ($iteration < max\ iteration$)

        Breed a new generation of θs from the best θ

        for each θ: evaluate $J(\theta)$

        Select the best θ that gives the best $J(\theta)$

Usually the termination of the optimization can be controlled by maximum iteration or when $\Delta J(\theta) < threshold$. Here we choose maximum iteration to better control the simulation time.

The problem that I studied is on how to manage one's asset through stock trading using a trading program with a trading policy. The trading policy and trading agent's parameters will be optimized using training data. We will use the trading agent and policy with the optimized parameter to evaluate the performance of testing data.

The main problem to solve is to optimize the policy and trading hyper parameters. I plan to use GA for the optimization of the hyper parameters. Fig. 2 shows the conventional Generic Algorithm implementation diagram [17]. It includes the following steps:

1. Input data is read
2. an initial population of M genotypes is generated.
3. A fitness function is applied to determine the fitness of each genotype..
4. A new generation of genotypes is bred by using reproduction, crossover and mutation.
5. The fitness of each genotype is again evaluated.
6. The M most fit genotypes are selected from the new generation.
7. A determination is made as to whether the best genotype resulting from the previous Selection is Satisfactory.
8. If the best genotype is not satisfactory, Steps 4 to 7 are repeated.
9. If the best genotype is satisfactory, the best genotype is displayed.

A typical genetic algorithm requires:

1. a genetic representation of the solution domain,
2. a fitness function to evaluate the solution domain.

In this project, I plan to represent the policy and trading hyper parameters as the genotype which are to be optimized. I may choose ROI as the fitness function for evaluation of the fitness of the hyper parameters. I may modify or change the optimization algorithm if it doesn't work well for my problem.

With the project design, we can also see that this study assumes that certain underlining statistics of the stock remain unchanged for training data and testing data (from previous year to current year). It is expected that using optimized parameters may not always beat the performance of the default settings and may not work always work well for future data.

Nevertheless, in this project, I will use the Machine learning knowledge to optimize the hyper parameters of the trading policy and trading agent using its previous year's data and hopefully it works well for the next year's data. I will also use the program to evaluate its performance for a few different stocks.

References

[1] Wee Mien Cheung et al, “A Fuzzy Logic Based Trading System”, https://pdfs.semanticscholar.org/4cfd/0ab4a3bcd883a742a0d4706746859d075fb5.pdf.

[2] TIan Yu et al, “Using Genetic Programming with Lambda abstraction to find the Technical Trading Rules”, https://www.researchgate.net/profile/Shuheng_Chen/publication.

