



Expertise
and insight
for the future

Nhan Phan

Developing Robo-advisor Software with Semivariance and Black-Litter- man Model

Metropolia University of Applied Sciences

Bachelor of Engineering

Degree Programme in Information Technology

Bachelor's Thesis

15 May 2020

Author Title	Nhan Phan Developing robo-advisor software with semivariance and Black-Litterman model
Number of Pages Date	37 pages + 1 appendix 15 May 2020
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Professional Major	Smart Systems
Instructors	Sami Sainio, Senior Lecturer
<p>The purpose of this thesis was to develop a robo-advisor program for Japanese pension investment fund. A robo-advisor can reduce behavioral biases and provide an objective suggestion for investors. The program will make an investment recommendation on bond and stock market of major economies in North America, Europe and Asia.</p> <p>The program is based on Markowitz's Modern Portfolio Theory. The Markowitz framework is used to compute optimal investment portfolios that offer the highest level of expected return for a predetermined level of risk. However, since its release in 1952, many and even Markowitz himself have suggested improvements to the model to improve its application in real world problems. Therefore, the program also implements semivariance, Black-Litterman model and portfolio simplification into the traditional model.</p> <p>Due to the large amount of data and the complexity of the program, the program must be capable of computing convex optimization with reasonable speed. Program development also takes into consideration the broad user group. To make sure anyone without technical skill can use the model, the program was designed to target simplicity and usability.</p> <p>Python with CVXPY library for convex optimization and Tkinter for GUI library were chosen for program development. Other Python libraries, NumPy and pandas, were deliberately used due to their ability in handling data and solve mathematic equations.</p> <p>Improvements were backtested using 10 years of ETF historical data from Bloomberg. The results showed that those improvements reduce the limitation of Modern Portfolio in real world and generally raise the return of the investment. However, further studies should be considered during the extreme market condition.</p> <p>Based on the backtest result, the final program is a useful tool for asset allocation. However, its limitation in using historical return to forecast is not fully eliminated. Further implementation will focus on developing a better return prediction model, based on the asset's potential and not on its past performance.</p>	
Keywords	portfolio optimization, cvxpy, semivariance, Black-Litterman

Contents

List of Abbreviations

1	Overview	1
2	Modern Portfolio Theory	2
2.1	Return and risk	2
2.2	Efficient Frontier	4
2.3	Sharpe Portfolio and Minimum Variance Portfolio	6
2.4	Criticisms	6
2.5	Black-Litterman model	7
3	Implementation	9
3.1	Programming Language and Libraries	9
3.2	Portfolio Construction	12
3.2.1	Equity construction	12
3.2.2	Bond and Alternative Investment Construction	15
3.3	Program flowchart	16
3.4	Program user interface	19
4	Improvement	23
4.1	Semivariance	24
4.2	Black-Litterman model	27
4.3	Portfolio Simplification	30
5	Conclusion	33
5.1	Summary	33
5.2	Discussion	34
	References	36
	Appendix	
	Appendix 1. Portfolio allocation in March 2020	

List of Abbreviations

AUM	Assets under management. The total market value of all the financial assets which a financial institution manages on behalf of its client and themselves.
STDEV	Standard deviation. Statistics number measures the variation of a data set within a period.
EF	Efficient Frontier
ETF	Exchange-traded fund. It's a type of security that often tracks an underlying index and is listed on stock exchange. ETF can be traded on stock exchange like a normal stock.
GDP	Gross domestic product
BL	Black-Litterman. Black-Litterman model is a financial model in portfolio allocation.
GUI	Graphical user interface

1 Overview

According to U.S Securities and Exchange Commission, robo-advisors are automated digital investment advisory programs, which provide investment advice with minimum human interaction [1]. Customer enters their financial information into those programs and receives investment service, with potential benefit of lower managing costs compared with traditional services. In the 1st quarter of 2018, Backend Benchmarking estimated the total assets under management (AUM) of those robo-advisors is close to \$178 billion [2]. Two years later, in their 4th quarter of 2019 report, Backend Benchmarking approximate total AUM have increased 50% to \$275 billion [3]. The main hypothesis of this thesis is that by developing a robot investment software based on the improved Markowitz's Modern Portfolio Theory, it builds a solid theoretical foundation for the model and subsequently is possible to improve the return of investment compared to existing robo-advisors.

Other renown investment banks, such as Goldman Sachs or JPMorgan Chase, are investing in quantum computing technology. A quantum computer can solve complex Monte Carlo simulations fast and can provide portfolio optimization service for bank's clients. However, experts believe there are a lot of challenges and expect it would take them at least 5 years to achieve the desirable outcome [4]. Even with the algorithm and hardware for quantum computing would be available, the availability of such quantum computers will be limited. The program developed in this thesis will still benefit the regular stock market investors.

Without quantum technology breakthrough, we can challenge this problem with a simpler solution, by building an investment advisor program based on a famous economic theory: Modern Portfolio Theory. Modern Portfolio Theory allow us to construct a line of optimal portfolios – Efficient Frontier (EF) – that can dynamically change with new information. The theory was first introduced by Harry Max Markowitz in 1952 and brought him the 1990 Nobel Memorial Prize in Economic Sciences [5].

2 Modern Portfolio Theory

2.1 Return and risk

Asset return is the historical return of asset within a period, often longer than one year. In practical, and in this thesis, asset return is calculated yearly unless stated otherwise. Because we average asset return using geometric method instead of arithmetic method, asset return is also called compound annual growth rate (CAGR). For example, if price of stock A was \$100 in 31/12/2015, and was \$150 in 31/12/2018. Stock A's CAGR and yearly return is $\sqrt[3]{\frac{150}{100}} - 1 = \sqrt[3]{1.5} - 1 = 14.47\%$ (a year).

Asset risks are measured using standard deviation (STDEV). In statistics, STDEV was used to measure the variety of a data set within a period. In finance, investors use STDEV to measure the volatility of their investment return [6, p. 182]. Assets with high STDEV have high uncertainty that their actual return will be the same as their expected return and therefore investors regard them as high-risk asset.

Portfolio return, or expected return is calculated by taking the sum of the product of individual asset return in the portfolio with the weight of that asset in the portfolio. As asset return is CAGR, portfolio is also CAGR. Formula to calculate portfolio return is:

$$\text{Portfolio return} = \sum_{i=1}^n w_i \times r_i \quad (1)$$

where w_i is weight of asset i in portfolio, and r_i is return of asset i .

Portfolio risk is more complex than other parameters. On one hand, it has the same interpretation of asset risk, as it also measured risks using standard deviation. On the other hand, it requires more input parameters than portfolio return, and the number of input parameter increases aggressively the more asset we include in our portfolio. Comparable to portfolio return, portfolio risk requires asset return and asset weight. It, however, requires the third parameter: correlation between two different assets in the portfolio. If our portfolio only consists of asset A and asset B, we only need to compute

correlation between asset A and asset B. When include asset C in our portfolio, we will need to compute correlation between A and B, B and C, and finally A and C

$$\sigma_{AB}^2 = w_A^2 \sigma_A^2 + w_B^2 \sigma_B^2 + 2w_A w_B \sigma_A \sigma_B \rho_{AB}$$

$$\sigma_{ABC}^2 = w_A^2 \sigma_A^2 + w_B^2 \sigma_B^2 + w_C^2 \sigma_C^2 + 2w_A w_B \sigma_A \sigma_B \rho_{AB} + 2w_A w_C \sigma_A \sigma_C \rho_{AC} + 2w_B w_C \sigma_B \sigma_C \rho_{BC}$$

In general, the portfolio variance (σ_p^2) is calculated by the formula

$$\sigma_p^2 = \sum_i w_i^2 \sigma_i^2 + \sum_i \sum_{j \neq i} w_i w_j \sigma_i \sigma_j \rho_{ij} \quad (2)$$

where σ_i is STDEV of asset i in portfolio, w_i is weight of asset i in portfolio, ρ_{ij} is correlation between asset i and asset j .

Using the formula (2) in programming would require loop functions to get all possible companions of two assets to calculate portfolio risk. Instead, we can use matrix and vector for variance, and for more complex model introduced in next section. To be able to use matrix to calculate variance, first we need to calculate covariance, which measure the combined movement of returns on two assets. Like variance, covariance measures the volatility of return, but between two assets. Covariance can be calculated from assets' STDEV and their correlation by using formula:

$$Cov_{i,j} = \rho_{ij} \sigma_i \sigma_j \quad (3)$$

From the formula above, we can see that covariance between two assets that have no correlation which other is 0, and covariance between asset i and itself is its variance. We can calculate covariance matrix of a portfolio P with m assets by using matrix multiplication:

$$\begin{bmatrix} Cov_{11} & \cdots & Cov_{1m} \\ \vdots & \ddots & \vdots \\ Cov_{m1} & \cdots & Cov_{mm} \end{bmatrix} = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_m \end{bmatrix} \times \begin{bmatrix} \rho_{11} & \cdots & \rho_{1m} \\ \vdots & \ddots & \vdots \\ \rho_{m1} & \cdots & \rho_{mm} \end{bmatrix} \times \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_m \end{bmatrix} \quad (4)$$

where $\begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_m \end{bmatrix}$ is the diagonal matrix (all values outside the main diagonal are zero)

of assets' STDEV, and $\begin{bmatrix} \rho_{11} & \cdots & \rho_{1m} \\ \vdots & \ddots & \vdots \\ \rho_{m1} & \cdots & \rho_{mm} \end{bmatrix}$ is correlation matrix of all m asset in portfolio.

After getting covariance matrix, we can calculate the portfolio variance:

$$\sigma_p^2 = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}^T \times \begin{bmatrix} Cov_{11} & \cdots & Cov_{1m} \\ \vdots & \ddots & \vdots \\ Cov_{m1} & \cdots & Cov_{mm} \end{bmatrix} \times \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix} \quad (5)$$

where $\begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}$ is weight vector of all assets in portfolio

2.2 Efficient Frontier

Efficient Frontier (EF) is a curve line contains all the best portfolios possible from a combination of assets. Its x-axis represents the risk (STDEV or variance) of the portfolio, while its y-axis represents the expected return. When first introduced by Markowitz, it was called the Efficient E, V (Expected Return, Variance) Combinations [7]. The graph of an EF is shown in Figure 1 below. The following exchanged-traded fund (ETF) were used with 5 years monthly data to construct the EF at the end of 31 March 2020: SPY (SPDR S&P 500 ETF Trust), EMB (iShares JP Morgan USD Emerging Markets Bond), BIV (Vanguard Intermediate-Term Bond ETF), IWF (iShares Russell 1000 Growth ETF), VIG (Vanguard Dividend Appreciation ETF), XLK (Technology Select Sector SPDR Fund). The EF was built using the robo-advisor software, while randomized portfolios were obtained by randomized the weight of each ETF in the portfolio. While the EF is much longer, since the Sharpe Portfolio and Minimum Variance Portfolio were the focus, the rest of the EF line was cut out.

