MFE 5100 Fall 2022 Project: Portfolio Optimization

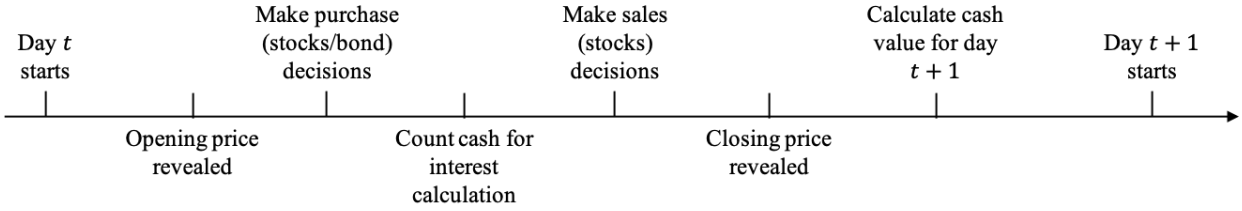
The purpose of this course project is to solve a real problem in financial engineering using optimization. This semester, we will spend much of our class time learning how to build and solve mathematical programs, such as linear programs, nonlinear programs, and integer programs. However, sometimes mathematical programs are not the best choice for optimization. Some problems can be solved to optimality using other techniques. For other problems, we can use heuristic techniques to get a quick solution that, although not guaranteed to be optimal, may be a good solution. It is now the time to use what you have learned, plus other techniques or heuristics you may have acquired from outside this class, to tackle a financial engineering related problem.

In this project, you are provided with the stock market data from the year of January 2000 to December 2017. Each of the .csv files corresponds to a stock, with its stock code the same as the .csv file name. Each file contains the date when the data was recorded, the opening price, the highest price, the lowest price, the closing price, the adjusted closing price in current value, and the traded volume of that specific date. Some stocks may not have the data for the full length of 18 years because they did not go on IPO until the first date recorded in the data. The unit of the stock price is in U.S. dollar ($), and that of daily trading volume is in shares.

You just graduated from the MFE program at CUHK-Shenzhen, and you got a job at a top hedge fund. Your first task in the company is to manage $100,000 starting from January 1, 2018, until December 31, 2019. One alternative is that you can invest them in stocks or government bonds. The bond has a 3% annual return, and it matures every six months, which means if you buy it on January 1st of the year, you will get your money back on July 1st of the same year. You cannot use this fund elsewhere between the bond purchase date and the maturity date, and you can only purchase the bond on January 1st or July 1st of a specific year. If January 1st/July 1st is not a working day, the bond matures on the last working day before January 1st/July 1st, and you can purchase the bond the first working day after January 1st/July 1st. Another alternative is to invest them in the stock market, where each day you can decide to buy after the opening price is revealed. If you decide to sell a stock on a specific date, the closing price will be considered the selling price. The third alternative is to hold the cash, and the daily interest rate is 0.005%.

At the beginning of each day, the opening price of the stock is revealed. Based on the opening price, the historical data, and the test data up till today, you can make decisions on buying stocks. On specific days you can also buy the bond. Once you have made the purchase decisions, the purchase transactions will take place, and the cash value will be recorded to calculate the interests of that day. You can also decide to sell some stocks, and the price sold will be the closing

price, which will not be revealed to you before you make the sales decision. After the closing price is revealed, your sales transaction takes place. Then we calculate the cash value going into the next day by summing up the cash and its interest after purchasing stocks and the bond and the sales value of the stock.



For example, suppose at day t on which we can buy the bond, the initial cash value is $\$C$. We hold 1,000 shares of stock A. We decide to buy 100 shares of stock B at the opening price $\$c_B$ per share and $\$C_B$ of the bond. We decide to sell 500 shares of stock A at the closing price $\$c_A$ per share. The next day's initial cash value is:

$$\$\,1.00005(C - 100c_B - C_B) + 500c_A.$$

And at the beginning of day $t+1$, we also hold 500 shares of stock A, 100 shares of stock B, and $\$C_B$ worth of the bond.

The company has a software department that has already coded the computer program to read in the data and implement your investment strategy. You do not have to use those codes, but they will be helpful for this project.

Your goal is to build your portfolio management strategy that involves investing in stocks, the bond, and cash to maximize the value of your portfolio on January 1st, 2020. The value of your portfolio equals the summation of the value of your stocks at the closing price of December 31, 2019, the value of the matured bonds, and the cash on January 1, 2020. You need to justify your choice of strategy by going through the tasks below, because your boss would like to know why you think your strategy will maximize the portfolio value.

Tasks:

1. **Fixed strategy:** We first consider a two-year fixed investment, which means that the investment remains the same throughout two years and no buy/sell actions are carried out.

   (a) Build a linear program model to maximize the mean portfolio's return by selecting a portfolio of assets (stocks, bonds, and cash). Here you can choose any method to estimate the mean return of the two years based on the historical data.

(b) Run your strategy on the sample data. If the sample data were reality, how much different would your realized profit be from the optimal value you obtain in part (a)? What would be the best portfolio selection if the sample data were reality? How does this "best portfolio selection" compare to your selection in part (a)?

(c) As you can see from the first two questions, the two-year return of a stock may be uncertain, and you cannot predict the future. To consider the uncertainty, you should start by estimating the variance for the two-year return of each stock. Notice that some stocks' prices may be correlated.