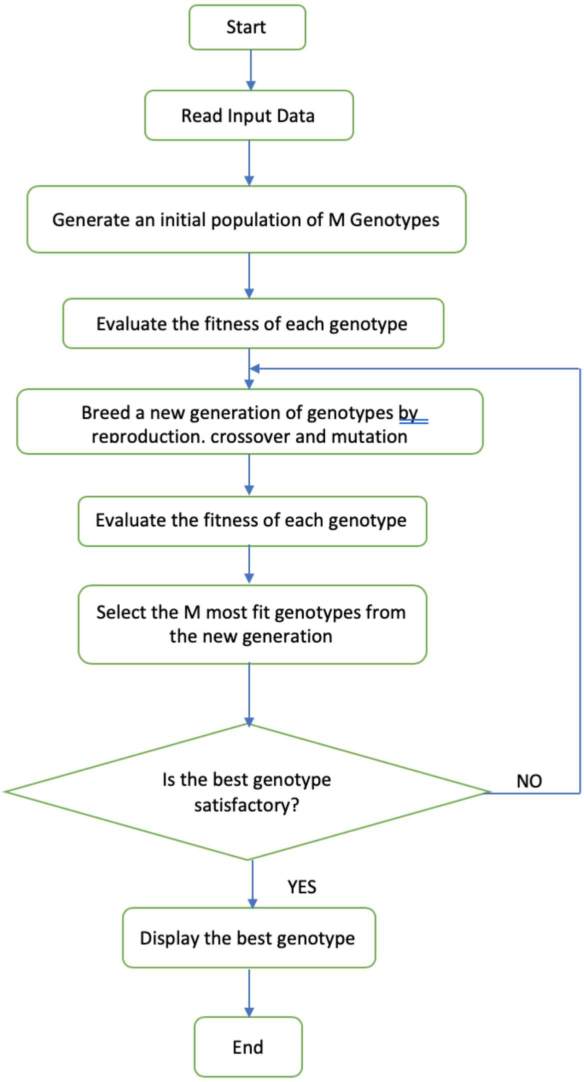

Figure 2: GA diagram.

[3] Harish K Subramanian, "Evolutionary Algorithms in Optimization of Technical Rules for Automated Stock Trading", http://www.cs.utexas.edu/ pstone/Courses/395Tfall06/resources/Thesis-harish.pdf.

[4] Sezer OB, Ozbayoglu M, Dogdu E. "A Deep Neural-Network Based Stock Trading System Based on Evolutionary Optimized Technical Analysis Parameters", Procedia Computer Science. 2017; 114:473–480.

[5] Dongdong Lv et al, "Selection of the optimal trading model for stock investment in different industries", https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0212137, February 13, 2019.

[6] Pardo R. The Evaluation and Optimization of Trading Strategies. 2nd ed. Hoboken:

John Wiley & Sons; 2008.

[7] Udacity, “Udacity Machine Learning Engineer Nanodegree Program, Part II, Lesson 18: Policy-Based Methods”, http://www.udacity.com.

[8] Pierpaolo G. Necchi, “Reinforcement Learning For Automated Trading”, https://pdfs.semanticscholar.org/f94b/ea4efc52acf695131c3128294a0f039629a8.pdf

[9] Wang et al., “Deep Q-trading”, CSLT TECHNICAL REPORT-20160036 [Sunday 8th January, 2017], http://cslt.riit.tsinghua.edu.cn/mediawiki/images/5/5f/Dtq.pdf

[10] Yahoo Finance, https://finance.yahoo.com.

[11] Investopedia, https://www.investopedia.com/terms/s/sharperatio.asp, 2019.

[12] Investopedia, https://www.investopedia.com/terms/s/sortinoratio.asp 2019.

[13] Singapore Gov Website, https://data.gov.sg/dataset/private-residential-property-price-index-by-type-of-property., Singapore, 2019.

[14] Singapore Gov Website, https://www.ura.gov.sg/Corporate/Property/Property-Data, 2019.

[15] Investopedia, https://www.investopedia.com/terms/a/adjusted_closing_price.asp.

[16] Yahoo, https://finance.yahoo.com/quote/SPY/holdings/.

[17] Joseph Thomas O’Neil, ”Method for utilizing a Generic Algorithm to provide constraint-based routing of packets in a communication network.” United States Patent No. US 6,912,587, Jun 28 (2006).

[18] Huseinzo, https://github.com/huseinzol05/Stock-Prediction-Models/tree/master/agent.

[19] Mitchell, Melanie. An Introduction to Genetic Algorithms. Cambridge, MA: MIT Press, 1996.

[20] Sadeghi, Javad; Sadeghi, Saeid; Niaki, Seyed Taghi Akhavan, ”Optimizing a hybrid vendor-managed inventory and transportation problem with fuzzy demand: An improved particle swarm optimization algorithm”. Information Sciences. 272: 126–144, 10 July 2014..

[21] Whitley, Darrell, ”A genetic algorithm tutorial” (PDF). Statistics and Computing. 4 (2): 65–85, 1994.

[22] S. Azodolmolky *et al.*, Experimental demonstration of an impairment aware network planning and operation tool for transparent/translucent optical networks,” *J. Lightw. Technol.*, vol. 29, no. 4, pp. 439–448, Sep. 2011.