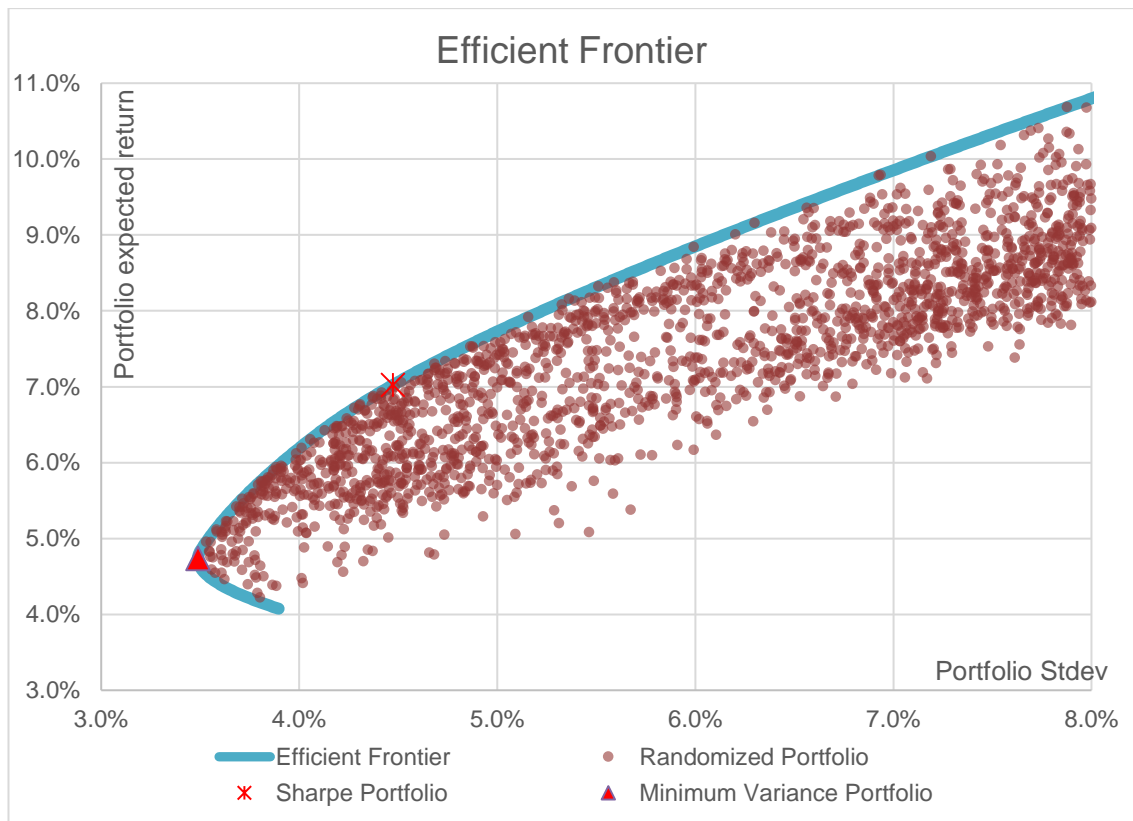


Figure 1. The Efficient Frontier

As illustrated in Figure 1, any point below the EF is a worse investment, as it offers lower expected return for the same amount of risk. Any rational and risk averse investor will not pick suboptimal portfolio, as it is an assumption in the Modern Portfolio Theory. It is also impossible to find any point higher than the EF, because the EF already contain all the highest possible expected return with a predetermined level of risk. Also note that the lower half of the EF (the line below 4.74% expected return) is not efficient, as we can replace it with any point in the upper half and have higher return with the same level of risk.

With the EF, investors can invest their money into the market with the expectation that their portfolio offers the highest expected return possible. Investors who want higher expected return can move along the EF line to the right, as it offer higher expected return by taking higher risk.

2.3 Sharpe Portfolio and Minimum Variance Portfolio

In Figure 1, there are two important portfolios. The first is Sharpe Portfolio, which have expected return of 7.02% and STDEV of 4.47%. The Sharpe Portfolio, by our own definition in this work, is the portfolio with the highest Sharpe ratio. William F. Sharpe introduced the ratio in 1966, and later revised it in 1994 with the name Ex Ante Sharpe Ratio [8]:

$$\text{Sharpe ratio} = \frac{\mathbb{E}(R_F - R_B)}{\sigma_F} \quad (6)$$

where R_F is return on portfolio F, R_B is return on benchmark (risk free rate) and σ_F is STDEV of portfolio F (with B is risk free rate).

Based on the above formula, the Sharpe ratio measure the expected excess return (portfolio return minus risk free rate) per unit of risk. The Sharpe Portfolio in Figure 1 have the highest Sharpe ratio of 1.4841 with risk free rate at 0.3844%, which mean for every unit of risk taken at that point, investor will be expected 1.4841 unit of excess return. As we move long the EF, the further away we get from the Sharpe Portfolio, the lower the Sharpe ratio we get, and lower excess return per each unit of risk we take. As different investors have different risk profiles, we cannot set a predetermined level of risk for our portfolio during our work. Therefore, we will base our thesis on the Sharpe portfolio, and investors will have option to adjust their risk level to suit their need in the final software.

The second important portfolio in Figure 1 is the Minimum Variance Portfolio (expected return: 4.74%, STDEV: 3.49%, Sharpe ratio: 1.25). It is the portfolio on the EF line that have the lowed STDEV among all possible portfolio. This portfolio would be the most suitable one for low risk tolerance investors.

2.4 Criticisms

In Modern Portfolio Theory, return, risk and other related parameters are calculated based on historical data. Therefore, the prediction power of the model is based on historical events. However, if there is a new event happen in future that never happened

before, such as the global coronavirus pandemic in 2020, then the model will fail to evaluate that event. One common warning in finance research report is that past performance is not a guarantee of future results, and Modern Portfolio Theory is using past performance to calculate portfolio expected return. In their research in 1991, Best and Grauer found that a small change in assets' expected return will result in a significant change in portfolio allocation [9]. Therefore, any small error in the estimation of expected return will lead to a notable different in investment performance.

Another criticism of Modern Portfolio Theory is that it measures risk in term of volatility. However, many investors don't view STDEV as risk. Warren Buffett, a famous and successful investor, once said that his partner and him would "much rather earn a lumpy 15% over time than a smooth 12%" [10]. For him, and for many investors, volatility is a part of stock investment and should not be a negative point when investing. This can be improved by using semivariance to measure the volatility when stock return is lower than expected instead of variance. We will discuss this in chapter 4.1.

2.5 Black-Litterman model

Among criticisms mentioned in chapter 2.4, using historical data to predict future return is rationally the most important point, as any small error would result in a critical different in the result. To address this problem, Fischer Black and Robert Litterman developed a model, named the Black-Litterman (BL) model. The BL model combine the expected return of a diversified market portfolio with the investor's subjective view to form a investors' expected return instead of the traditional expected return in Modern Portfolio Theory [11]. First, the concept of equilibrium expected return, the level of return that adequately compensate investors for their expected risk, was discussed. The BL model assumes that there are equilibrium expected returns in the market. Each asset has their own value of equilibrium expected return. And combine with their equilibrium weight, they form the equilibrium market portfolio that reflect the investors' expectation of the capital market. By solving for the implied equilibrium return, and include investors' view into the model, we get market's expected return, instead of historical expected return, for our model.

The BL model require several steps to find the investors' view-adjusted market equilibrium returns [12]:

- Determine market portfolio and equilibrium market weight. Calculate covariance matrix for all assets. Market portfolio, and subsequently market weight is based on the investors' investment geography. For example, an investor invest in United States equity only will consider all listed equities in United States as his market portfolio, and all equities weight, determined by their market capitalization, in that portfolio as equilibrium market weight.
- Perform reverse optimization process. From covariance matrix and asset weight back solving to get the equilibrium expected return. The reverse formula with m assets in the portfolio is:

$$\Pi = \begin{bmatrix} r_1 \\ \vdots \\ r_m \end{bmatrix} = \lambda \times \begin{bmatrix} Cov_{11} & \cdots & Cov_{1m} \\ \vdots & \ddots & \vdots \\ Cov_{m1} & \cdots & Cov_{mm} \end{bmatrix} \times \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix} \quad (7)$$

where Π is the Implied Excess Equilibrium Return Vector, λ is the risk aversion coefficient, calculated by the dividing portfolio excess return by portfolio risk

$$\lambda = \frac{(r_p - r_f)}{\sigma_p^2} \quad (8)$$

- Set view and confidence for each view. Even though this is an important step in BL model, it requires to set the view without knowing the actual result of the market in future. This is not suitable for back testing process, where the actual results are known, and will be skipped during model improvement.
- Calculate the view-adjusted market equilibrium return.
- Solve the optimization model and obtain the EF.

The BL were tested in practice and showed that it can help mitigate the problem of over sensitive in estimation [12]. Therefore, we added it into our program as an option for investors to choose for their asset allocation decision.

3 Implementation

While EF is easy to use in investment decision, calculating it is more complex in real world. When we find a good return-risk combination from our assets, we need to assure that return-risk combination is the best we could have. The calculation requires a long computing time, as we need to calculate a list of assets with their weight in the portfolio, get our return-risk combination, and then repeat the process until we are confident that our result is optimal.

3.1 Programming Language and Libraries

Initially, we used Monte Carlo simulation to obtain the EF. By generating a large set of randomized portfolios, we can reasonably assume that the set represents all possible portfolios. All the optimal portfolios from that set, i.e. portfolios with highest expected return for a level of risk, will form the EF line. Monte Carlo simulation also can be used to address the lack of data in the model. The general market condition during our dataset is bullish, without a long bearish time like the 2008 financial crisis. As a result, we could not test our model during unfavorable economic environment. With Monte Carlo simulation, we can produce randomly hundreds or thousands of different scenarios, with many modifiable variables [13]. This is the main advantage of Monte Carlo simulation against portfolio optimization method. Monte Carlo also can be used as a complement to convex optimization model, as it offers the input of practical issues into the model, such as different tax bracket, which are often difficult using mathematic formula. However, during our work, we found that Monte Carlo simulation was inefficient. Monte Carlo simulation spend a significant amount of resource to generate non-optimal portfolios, while our target is only optimal portfolios. As mentioned in the beginning of this thesis, a quantum computer will make the wasted resources become insignificant and Monte Carlo simulation could become a good method due to its mentioned advantage. Nevertheless, with current technology, it was more efficient to use a method that actively seeks the optimal portfolio.

In the end, we choose convex optimization to solve for the optimal portfolio. And we choose Python as programming language. Python has NumPy library [14] and pandas library [15], those are two powerful libraries for data processing and matrix manipulation,

which is important in calculating the EF. We use pandas' DataFrame class to store our data, especially historical data and matrixes (correlation, covariance, variance). The most important function we used from NumPy is matmul, which is matrix multiplication. As we discussed in chapter 2, to calculate covariance, variance and BL model, we will need NumPy's matmul. It works with panda's DataFrame class; therefore, we do not need to change the data format. Python also has CVXPY library [16], an open-source library which is our main tool to solve convex optimization problem in this thesis.

One main disadvantage of Python is its speed performance. Unlike C or C++, Python is an interpreted programming language, and must spend time translate code before executing it. However, Python libraries: NumPy and pandas, are optimized to perform faster when handle big data, therefore it can improve speed performance when executing mathematic related functions. Moreover, due to the simplicity of the language, and the availability of supporting library, development speed of program using Python is much faster. For example, the code to calculate monthly return for 75 ETFs for 1200 months randomized price value is listed below:

```
import time
import pandas as pd

# Measure execution time using time.time() function
# Method 1: Do not use any Python specific functions
# Monthly price data is store in a list: priceList

for stockCode in priceList:
    temp = list(stockCode)
    stockCode = list(stockCode)    # To make sure stockCode is a list
    temp.append(temp.pop(0))
    tempList = list([])
    for price in stockCode:
        tempList.append(price/temp[stockCode.index(price)] - 1)
    returnList.append(tempList)

# Program execution time: 0.2862522602081299 seconds
#-----

# Method 2: Use Python zip() function
# Monthly price data is store in a list: priceList

for stockCode in priceList:
    temp = list(stockCode)
    temp.append(temp.pop(0))
    returnListZip.append([x/y - 1 for x, y in zip(stockCode, temp)])

# Program execution time: 0.05186128616333008 seconds
#-----

# Method 3: Use Python Pandas library
# Monthly price data is store in a DataFrame: priceDF
```

```
returnDataDF = priceDF.pct_change(-1)
# Program execution time: 0.01795196533203125 seconds
```

Listing 1. Different methods to calculate monthly return

The first method to calculate monthly return is to loop through the list of all stock data, and then perform another loop to going through all the monthly price data and divide current price by previous month price, then minus 1 to get the monthly return. This method is done without any specific Python function. Therefore, it can be performed similarly in other programming language. It also has the longest execution time. The second method used the same concept. It, however, used Python zip function to avoid two loops function. By using the optimized function in Python instead of a general loop, its performance speed is at least 5 times faster. Both methods, however, is complex when compared with the third method using pandas' DataFrame. The third method only requires a single line of code to call the pandas' pct_change function. It is also the fastest method despite being the simplest. Obviously, using pandas' DataFrame is the best option in this situation.