(d) Following part (c), assume the prices are normally distributed. Sample 5000 scenarios to calculate the 5%-, 25%-, 50%-, 75%-, 95%-percentile of your selected stocks' return.

(e) Build a mathematical program to construct a portfolio that takes variance into consideration (e.g., Markowitz models covered in the lecture). How does this portfolio differ from the one obtained only considering the mean return? For the same 5000 scenarios, calculate the 5%-, 25%-, 50%-, 75%-, 95%-percentile of your selected stocks' return.

(f) (Optional) You do not have to limit yourselves to the expected value and the variance when setting up the objective function for your optimization model. You can build a predictive model (e.g., regression and random forest) for the return of each stock based on historical data. Construct an optimization model based on your prediction and discuss your strategy's performance on the sample data set and the random 5000 scenarios.

2. **Monthly-trading strategy:** Next, we consider a monthly trading strategy, which means you can make a buy/sell decision at the beginning of each month.

(a) The monthly-trading strategy is more flexible than the fixed strategy. Discuss how this flexibility may benefit the final return.

(b) (Deterministic rolling horizon model) Once a month ends, you can observe the return of each stock. With this new observation, you can update the mean/variance estimation similar to part (a) and (c) or your prediction similar to part (f) in the fixed strategy tasks. Design a model or a process to update your estimation/prediction, use the updated information, and sequentially optimize each month's investment decisions.

(c) (Stochastic rolling horizon model) Suppose you want to use the future prediction as well every time you update your model parameters. This is a rolling horizon process with a two-stage stochastic program solved at every update. Construct such a two-stage stochastic program properly, with a clear explanation about the modeling details and necessary comparison results, including but not limited to: the length of your second-stage time horizon and the reason you choose this; the prediction scenario generation process; the performance of your stochastic rolling horizon model and comparison to

the fixed strategy and the deterministic rolling horizon model on the sample data set; advantages and disadvantages of such stochastic rolling horizon model.

3. **Dynamic strategy:**

   (a) You can further extend your model to a daily-trading strategy or even an intra-day trading strategy (our data do not permit this). Since this type of dynamic strategy may require fast execution, you may not be able to solve an optimization problem but rely on some heuristics, e.g., scalping strategy. Select one such heuristic trading strategy that does not require any optimization via your research. Describe and implement the strategy. Compare this strategy with the fixed strategies and monthly-trading strategies. Do the optimization methods you use in parts 1 and 2 actually make your profit better? Discuss your observation.

Implement your strategies using python. You can use the standard code flow that is provided to you to process the input and the output, or you can write your own from scratch.

Codes Provided and Inputs:

1. `Stock_Data.zip`: the .zip contains stock historical data. After decompressing, you should be able to find .csv files in two folders. Each .csv file corresponds to a stock. Each row is for a trading day, and each column represents the related data field. The folder "train" contains the historical data. The folder "sample_test" contains sample test data that can be used for your code development. Notice that the sample test data **in no way** represents reality. Do not directly base your strategy on the sample test data.

2. `main.py`: the python codes provided to you, which contain the data preprocessing and output procedures. Comments have been added to guide you to modify it so that your strategy is implemented and generates the right output for testing reasons.

   You can change the function `main` so that it takes in the portfolio state at the beginning of a given day and outputs the `buy_action` and `sell_action` that represent the stock/bond purchase and stock sales you perform during that day.

   The code has specified which day you can purchase the bond and which day the bond matures (`bond_purchase_days` and `bond_mature_days`). For the bond, the buy action should be recorded in $-value, but for stocks, the buy/sell actions should be recorded in shares. You do not have to worry about bond maturing as they will be calculated by the codes provided by the software department.

Three sample strategy functions `main_cash`, `main_bonds`, and `main_longterm_stock` are included in `main.py` to serve as examples.

3. `test_stock.py`: You can use the functions in `test_stock.py` for reading in the data and other testing purposes, but please do not change them so errors will not occur.

Deliverables:

1. A project report in PDF format, written in English, containing the following:
   - Detailed descriptions of your strategy by answering the questions in the "Task" session. For the mathematical programming models involved in your strategy, you need to clearly state your formulation, including the definition and meaning of index sets, decision variables and parameters, the objective function, and the constraints you choose. Your descriptions should make sense to a person with no optimization background.
   - If your strategy uses techniques outside mathematical programming, you need to describe each step of it in detail. You need to state the numeric supporting evidence.
   - All transaction records should match the result calculated from your codes. If the list of transaction records is too long, you can attach another file for that, e.g., a spreadsheet (.xls file).
   - Any required instructions to successfully run your code, including the name of packages you installed to perform calculations and the instructions to run scripts.
   - Citations for any sources you used.
2. A .zip file containing all your python codes. You can use the provided `main.py` or create other .py files to help you perform the calculation. You should also include any code that analyzes the data and produces your strategy.
3. Each student needs to fill in an evaluation form (empty form posted on Blackboard) and submit it on Blackboard individually. Consider you have \$1,000n, n is the number of people in your group. Allocate this money based on the contributions of each group member. Giving a member \$1,000 means that he/she has done his/her share. If you think group member A has done a significant amount of work, you can give him/her more than \$1,000; if you believe group member B did not contribute at all, you can give him/her less than \$1,000. Give a brief reason why you make your evaluation this way.

You will be graded based on the correctness of your formulations, on the clarity of your descriptions, and on the depth of your analysis.