To develop Graphic user interface (GUI) for the program, we use Tkinter, and open-source library that is included with install of Python [17]. Compared with PyQt, another popular GUI framework for Python, Tkinter has two advantages that convinced us to choose it for our software development:

- Tkinter is included in Python, as a result, when PyInstaller package the program as an executable, it will not need other dependencies or configurations for Tkinter.
- It's simple to understand and faster to develop.

The main disadvantage of Tkinter is due to its simplicity, it does not have advanced widgets and does not have a native look interface. However, this disadvantage does not affect our program, as users require a simple GUI.

Because the target user group of our program is investors, not programmers, they cannot open a python file. Instead, PyInstaller will package all python files into a standalone executable, which can be run in Windows, GNU/Linux or Mac OS X [18].

3.2 Portfolio Construction

3.2.1 Equity construction

Before we can compute the EF line, we first must have a list of assets that we can invest in real world. Ideally, we would want to include all countries and their equity markets in our portfolio to form a global portfolio. However, there were several real-world problems that limit us in building the ideal global portfolio.

Firstly, all assets in our portfolio must be easily traded in stock exchange. Therefore, we can't use indexes to build the portfolio. Instead we decided that we can use relevant ETF, which underlying value is the index itself. For example, we can use S&P 500 Index to represent United States' equity market. However, we cannot invest in S&P 500 Index directly and as an alternative use SPDR S&P 500 ETF Trust (SPY), an ETF that tracks S&P 500 Index, to build our global portfolio. ETF also have advantages that make them more attractive suitable for pension fund investment, such as lower cost than traditional mutual funds [19], easy exposure to specific industry by invest in industry specific ETF, or flexibility in trading.

Secondly, since we choose ETF to build our portfolio, we have a problem of data availability. To calculate the expected return and STDEV, we use 5 years data to calculate expected return and STDEV. The 5 years data range is subjective and can be changed based on investors situation. Furthermore, we need to backtest the investment strategy to see if how it performed during the past, through bull market and bear market. If possible, we should backtest it during the 2008 financial crisis and the 2000 dot-com bubble. However, that would require a minimum of 25 years of historical data. To calculate the actual portfolio performance in April 2000, we would need historical data from April 1995 to March 2000 (5 years historical data) to solve for the optimal portfolio weight, then multiply the weight with the actual return of each assets in April 2000 to get the actual return of our portfolio. In our list, only SPY have data available in 1995, and only 33 ETFs

represent big economic have data available in 2003 to backtest the 2008 market. There was a tradeoff between the number of ETFs available in portfolio, and the range of data for backtesting. In the end, we set the cutoff date to be February 2010 and have a total of 75 ETFs to form the global portfolio. While most of the core ETFs in our portfolio fulfill that requirement, we had to remove many other ETFs that did not have enough data at the time we construct our portfolio, including but not limit to: RSX (VanEck Vectors Israel ETF), NORW (Global X MSCI Norway ETF), EDEN (iShares MSCI Denmark ETF), EFNL (iShares MSCI Finland ETF), and LEMB (iShares J.P. Morgan EM Local Currency Bond ETF).

The last problem showed up when we first run the model. If two assets are highly correlated with each other (correlation coefficient close to 1), and one asset have better return-risk combination, Modern Portfolio Theory will always prefer that asset in the EF. As a result, sometimes we would end up with the Sharpe Portfolio with 20% allocation to countries with small Gross domestic product (GDP) like Vietnam or Egypt, while only 5% allocation to United States, which have GDP about 81 times of Vietnam or Egypt. Even though the allocation could be reasonable in term of maximizing return, it was not what we wanted for a global, diversified pension investment. Allocating a significant amount of portfolio to a single country will increase our expose to country risk. Country risk is commonly referred to the risk of foreign government declare bankruptcy, and refused to pay back its debt, but also referred to political risk, exchange-rate risk or economic unrest. The target of a global pension portfolio is to reduce a specific risk in a country, therefore expose to country risk is not desirable. To solve this problem, we included a maximum allocation for each EF, based on the 2018 GDP of its underlying country [20].

Table 1. List of equity ETF in portfolio and their maximum allocation

Country	2018 GDP (billion USD)	% GDP in portfolio	ETF	Maximum allocation
United States	20,544	28.7%	SPY, IWM, IWF, VTV, VIG, XLF, XLK, XLE, XLV, XLY, XLI, XLP, XLU, XBI, XLB, XHB, XME	35.0%
China	13,608	19.0%	HAO, CHIQ, FXI, CQQQ, CHIX	20.0%
Japan	4,971	6.9%	EWJ, SCJ	8.0%
Germany	3,948	5.5%	EWG	6.0%

United Kingdom	2,855	4.0%	EWU	4.0%
France	2,778	3.9%	EWQ	4.0%
India	2,719	3.8%	INDY	4.0%
Italy	2,084	2.9%	EWI	3.0%
Brazil	1,869	2.6%	EWZ	3.0%
Canada	1,713	2.4%	EWC	2.5%
Russia	1,658	2.3%	RSX	2.5%
South Korea	1,619	2.3%	EWY	2.5%
Australia	1,434	2.0%	EWA	2.0%
Spain	1,419	2.0%	EWP	2.0%
Mexico	1,221	1.7%	EWW	2.0%
Netherlands	914	1.3%	EWN	1.5%
Turkey	771	1.1%	TUR	1.5%
Switzerland	705	1.0%	EWL	1.0%
Taiwan [21]	590	0.8%	EWT	1.0%
Sweden	556	0.8%	EWD	1.0%
Belgium	543	0.8%	EWK	1.0%
Thailand	505	0.7%	THD	1.0%
Austria	455	0.6%	EWO	1.0%
South Africa	368	0.5%	EZA	1.0%
Hong Kong	363	0.5%	EWH	1.0%
Singapore	364	0.5%	EWS	1.0%
Colombia	331	0.5%	GXG	1.0%
Chile	298	0.4%	ECH	1.0%
Egypt	251	0.4%	EGPT	1.0%
Vietnam	245	0.3%	VNM	1.0%
European market	N/A	N/A	VGK, DFE, FDD, IEUS	4.0%
Emerging market	N/A	N/A	EEM	3.0%
Total	71,699	100.0%	56 ETFs	123.5%

From Table 1, we can see that the total GDP cover by countries in our portfolio is 71,699 billion USD. According to World Bank data, global GDP at 2018 is about 85,910 billion USD [20]. Our portfolio represents 83.5% of the world GDP, and exposures to the big market in North America, Europe, East Asia and emerging markets. Two countries with

significant allocation are United States and China which have a stable political and economic environment. In our current model, we use a portfolio with 60% maximum allocation to equity, with the total maximum allocation of 123.5%. This allow our software to choose among 56 ETFs, with 123.5% allocation to find the optimal 60% allocation for the portfolio. The maximum allocation put a limitation on the software but leave enough room for optimization.

3.2.2 Bond and Alternative Investment Construction

For bond ETF, we could not use the same method as equity ETF, as the number of bond ETF satisfy our condition of 10 years data is small. Also, bond ETF that cover only one country is not popular, especially in emerging countries. Bank for International Settlements estimated that at the end of September 2019, United States' total debt outstanding is about 39% of the total outstanding debt [22]. And because United States debt market is dominating the world, we decided to mostly select United States bond ETF for our portfolio. Another reason is due to the low return of bond ETF, the selection of bond will not make a big different compared with equity.

Table 2. List of bond and alternatives ETF in portfolio and their maximum allocation

Code	Class	Region	Type	Maximum allocation
VGSH	Bond ETF	North America	Government 1-3Y	4.0%
IEI	Bond ETF	North America	Government 3-7Y	4.0%
IEF	Bond ETF	North America	Government 7-10Y	4.0%
HYG	Bond ETF	North America	USD High yield	4.0%
LQD	Bond ETF	North America	Investment Grade Corp	4.0%
AGG	Bond ETF	North America	Investment Grade Govt	4.0%
EMB	Bond ETF	Emerging	USD	4.0%
BIV	Bond ETF	North America	Intermediate term Corp	1.5%
TLH	Bond ETF	North America	Government 10-20Y	6.0%
TIP	Bond ETF	North America	TIP	4.0%
BSV	Bond ETF	North America	Short term	1.5%
MBB	Bond ETF	North America	MBS	1.5%

BWX	Bond ETF	Non-US	Government	2.5%
IEAC	Bond ETF	Western Europe	Investment Grade Corp	1.5%
USO	Alternatives	Global	WTI Crude Oil	3.0%
GLD	Alternatives	Global	Gold	3.0%
SLV	Alternatives	Global	Silver	1.0%
NIB	Alternatives	Global	Cocoa	1.0%
VNQ	Alternatives	North America	Real estate	1.2%
Total				55.7%

Most of bond ETF in Table 2 are investment grade and government bond, except HYG and EMB, as a result their expected return is low to trade off for their lower risk. Therefore, their allocation will not impact portfolio performance significantly. Adding alternatives investment such as crude oil or gold would also benefit portfolio diversification. For example, at February 2020, GLD (gold ETF) has a correlation of -0.08 with SPY (S&P 500 ETF). Negative correlation implies that adding GLD and SPY together in portfolio will lower portfolio's variance and STDEV. At the end of 2019, there were 6 robo investment accounts have miscellaneous (alternatives) investment among 64 taxable accounts reported by Backend Benchmarking [3].

3.3 Program flowchart

Bloomberg provided all historical data for our calculation. They have an API which allow us to easily extract and processing data. In our work, two most important feature of Bloomberg API is they can make price adjustment for dividend, and handle holiday different between countries. For example, Japan have Golden Week holiday, which is national holiday from 29 April to early May. Tokyo Stock Exchange will be closed during the Golden Week, therefore, any ETF listed in Tokyo Stock Exchange will not have historical data available during that time. The holiday problem is critical if we use daily or weekly historical data for the model but is less important if we use monthly data.

The flow of our program is illustrated by Figure 2 on page 18. Right after loading raw data from Bloomberg, our program will process them to remove any not applicable (N/A) data and convert data from string type to suitable type. There's a function running in the background to interpret received data and converted it to reasonable value. For example,

when edit target STDEV, if user entered number 12, program will understand that user want to enter 12% or 0.12 and will store the entered value as 0.12. Data is then displayed in program's GUI. Then at step 3, model is updated with new parameters from user input to make sure program is working with the latest information.

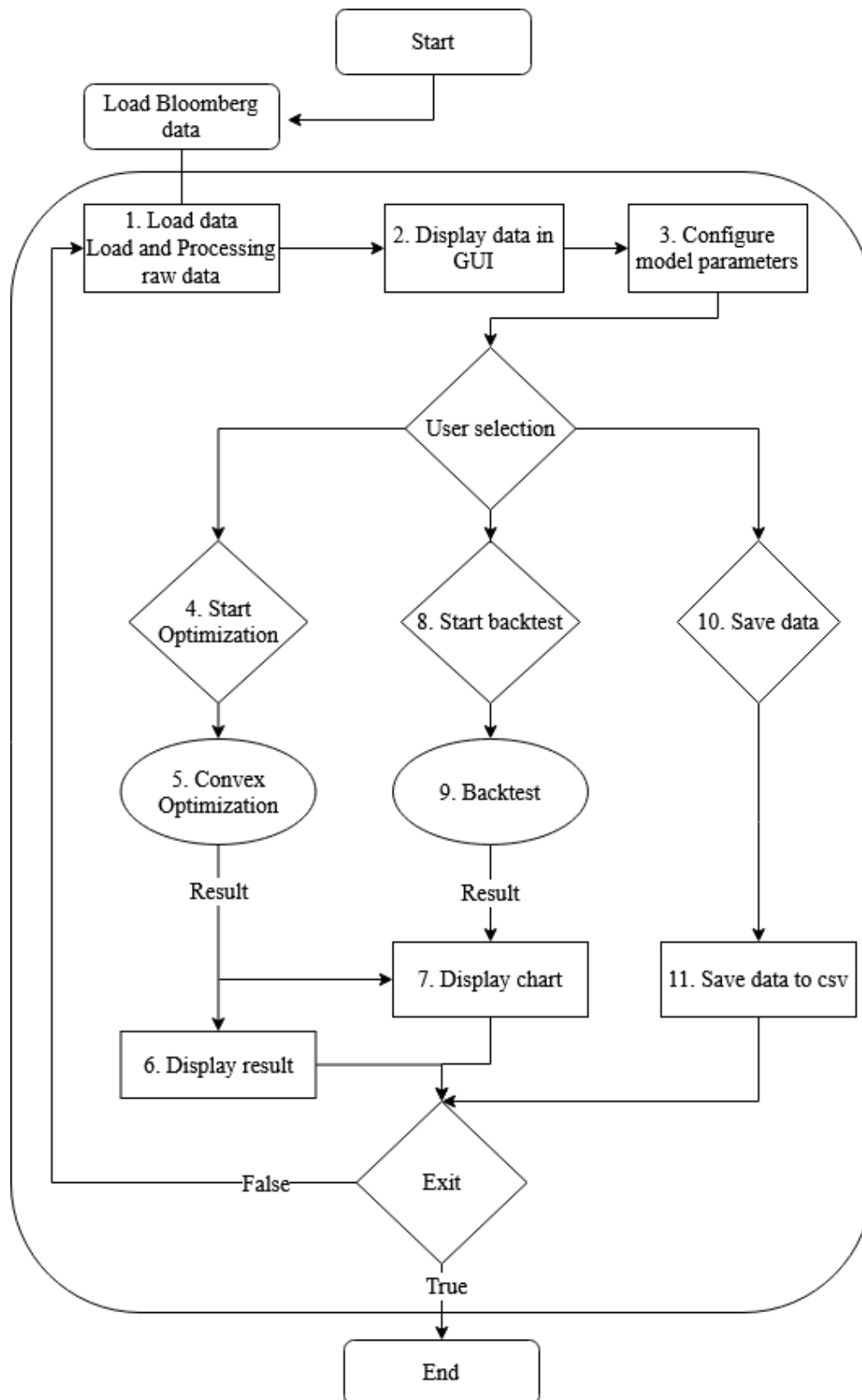


Figure 2. Program flowchart

After processing and editing data, user can start using program to solve for the EF (step 4: start optimization) and backtest the current portfolio (step 8 - start backtest). In each of those steps, program will perform the calculation in the background and then output the relevant chart. The chart will be the EF chart similar to Figure 1 in page 5, if user chose optimization, or the cumulative return of portfolio if user chose backtest.

Finally, program also have option to save data as csv file format. The saved data are useful in case user want to save the result of the optimization or want to share the output with clients or use it as input for other software. Save function is useful in backtest process, when users need to test their model for monthly performance from March 2015 onward. Depend on the precision setting, such backtest would take from 1 to 4 hours to complete and saving function would help users avoid running the same test repeatedly.

3.4 Program user interface

The target users of this program are investors, who familiar with economic concept but may not skilled in technology. Therefore, the program was developed with a simplified user interface. Figure 3 demonstrated the main interface when users open the program:

1							3			
Save		Calculate		Backtest						
No	2 Ticker	Class	Return	Stdev	Min	Max	8.49%, 8.84%	8.06%, 8.50%	8.63%, 9.00%	9.31%, 9.99%
1	Cash	Cash	0.0000	0.0000	0	0.0	0.00%	0.00%	-0.00%	0.00%
2	Bond ETF A	Bond ETF	0.0126	0.0098	0	0.15	0.00%	0.00%	0.00%	0.00%
3	Bond ETF B	Bond ETF	0.0253	0.0271	0	0.15	10.00%	15.00%	10.00%	0.00%
4	Bond ETF C	Bond ETF	0.0262	0.0220	0	0.1	0.00%	7.00%	0.00%	0.00%
5	Bond ETF D	Bond ETF	0.0236	0.0357	0	0.1	0.00%	0.00%	0.00%	0.00%
6	Bond ETF E	Bond ETF	0.0500	0.0522	0	0.1	10.00%	0.00%	10.00%	10.00%
7	Bond ETF F	Bond ETF	0.0433	0.0521	0	0.1	0.00%	-0.00%	0.00%	0.00%
8	Bond ETF G	Bond ETF	0.0506	0.0600	0	0.1	0.00%	0.00%	0.00%	10.00%
9	Alternative A	Alternatives	0.0993	0.1429	0	0.8	2.08%	0.00%	4.28%	10.00%
10	Alternative B	Alternatives	0.0380	0.1359	0	0.08	7.92%	8.00%	5.72%	0.00%
11	Alternative C	Alternatives	-0.0450	0.1859	0	0.06	-0.00%	0.00%	-0.00%	0.00%
12	Equity ETF A	Equity	0.1071	0.1180	0	0.2	20.00%	20.00%	20.00%	20.00%
13	Equity ETF B	Equity	0.1494	0.1500	0	0.15	15.00%	15.00%	15.00%	15.00%
14	Equity ETF C	Equity	0.0273	0.1280	0	0.1	10.00%	10.00%	10.00%	6.77%
15	Equity ETF D	Equity	0.0538	0.1209	0	0.1	10.00%	10.00%	10.00%	10.00%
16	Equity ETF E	Equity	0.0526	0.2094	0	0.1	0.00%	0.00%	-0.00%	3.23%
17	Equity ETF F	Equity	0.0320	0.1702	0	0.05	5.00%	5.00%	5.00%	5.00%
18	Equity ETF G	Equity	0.0203	0.1525	0	0.08	0.00%	0.23%	0.00%	0.00%
19	Equity ETF H	Equity	0.1880	0.1867	0	0.1	10.00%	9.77%	10.00%	10.00%

Figure 3. Program main interface

The main interface has three parts. The part labeled with number 1 is buttons related to step 10 (save data), step 4 (start optimization) and step 8 (start backtest) in Figure 2 page 18. The second part is where user can enter or edit their preferences. For example, user who want to take less risk can change the minimum and maximum allocation of cash in their portfolio, any change in user's allocation will be reflected in the calculated result. The part labeled 3 is the output of the EF result. The first column represents the allocation of the highest Sharpe ratio portfolio. The remaining three columns on the right are the allocation that would result in user's target STDEV. In this case it would be 8.5%, 9% and 9.99% from left to right.

Bond ETF A	Bond ETF B	Bond ETF C	Bond ETF D	Bond ETF E	Bond ETF F	Bond ETF G	Alternative A	Alternative B	Alternative C	Equity ETF A
1.0	0.920043591	0.783298224	0.691834348	0.63308381200	-0.030872426	0.37542844700	0.241874277	0.470813999	-0.251651746	-0.339552911
0.920043591	1.0	0.86737643400	0.745097067	0.69580557	-0.10302689500	0.383606127	0.292517142	0.535119097	-0.30896868000	-0.380532052
0.783298224	0.86737643400	1.0	0.738351322	0.709555906	0.05917438400	0.49798239200	0.36750080700	0.401333196	-0.24605143100	-0.17600619
0.691834348	0.745097067	0.738351322	1.0	0.78987072	0.301132591	0.627638618	0.488360442	0.571481303	-0.14559556	0.0029778
0.63308381200	0.69580557	0.709555906	0.78987072	1.0	0.43491281290	0.77192806	0.595823686	0.51543843290	-0.139250373	0.158038433
-0.030872426	-0.10302689500	0.05917438400	0.301132591	0.43491281290	1.0	0.645650066	0.416329478	0.21300685800	0.34715475700	0.721105478
0.37542844700	0.383606127	0.49798239200	0.627638618	0.77192806	0.645650066	1.0	0.390352387	0.477091781	0.08538074300	0.327807013
0.241874277	0.292517142	0.36750080700	0.488360442	0.595823686	0.416329478	0.390352387	1.0	0.03642692	0.00180251390	0.473697763
0.470813999	0.535119097	0.401333196	0.571481303	0.51543843290	0.21300685800	0.477091781	0.03642692	1.0	0.037145204	-0.18651068100
-0.251651746	-0.30896868000	-0.24605143100	-0.14559556	-0.139250373	0.34715475700	0.08538074300	0.00180251390	0.037145204	1.0	0.35815256
-0.339552911	-0.380532052	-0.17600619	0.0029778	0.158038433	0.721105478	0.327807013	0.473697763	-0.18651068100	0.35815256	1.0
-0.297244515	-0.283915513	-0.082780644	0.05359979400	0.138157663	0.627603644	0.311896446	0.383543636	-0.18433224	0.232804747	0.921039185
-0.216737016	-0.260716255	-0.073622083	0.10810357400	0.26187528	0.706473419	0.49500081790	0.285290357	-0.069002735	0.23032749600	0.78141701590
-0.364656992	-0.341203076	-0.148527304	0.041366716	0.131547626	0.655209081	0.39957034790	0.31801573	-0.181253971	0.296998943	0.784672155
-0.214521844	-0.19096762800	-0.025040195	0.135459215	0.220899464	0.586209038	0.436321866	0.276259578	0.05826401290	0.47227361790	0.69833209400
0.137970124	0.202608117	0.26628387	0.32091863600	0.36838924700	0.251905304	0.3202121563	0.421576066	-0.026232952	0.011231728	0.34101304290
-0.082215863	-0.104948628	0.054472278	0.289829222	0.366843782	0.660569027	0.626884333	0.310005001	0.183359092	0.440195231	0.67711098200
-0.3036137000	-0.269598739	-0.12387285100	0.05409210290	0.136987245	0.613657341	0.36944137	0.215331582	-0.081024733	0.271646424	0.819601413

Figure 4. A screenshot of program's correlation matrix

Program also displays correlation matrix of all portfolio's assets (see Figure 4). To improve user's ease of use, each matrix is colored based on their value. With the green color (hex color code #33cc33) for any value close to 1, the green color change to lighter green and to lighter yellow (hex color code #fff0b3) when correlation fall below 0.6. Eventually, the cell will change to red (hex color code #ff3333) when correlation get closer to 0 or fall below 0. Program also have a horizontal scroll bar (not shown in figure) in case the correlation matrix is too long to view in one screen.

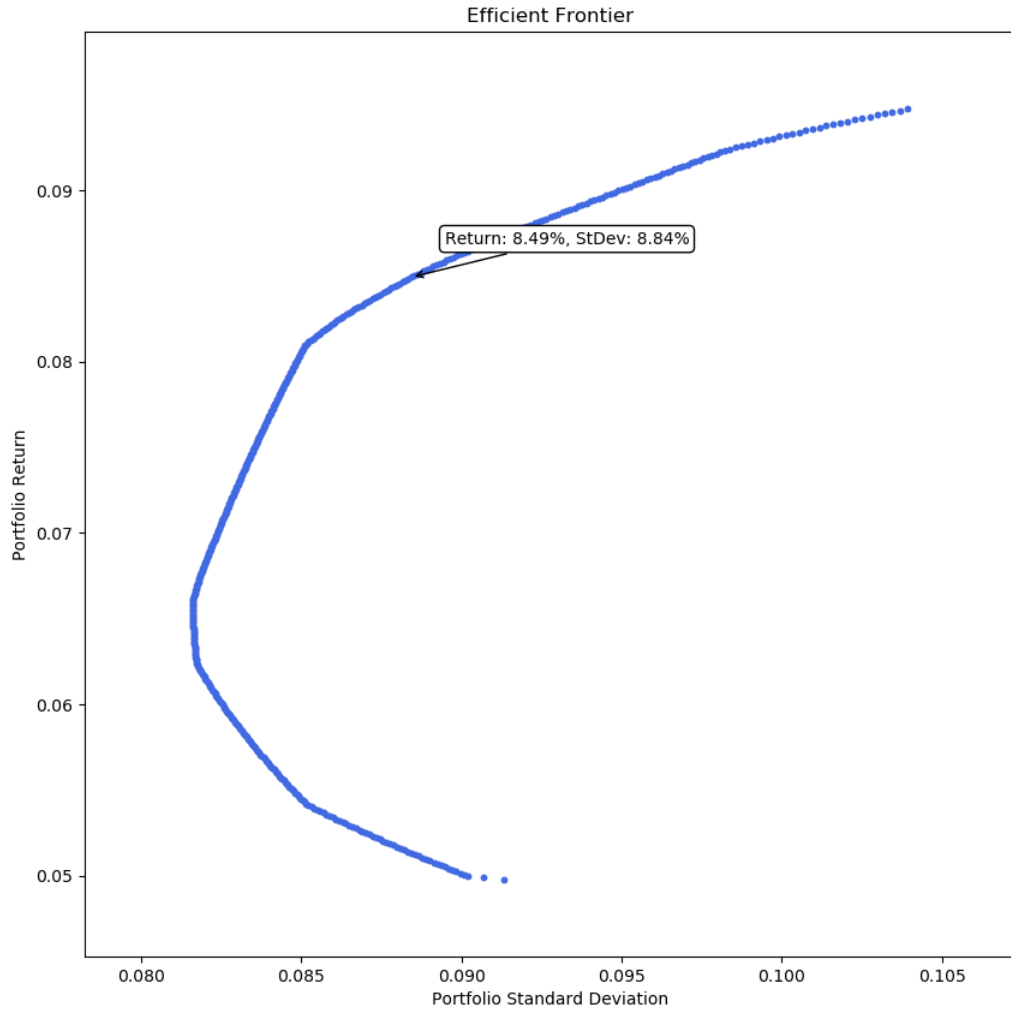


Figure 5. The Efficient Frontier result from optimization

After user click on the calculate (optimization) button, program will use available data to calculate the EF and related allocation. Aside from output the result in the main interface in Figure 3, program will also output the EF chart for user (see example in Figure 5). The chart was drew using matplotlib library, and it is an interactive chart. It keeps track of user's mouse hover action and will display the portfolio expected return and STDEV whenever user hover the mouse over any dot in the EF. The chart help user to decide which portfolio to invest in. For example, in Figure 5, if a user prefers a higher expected return portfolio, he or she can move the mouse to the top right and select a portfolio with 9% expected return and about 10% STDEV.

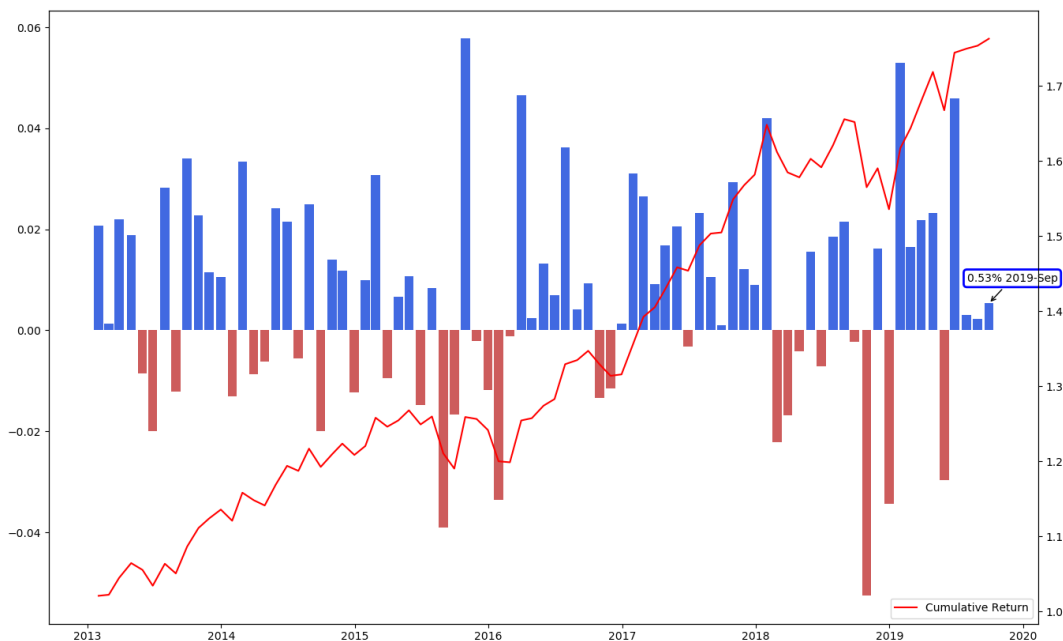


Figure 6. Chart display backtest result

Another feature that would help user in his investment decision making is backtest. Figure 6 showed an example of the result of the backtest function. From this chart, user can easily see the monthly performance of the portfolio in the past, as well as the cumulative result if he has invested 1 dollar at the beginning of the period. The length of the period is depending on the available data the program can get. The return from backtest is net return, which means it also includes management fees and any transaction costs. The chart is also an interactive chart, which will display the actual return and month of any column when user hover mouse over it.

4 Improvement

The EF provides the solution for the most basic target of investment: optimized portfolio allocation to maximize return and minimize risks. Also, the Modern Portfolio Theory was

accepted as the foundation for portfolio theory since its first publication in 1952. Nevertheless, the traditional Markowitz' theory was not commonly used in practice [23]. In our model, we set the constraint of asset's maximum allocation, partially to address one of the flaws of the Modern Portfolio Theory: extreme weight allocation with a small change in expected return. In this chapter, we added more improvements for our program to improve its practical use.

4.1 Semivariance

One of the main criticisms of the EF is that it uses STDEV as the measurement of risk, whereas investors in practice define risk as the probability of actual return less than the expected return. When the market is on downtrend, both STDEV and investors' definition of risk are similar. However, when the market is on uptrend, investors will not consider the stock price moving up as risk, while STDEV will measure the upside potential as risk. Therefore, to reflect the investors' view better in the EF, instead of using STDEV and variance as measurement of risk, we should use semi-STDEV and semivariance to measure only downside risk. Unlike variance, semivariance only measures the variety of stock return when it is less than the expected return. Markowitz also agreed that it is more reasonable to replace variance with semivariance in his model [24, p. 476].

To implement this improvement and allow users to change between variance and semivariance with ease, a separate function was created:

```
def calculateStdev(returns, rfr=0.0, isDownside=False):
    """
    Determines the Stdev of the portfolio.

    Parameters
    -----
    returns: :py:class:`pandas.Series`
        Daily returns of the strategy, noncumulative.
        Return is in form of r% (not 1 + r%)
    rfr: :class:`float`, optional
        minimum acceptable return, expected return
    isDownside: :class:`bool`, optional
        If True, will calculate downside risk instead of STDEV

    Returns
    -----
    :py:class:`pandas.Series`
```

```

    """ Stdev of the portfolio
    """

    if (isDownside):

        print('Calculating semivariance instead of variance')
        downsideReturn = (returns - rfr)
        mask = downsideReturn >= 0
        downsideReturn[mask] = 0
        return downsideReturn.std()

    return returns.std()

```

Listing 2. Python function to calculate STDEV or semi-STDEV (downside risk) depend on user's input

For our case, risk free rate was chosen as expected return. If user selects semivariance in setting, semi-STDEV will be calculated instead, and any return higher than expected return will be ignored in the STDEV calculation. Then program will use the result of semi-STDEV to calculate portfolio semivariance in next step. We backtested the semivariance model from March 2015 to March 2020, using 5 years historical data from February 2010 and obtained the result in Figure 7:

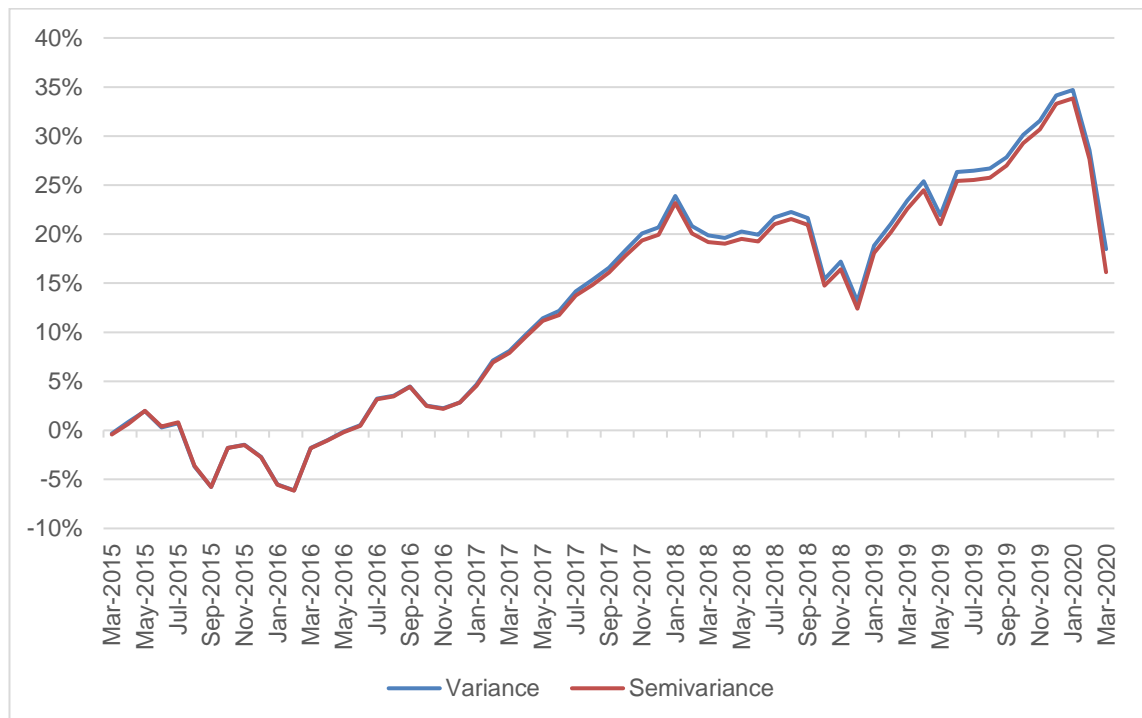


Figure 7. Cumulative gross return of Sharpe portfolios using variance and semivariance

From the backtest result above, we can see that variance and semivariance portfolio have similar performance. And variance portfolio in fact has higher return than semivariance portfolio. Even when excluding the February 2020 and March 2020, two outlier months when the global coronavirus pandemic happen, variance portfolio has 34.697% return, which is 0.86% higher than semivariance portfolio. There were a few months that semivariance portfolio performed slightly better than its counterpart, and there were months, especially during 2018 and 2019, that we can clearly see variance portfolio was better. Two portfolios have performed nearly identical to each other during the tested period, with monthly correlation was 0.99816. Variance portfolio, in this case, is as practical as semivariance. Refer to Markowitz's article about semivariance, he also mentioned that he found mean-variance model can provide an sufficient estimate for expected return as mean-semivariance [24].

Another study of semivariance were published by Aleksei Tcysin and Pivovarov Dmitrii from Lapland University of Applied Sciences [25]. Their work was studied to has better understanding of the benefit of semivariance in Modern Portfolio Theory. In their thesis, author studied the performance of semivariance model and compared it with variance model and SPY ETF (tracking the S&P 500 Index). They tested the model with three different environments: bull market (2014), bear market (2008) and sideway market (2015) and the results show that while semivariance portfolio outperform variance and SPY in 2014 and 2015, but it underperformed in 2008. The authors concluded that while the combination of semivariance and Bayes-Stein estimators generally produce superior results, they cannot identify the source of the exceed return is semivariance or Bayes-Stein estimator framework. They also noted that the model needs further testing and research before putting in practice.

There were several differences in Tcysin and Dmitrii model when compared with the model used in this thesis, which could explain the difference. First, the authors used Bayes-Stein estimators, while we did not. Second, they used 30 stocks representing Dow Jones Industrial Average for testing, which is a 100% equity portfolio, while our portfolio has a maximum allocation of equity of 60%. Lower allocation to equity would result is lower volatility and as the result the different between variance and semivariance portfolio would be smaller. Our portfolio also has maximum allocation for each ETF, which increases the diversification in portfolios and lowers their volatility. Another point is that

Tcysin and Dmitrii tested their model with different level of risk tolerance, and the higher the risk tolerance, the larger the difference. With risk tolerance of 0, the difference in return between variance and semivariance was lowered to 2.1% in bullish market. And finally, instead of testing only for one market condition, we test 5 years performance from March 2015 to March 2020, during that period two portfolios experienced bull market, bear market and sideways market. The difference in performance in bull market condition could be offset by bear market followed it.

Nevertheless, semivariance improvement added to the program could help users improve their portfolio performance. Users, however, need to understand that semivariance portfolio does not always outperform variance portfolio. Also, they need to perform their own backtest research, which is also available as a function in the program, thoroughly with their target return and risk to see which model performed better.

4.2 Black-Litterman model

In our model, equity portfolio is defined as all ETFs listed in Table 1, bond portfolio and alternatives investments are listed in Table 2. The BL's market portfolio is the portfolio formed by combine 60% equity portfolio and 40% bond and alternatives portfolio. The ratio 60%/40% is the constraint in our pension investment strategy during the backtest process. The market portfolio, regardless of its allocation, also follow the same constraint. While the market capitalization data of each ETF is available in Bloomberg, their historical data is not. For this reason, we used the fund total assets data instead. Because ETFs are designed to track their underlying assets, the different between ETFs' market capitalization and their total assets is negligible. We can calculate matrix multiplication using NumPy's `matmul` function. Listing 2 show a simplified version, without debug code and error handling, of the program's Black-Litterman model.

```
import numpy as np
import pandas as pd
def blackLitterman(cagr, stdev, correlation, rfr):
    """
    Calculate Black-Litterman model's implied equilibrium return

    Parameters
    -----
    cagr: :class:`pandas.Series`
        Expected return of all assets in portfolio
```

```

stdev: :class:`pandas.Series`
    Stdev of all assets in portfolio
correlation: :class:`pandas.DataFrame`
    Correlation matrix of all assets on portfolio
rfr: :class:`float`
    risk free rate

Returns
-----
:py:class:`pandas.Series`

    """
    Implied equilibrium returns matrix from Black-Litterman model
    """
    # Calculate market weight matrix using a helper function
    portWeight = calculateMarketCapWeight()

    # This code helps match each asset with its return, stdev and weight
    dataBL = pd.DataFrame({'Return':cagr, 'Stdev':stdev,
                          'Weight':portWeight.iloc[:,0]}, index = cagr.index)

    # Calculate portfolio return
    portReturn = sum(dataBL.Return*dataBL.Weight)

    # Calculate portfolio covariance matrix using a helper function
    portCov = calculateCov(correlation.values, stdev)

    # Calculate portfolio risk using weight matrix and covariance matrix
    portRisk = np.matmul(dataBL.Weight.transpose(), portCov)
    portRisk = np.matmul(portRisk, dataBL.Weight)

    # Calculate risk aversion coefficient
    riskAversion = (portReturn - rfr)/portRisk

    # Calculate implied equilibrium return
    equiReturn = riskAversion*np.matmul(portCov, dataBL.Weight)

    # Convert result to pandas.Series format
    equiReturn = pd.Series(equiReturn[:,], index = cagr.index)

    return equiReturn

```

Listing 3. Simplified version of Black-Litterman model in the program.

The helper function `calculateMarketCapWeight` will need fund total assets data from Bloomberg when it calculates weight matrix. We use helper function `calculateCov` instead of NumPy's `cov` function to calculate covariance matrix, as it allows user to choose semivariance option for the model. Whenever user start the optimization process, program will check if Black-Litterman model is selected. If it was, then the program will first run the `blackLitterman` function to calculate equilibrium return and use that in the optimization model. Otherwise, the program will use historical 5-year CAGR as expected return instead.

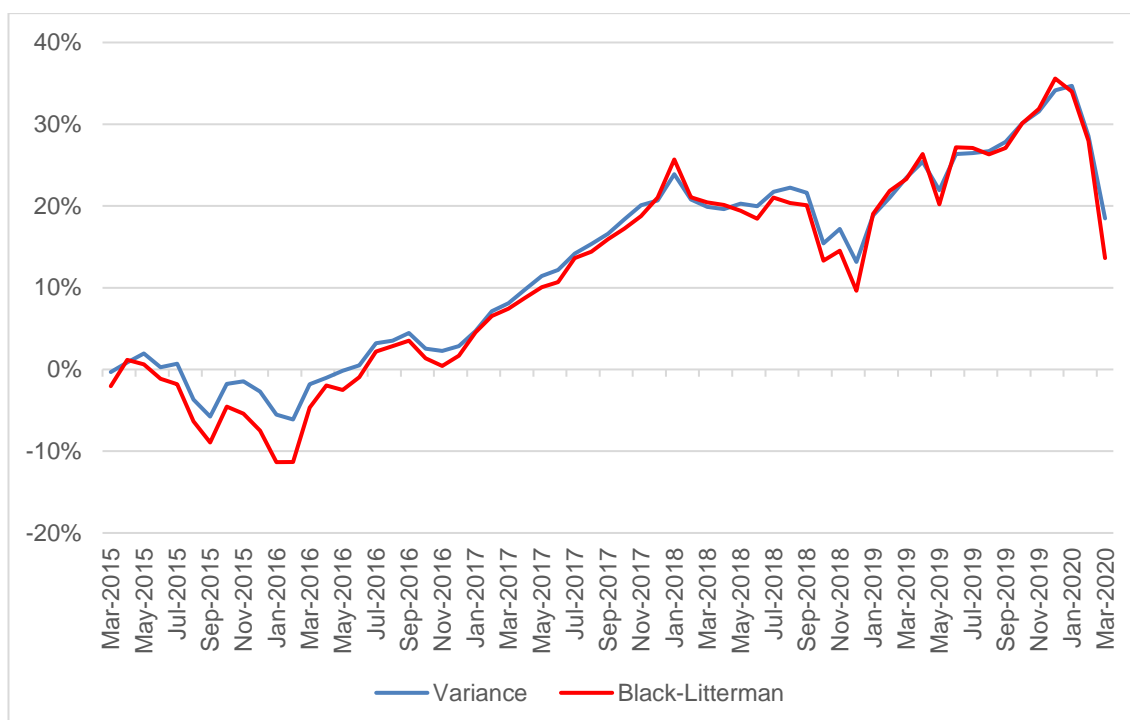


Figure 8. Cumulative gross return of Sharpe portfolios using variance and Black-Litterman

The performance of two Sharpe portfolios, one calculated using traditional return-variance method, and other using equilibrium return from BL model can be seen in Figure 8. Semivariance method was not included in the figure because its performance is closely represented by variance portfolio. Although BL portfolio failed to compete in the end of 2015 and first quarter of 2016, it recovered quickly and outperformed the variance portfolio during the bull market period after that. And it yielded higher return for the entire year of 2019. Another important period is the 1st quarter of 2020, when the coronavirus pandemic reached outside China. At January and February 2020, BL performance was higher than variance portfolio, however, during March 2020 when the number of daily cases in United States and European countries increased significantly [26], BL portfolio actual return was -11.222% (while variance portfolio has return of -7.842%). Looking at the allocation in appendix 1, we found the reason is due to BL model allocation more equity United States and European countries. The economic downturn due to coronavirus was an outlier event that rarely happen, with the closest event can related to it is the Great Depression in 1930s [27]. If we remove the 2020 performance for this reason, we can say that during the period from March 2015 to December 2019, Black-Litterman portfolio has outperformed traditional variance portfolio.

As mentioned in chapter 2.4, during the backtest process, we cannot use investor's view as an input. The result we obtained is the BL model without investor's view. In practice use, however, program allow investors to input their view of future performance, and their confident level of the views in the model. Users need to perform their own backtest to see the impact of their views to the portfolio performance, as their forecasts may not correctly reflect the actual result of the market.

4.3 Portfolio Simplification

Up until this section, portfolio return was calculated without including any fees, i.e. gross return. In practice, when invested in ETFs, investors must pay at least 3 types of fee: commission fee, management fee and bid-ask spread. Commission fee is the fee investors must pay to their stockbrokers for processing their purchase or sales of ETFs. This fee varies depend on the stockbrokers and is often based on the total value of the trade. Management fee is fee collected by the ETFs management company to cover for administration cost. Generally, for tracking ETFs, the bigger the fund assets value, the smaller the percentage of the fee. Bid-ask spread is the different in the price the buyer willing to pay (bid) and the price the seller want to accept (ask). High bid-ask spread will cost investors when they want to trade their ETFs. For example, ETF with bid-ask spread of 1% mean that investor purchase the ETF with \$100, when immediately sell them back can only get \$99. ETFs with high trading volume will have lower bid-ask spread, while illiquid ETFs will have higher spread.

Our portfolio has 75 ETFs, among those ETFs, there are ETFs have higher fee than other. For bid-ask spread, we have EGPT (5.677%) and GXG (3.997%). IWM have 1.19% management fees, while SPY has only 0.095% and United States Select Sectors ETF (XLF, XLK, XLE ...) have 0.03% to 0.035% annual fee. When including fee in return calculation, i.e. net return, we found a significant amount of portfolio return was offset by fees. While the portfolio is suitable for studying the effect of different market condition and model, it is not applicable in real investment world. Instead, we composed a simplified version of portfolio, with target to replicate the global portfolio, while focus on ETFs with lower fees. Details of portfolio is listed in Table 3 below:

Table 3. Detail of simplified portfolio

Code	Class	Name	Bid-ask spread	Management fee	Max. allocation
VGSH	Bond ETF	Vanguard Short-term Treasury	0.016%	0.050%	10%
IEI	Bond ETF	iShares 3-7Y Treasury	0.008%	0.070%	10%
MBB	Bond ETF	iShares MBS ETF	0.020%	0.060%	10%
TIP	Bond ETF	iShares TIPS Bond ETF	0.011%	0.190%	10%
LQD	Bond ETF	iShares iBoxx Investment Grade Corporate Bond ETF	0.012%	0.150%	10%
HYG	Bond ETF	iShares iBoxx High Yield Corp	0.013%	0.490%	10%
EMB	Bond ETF	iShares J.P. Morgan USD Emerging Markets Bond ETF	0.015%	0.390%	10%
VNQ	Alternatives	Vanguard Real Estate ETF	0.014%	0.110%	6%
IAU	Alternatives	iShares Gold Trust	0.062%	0.250%	6%
CPER	Alternatives	United States Copper Index Fund	0.723%	0.650%	3%
SPY	Equity	SPDR S&P 500 ETF Trust	0.004%	0.095%	20%
QQQ	Equity	Invesco QQQ Trust	0.005%	0.200%	15%
VGK	Equity	Vanguard FTSE Europe ETF	0.022%	0.090%	15%
EWJ	Equity	iShares MSCI Japan ETF	0.020%	0.490%	10%
MCHI	Equity	iShares MSCI China ETF	0.017%	0.590%	15%
INDA	Equity	iShares MSCI India ETF	0.039%	0.650%	6%
IEMG	Equity	iShares Core MSCI Emerging Markets ETF	0.023%	0.130%	9%

The new portfolio, called simplified portfolio, have 17 ETFs and data available from October 2012 to March 2020. Apart from CPER, an important ETF to gain exposure to material sector, each ETFs was chosen based on their low bid-ask spread and low management fee. We partially based maximum allocation on GDP to represent a global portfolio, while put higher allocation in SPY and QQQ (United States equity) due to its lower fee. The performance of 75 ETFs portfolio and simplified portfolio is compared in Table 4:

Table 4. Performance comparison during November 2017 - March 2020

	75 ETFs Portfolio	Simplified portfolio
Gross accumulated return	0.088%	2.446%
Net accumulated return	-1.605%	1.668%
Monthly average gross return	0.040%	0.133%
Monthly average bid-ask spread fee	0.017%	0.004%
Monthly average management fee	0.025%	0.019%
Monthly average commission fee	0.006%	0.011%
Correlation	0.99264	

Simplified portfolio did not have price data before October 2012; therefore, we can only calculate its performance during November 2017 to March 2020 (data from October 2012

to October 2017 is used to calculate Sharpe Portfolio, then portfolio was tested with actual result in November 2017). We set a fixed commission fee of 0.1% for both portfolio during the calculation. While simplified portfolio outperforms its peer, without more data we cannot be certain about its performance. Nevertheless, from cost saving perspective, we can confirm that simplified portfolio is better. On average, it cost 0.005% a month more in commission fee, due to higher trading value, but saving 0.019% a month in both bid-ask spread and management fee. During the testing period, invested in original portfolio would cost investor total 1.693% of their return on fees, while invested in simplified portfolio cost investor 0.778% of total return. The correlation of 0.99264 implies that the simplified portfolio has succeeded in tracking the global portfolio during the period.

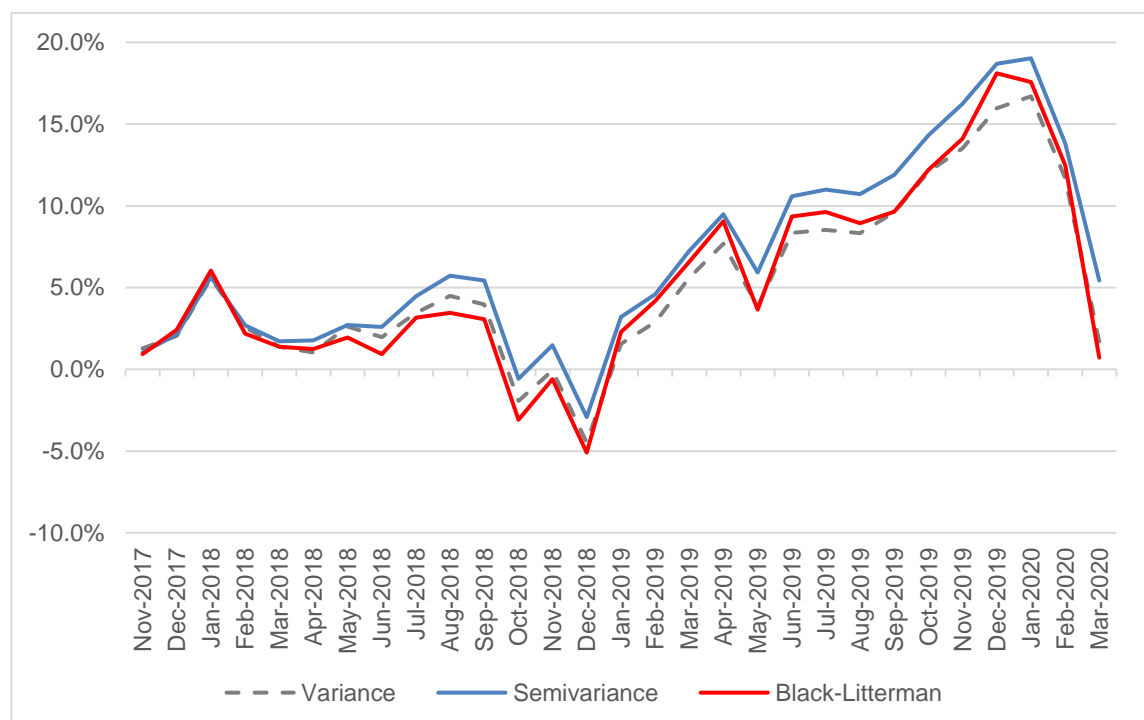


Figure 9. Cumulative net return of simplified Sharpe portfolios with improvements

Results of the actual performance of simplified portfolio using variance, semivariance and BL model are shown in Figure 9. We found that BL model performance is consistent compared with original portfolio: good performance during bull market and worst performance during market crash. The semivariance portfolio, which had similar performance with original portfolio, beat the variance simplified portfolio. This could be caused by its higher maximum allocation, allow the difference in semivariance and variance widen.

With the lack of data, it is difficult to confirm the benefits of improvement in simplified portfolio. For that reason, investors are advised to testing the performance of each model further until they apply it in practice.

5 Conclusion

5.1 Summary

In this thesis, the process of developing a robot-advisor program with semivariance and Black-Litterman model is introduced, implement and studied. The program is developed and backtested with historical data from February 2010 to March 2020, using 75 different ETFs to represent a global portfolio with 60% equity, 40% bond and alternatives investment.

Python was chosen as programming mainly due to its useful libraries. Those libraries provide the tool to handle big data, perform matrix operations, draw interactive charts, and solve convex optimization problems; all are needed to develop the robo-advisor program. Readily available mathematical functions also help to improve development speed significantly when working with complex calculation problems.

The result for semivariance improvement is mixed: while the rationale behind semivariance is well founded, its actual performance is no different than portfolio using variance. One reason is the assets in portfolio are restricted by max allocation, which makes the portfolio more diversified and less volatile. Other studies also pointed out that the performance of semivariance portfolio are depended on market condition, and, consequently, they are not always better than variance portfolio.

The Black-Litterman (BL) model is a portfolio allocation model seeking to overcome the practical problem of Modern Portfolio Theory in estimating expected return. Instead of using historical expected return, BL uses market equilibrium returns and investor's views as inputs. The result from using BL model is generally positive, the model generally significantly outperforms variance portfolio during the testing period, except for the extreme market condition in March 2020. It is important to further test the BL model during

economic decline, as it showed that BL did worse during such environment. Furthermore, the impact of investor's view was not studied in the backtest, therefore it should be researched before putting into practice.

While the portfolio was beneficial in study improvement by different models, it was not practical due to higher fees. A simplified portfolio was constructed to track the portfolio with the benefit of lower fees. Historical performance showed that simplified portfolio was cheaper, while closely resembling the original portfolio. In practice, investors are recommended to use simplified portfolio to reduce cost and consequently yield higher return. The improvements applied to simplified portfolio generally showed positive results but need more data to be able to confirm its benefit.

5.2 Discussion

During the construction of portfolio, one of the important variables in the model is the expected return. From the variance model to the BL model, we only use historical data to calculate the expected return. Using past performance in predicting future result might not yield a good result, it is the reason BL model used market expectation and included investors' view on the market. We did not include the investors' view because we did not have the historical data on investor's views, furthermore if investors are not familiar with the countries they invested in, their views are not reliable. Seeking advice from equity analysts and using their forecast as expected return or investors' view could improve the performance. However, more human interaction would result in higher consulting fees, and go against the purpose of a low cost robo-advisor software. One solution for this problem is to create an artificial intelligence software to find the sentiment regarding the market performance in the future. That software will browse through all public news, report and research. This process is called web crawling and it extracts the important keywords related to market. Then, using natural language processing technology, we can extract the market view on future, and use it as input for our model. It would be complex to develop a global sentiment software, as for different countries, we will need different languages for the software. Even so the result we obtain could provide better theoretical and practical benefit for the robo-advisor.

Another point to be addressed is how the portfolio can protect investors' wealth during a turbulent market. As Figure 9 showed, the cumulative return gained during the period was canceled by the coronavirus pandemic in February and March 2020. This event is rare and, therefore, did not have any historical data to base the model on. As a result, when it does happen, the model has terrible performance. Instead of relying on historical data, extreme events can be created from Monte Carlo simulation. Furthermore, Modern Portfolio Theory assume that market is efficient and investors are rational in making investment decisions. Monte Carlo simulation can also simulate an inefficient market with irrational investors for testing the model. Generating such simulation for global market, with different countries, different probability distributions and many variables would be complex. Also, to achieve a rare market condition, for example with less than 0.01% probability or lower, a large amount of data must be generated. As discussed earlier, the performance speed of Monte Carlo simulation can be improved with quantum computers. It should be noted that the development of such extensive, complex simulation, and the application of quantum computer in investment would take a long time until we have a positive result.

To solve the problem of data availability, we can replace ETFs with country equity indexes. While index cannot be traded and it is not applicable in real portfolio, building a portfolio from it allows us to backtest portfolio performance during 2008 crisis, 2000 dot-com bubble, or the Black Monday on October 19, 1987. Constructing such portfolio is difficult, as not all index tracks are relevant ETFs in our portfolio. Using equity index to represent ETFs should be done with caution, as they might not correlate with each other in the past. For example, many big technology companies in 2020 were founded during the 1990s, and any technology indexes before 1990s did not include those companies in their data.

Another point is the rebalancing of portfolio, which was done monthly in this thesis. Monthly balancing allows the portfolio to update with new information, however it comes with the cost of higher transaction fees. Further research should focus on the rebalancing frequency, e.g., monthly, quarterly or semi-annually, to find the optimal point where the cost of transaction fee diminishes the benefit of portfolio rebalancing.

References

- 1 U.S. Securities and Exchange Commission, "Investor Bulletin: Robo-Advisers," 23 February 2017. [Online]. Available: https://www.sec.gov/oiea/investor-alerts-bulletins/ib_robo-advisers.html. [Accessed 11 April 2020].
- 2 Backend Benchmarking, "The Robo Report First Quarter 2018," 2018.
- 3 Backend Benchmarking, "The Robo Report Fourth Quarter 2019," 2020.
- 4 Waters, R. "Wall Street banks ramp up research into quantum finance," 6 January 2020. [Online]. Available: <https://www.ft.com/content/45f3f1fe-2b14-11ea-bc77-65e4aa615551>. [Accessed 15 April 2020].
- 5 Nobel Media AB, "Press release," 16 October 1990. [Online]. Available: <https://www.nobelprize.org/prizes/economic-sciences/1990/press-release/>. [Accessed 10 April 2020].
- 6 Andrew, C. "Performance Evaluation," 2014. [Online]. Available: <https://www.cfainstitute.org/-/media/documents/support/programs/investment-foundations/19-performance-evaluation.ashx?la=en>. [Accessed 23 November 2019].
- 7 Markowitz, H. M. "Portfolio Selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77-91, 1952.
- 8 Sharpe, W. F. "The Sharpe Ratio," *The Journal of Portfolio Management*, vol. 21, no. 1, pp. 49-58, 1994.
- 9 Best, M. J. and Grauer, R. R. "On the Sensitivity of Mean Variance-Efficient Portfolios to Changes in Asset Means: Some Analytical and Computational Results," *The Review of Financial Studies*, vol. 4, no. 2, pp. 315-342, 1991.
- 10 Buffett, W. E. "Chairman's Letter 1996," 28 February 1997. [Online]. Available: <http://www.berkshirehathaway.com/letters/1996.html>. [Accessed 24 November 2019].
- 11 Black, F. and Litterman, R. "Asset Allocation Combining Investor Views with Market Equilibrium," *Journal of Fixed Income*, osa/vuosik. 1, nro 2, pp. 7-18, 1991.
- 12 Maginn, J. L., Tuttle, D. L., McLeavey D. W. and Pinto, J. E. *Managing Investment Portfolios: A Dynamic Process* 3rd edition, New Jersey: John Wiley & Sons, Inc., 2007.
- 13 Vose, D. *Risk Analysis - A quantitative guide*, Chichester: John Wiley & Sons, Ltd, 2008, p. 45.

- 14 Oliphant, T. E. A guide to NumPy, Trelgol Publishing USA, 2006.
- 15 McKinney, W. "Data structures for statistical computing in python," Proceedings of the 9th Python in Science Conference, vol. 445, pp. 51-56, 2010.
- 16 Diamond, S. and Boyd, S. "CVXPY: A Python-Embedded Modeling Language for Convex Optimization," 2016. [Online]. Available: https://web.stanford.edu/~boyd/papers/pdf/cvxpy_paper.pdf. [Accessed 25 November 2019].
- 17 Grayson, J. E. Python and Tkinter Programming, vol. 2000, Greenwich: Manning, 2000.
- 18 PyInstaller Development Team, "PyInstaller official website," 14 February 2020. [Online]. Available: <https://www.pyinstaller.org/>. [Haettu 09 May 2020].
- 19 U.S. Securities and Exchange Commission, "Investor Bulletin: Exchange-Traded Funds (ETFs)," August 2012. [Online]. Available: <https://www.sec.gov/investor/alerts/etfs.pdf>. [Accessed 12 April 2020].
- 20 World Bank, "GDP (current US\$)," 2019. [Online]. Available: <https://data.worldbank.org/indicator/NY.GDP.MKTP.CD>. [Haettu 12 April 2020].
- 21 Republic of China (Taiwan), "Economy," [Online]. Available: https://www.taiwan.gov.tw/content_7.php. [Accessed 12 April 2020].
- 22 Bank for International Settlements, "Summary of debt securities outstanding," 27 February 2020. [Online]. Available: <http://stats.bis.org:8089/statx/srs/table/c1?f=xlsx>. [Accessed 12 April 2020].
- 23 He, G. and Litterman, R. The Intuition Behind Black-Litterman Model Portfolios, New York, 1999.
- 24 Markowitz, H. M. "Foundations of Portfolio Theory," The Journal of Finance, vol. 46, no. 2, pp. 469-477, 1991.
- 25 Tcysin, A. and Dmitrii, P. Bayes-Stein Estimators in Mean-Semivariance, 2017.
- 26 "COVID-19 Coronavirus data," 08 May 2020. [Online]. Available: <https://www.ecdc.europa.eu/sites/default/files/documents/COVID-19-geographic-disbtribution-worldwide.xlsx>. [Accessed 08 May 2020].
- 27 Gopinath, G. "The Great Lockdown: Worst Economic Downturn Since the Great Depression," 14 April 2020. [Online]. Available: <https://blogs.imf.org/2020/04/14/the-great-lockdown-worst-economic-downturn-since-the-great-depression/>. [Accessed 08 May 2020].

Appendix 1: Portfolio allocation in March 2020

Code	Name	Variance	Semivariance	Black-Litterman
SPY	SPDR S&P 500 ETF TRUST	6.00%	6.00%	6.00%
VGK	VANGUARD FTSE EUROPE ETF	0.00%	0.00%	0.00%
EWG	ISHARES MSCI GERMANY ETF	0.00%	0.00%	3.17%
EEM	ISHARES MSCI EMERGING MARKET	0.00%	0.00%	0.00%
EWX	ISHARES MSCI MEXICO ETF	0.00%	0.00%	0.00%
IWM	ISHARES RUSSELL 2000 ETF	3.00%	3.00%	3.00%
EWO	ISHARES MSCI AUSTRIA ETF	0.00%	0.67%	0.00%
VNM	VANECK VIETNAM ETF	0.00%	0.00%	0.86%
RSX	VANECK RUSSIA ETF	2.50%	2.50%	0.00%
EWJ	ISHARES MSCI JAPAN ETF	6.00%	0.00%	6.00%
SCJ	ISHARES MSCI JAPAN SMALL-CAP	2.00%	2.00%	0.70%
EWZ	ISHARES MSCI BRAZIL ETF	0.00%	0.94%	0.00%
TUR	ISHARES MSCI TURKEY ETF	0.00%	0.00%	0.00%
EWC	ISHARES MSCI CANADA ETF	0.00%	0.00%	1.62%
EWU	ISHARES MSCI UNITED KINGDOM	0.00%	0.00%	0.00%
HAO	INVESCO CHINA SMALL CAP ETF	3.50%	3.50%	0.00%
EWN	ISHARES MSCI NETHERLANDS ETF	1.50%	1.50%	1.50%
DFE	WISDOMTREE EUR S/C DIVIDEND	0.00%	1.00%	0.00%
IND	ISHARES INDIA 50 ETF	1.68%	4.00%	0.54%
EWQ	ISHARES MSCI FRANCE ETF	4.00%	4.00%	0.77%
EWD	ISHARES MSCI SWEDEN ETF	0.00%	0.00%	0.33%
FDD	FIRST TRUST STOXX EUROPE	1.00%	1.00%	1.00%
CHI	GLOBAL X MSCI CHINA CONSUMER	3.50%	3.50%	0.77%
EWI	ISHARES MSCI ITALY ETF	0.00%	0.00%	0.00%
EWP	ISHARES MSCI SPAIN ETF	0.00%	0.00%	0.00%
EWY	ISHARES MSCI SOUTH KOREA ETF	0.00%	0.00%	0.00%
VGS	VANGUARD SHORT-TERM TREASURY	0.50%	0.50%	1.85%
IEI	ISHARES 3-7 YEAR TREASURY BO	4.00%	4.00%	0.50%
IEF	ISHARES 7-10 YEAR TREASURY B	4.00%	4.00%	3.90%
HYG	ISHARES IBOXX HIGH YLD CORP	0.50%	0.50%	5.00%
LQD	ISHARES IBOXX INVESTMENT GRA	5.00%	5.00%	5.00%
AGG	ISHARES CORE U.S. AGGREGATE	5.00%	5.00%	5.00%
EMB	ISHARES JP MORGAN USD EMERGI	5.00%	5.00%	5.00%
BIV	VANGUARD INTERMEDIATE-TERM B	2.00%	2.00%	2.00%
TLH	ISHARES 10-20 YEAR TREASURY	4.00%	4.00%	0.39%
TIP	ISHARES TIPS BOND ETF	0.00%	0.00%	4.00%
BSV	VANGUARD SHORT-TERM BOND ETF	2.00%	2.00%	2.00%
MBB	ISHARES MBS ETF	2.00%	2.00%	0.02%
BWX	SPDR BBG BARC INTL TREASURY	0.00%	0.00%	0.00%
IEA	ISHARES CORE EURO CORP BOND	0.00%	0.00%	0.24%

USO	UNITED STATES OIL FUND LP	0.00%	0.00%	0.53%
GLD	SPDR GOLD SHARES	6.00%	6.00%	1.98%
SLV	ISHARES SILVER TRUST	0.00%	0.00%	0.09%
NIB	IPATH BLOOMBERG COCOA SUBIND	0.00%	0.00%	0.00%
VNQ	VANGUARD REAL ESTATE ETF	0.00%	0.00%	2.50%
IWF	ISHARES RUSSELL 1000 GROWTH	2.00%	2.00%	2.00%
VTV	VANGUARD VALUE ETF	2.00%	2.00%	2.00%
VIG	VANGUARD DIVIDEND APPREC ETF	2.00%	2.00%	2.00%
FXI	ISHARES CHINA LARGE-CAP ETF	0.00%	0.00%	0.03%
CQQ	INVESCO CHINA TECHNOLOGY ETF	2.32%	3.39%	0.73%
CHI	GLOBAL X MSCI CHINA FINANCIA	0.00%	0.00%	0.39%
EWA	ISHARES MSCI AUSTRALIA ETF	0.00%	0.00%	2.00%
EWK	ISHARES MSCI BELGIUM ETF	0.00%	0.00%	1.00%
ECH	ISHARES MSCI CHILE ETF	0.00%	0.00%	0.00%
GXG	GLOBAL X MSCI COLOMBIA ETF	0.00%	0.00%	0.00%
EGP	VANECK EGYPT	0.00%	0.00%	0.00%
EWS	ISHARES MSCI SINGAPORE ETF	0.00%	0.00%	1.00%
EWH	ISHARES MSCI HONG KONG ETF	0.00%	0.00%	0.00%
EZA	ISHARES MSCI SOUTH AFRICA ET	0.00%	0.00%	0.00%
EWT	ISHARES MSCI TAIWAN ETF	1.00%	1.00%	1.00%
THD	ISHARES MSCI THAILAND ETF	0.00%	0.00%	0.59%
EWL	ISHARES MSCI SWITZERLAND ETF	1.00%	1.00%	1.00%
IEU	ISHARES MSCI EUROPE SMALL-CA	1.00%	1.00%	0.00%
XLF	FINANCIAL SELECT SECTOR SPDR	2.00%	2.00%	2.00%
XLK	TECHNOLOGY SELECT SECT SPDR	2.00%	2.00%	2.00%
XLE	ENERGY SELECT SECTOR SPDR	0.00%	0.00%	2.00%
XLV	HEALTH CARE SELECT SECTOR	2.00%	2.00%	2.00%
XLY	CONSUMER DISCRETIONARY SELT	2.00%	2.00%	2.00%
XLI	INDUSTRIAL SELECT SECT SPDR	2.00%	2.00%	2.00%
XLP	CONSUMER STAPLES SPDR	2.00%	2.00%	2.00%
XLU	UTILITIES SELECT SECTOR SPDR	2.00%	2.00%	2.00%
XBI	SPDR S&P BIOTECH ETF	0.00%	0.00%	1.00%
XLB	MATERIALS SELECT SECTOR SPDR	0.00%	0.00%	1.00%
XHB	SPDR S&P HOMEBUILDERS ETF	0.00%	0.00%	1.00%
XME	SPDR S&P METALS & MINING ETF	0.00%	0.00%	1.00%