# Stochastic Control and Optimization Project 2 report

*Stock selection and portfolio analysis*

**Abhinav Sharma (ass2575)**

**Aritra Chowdhury (ac79277)**

**Namit Agrawal (nra544)**

**Qingye Ding (qd854)**

## INTRODUCTION

An index fund is an investment that tracks a market index. In this project, we selected the NASDAQ-100 index. Generally, we could find at least one index fund that tracks the index. However, when we have multiple choices, we would like to figure out which index fund most closely tracks the performance of the index. After selecting the index fund, we decide how many of each chosen stock to buy.

### Method 1: Stock Selection and Weights Calculation - Integer Programing

We used two methods to select stocks. For the first method, we first calculated the correlation between the fund and each stock in the index to maximize the similarity between the index stock and its representative in the fund. The objective function and constraints are:

$$\max_{x,y} \sum_{i=1}^{n} \sum_{i=1}^{n} \rho_{ij} x_{ij}$$

$$s.t. \sum_{j=1}^{n} y_j = m$$

$$\sum_{j=1}^{n} x_{ij} = 1 \ for \ i = 1, 2, \ldots, n$$

$$x_{ij} - y_j \leq 0 \ for \ i, j = 1, 2, \ldots, n$$

$$x_{ij}, y_j \in \{0, 1\}$$

An example of the correlation matrix is as shown in Table 1, which indicates that a stock has the strongest correlation (equals to 1) with itself.

| | ATVI | ADBE | AMD | ALXN | ALGN | GOOGL | GOOG |
|---|---|---|---|---|---|---|---|
| **ATVI** | 1.000000 | 0.399939 | 0.365376 | 0.223162 | 0.216280 | 0.433097 | 0.426777 |
| **ADBE** | 0.399939 | 1.000000 | 0.452848 | 0.368928 | 0.363370 | 0.552125 | 0.540404 |
| **AMD** | 0.365376 | 0.452848 | 1.000000 | 0.301831 | 0.344252 | 0.418861 | 0.417254 |
| **ALXN** | 0.223162 | 0.368928 | 0.301831 | 1.000000 | 0.332433 | 0.315993 | 0.307698 |
| **ALGN** | 0.216280 | 0.363370 | 0.344252 | 0.332433 | 1.000000 | 0.248747 | 0.250316 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **WBA** | 0.218149 | 0.228106 | 0.281950 | 0.192720 | 0.219595 | 0.232900 | 0.230603 |
| **WDAY** | 0.311659 | 0.650430 | 0.407626 | 0.416396 | 0.308968 | 0.379493 | 0.371826 |
| **WDC** | 0.303077 | 0.361516 | 0.438892 | 0.289908 | 0.284407 | 0.328619 | 0.322110 |
| **XEL** | 0.043389 | 0.207403 | 0.017283 | 0.047947 | 0.088059 | 0.059930 | 0.052570 |

Table 1. An example of the correlation matrix

We then calculated the weighted return of each chosen stock to match the returns of the index. The objective function and constraints are:

$$\min_{w} \sum_{t=1}^{T} d_t$$

$$s.t. \sum_{i=1}^{m} w_i = 1$$

$$d_t + \sum_{i=1}^{m} w_i r_{it} \geq q_t \quad for\ t = 1, 2, \ldots, T$$

$$d_t - \sum_{i=1}^{m} w_i r_{it} \geq -q_t \quad for\ t = 1, 2, \ldots, T$$

$$w_i \geq 0$$

The return of each stock, $r_{it}$, is calculated using the pandas pct_changes function, which computes the percentage change from the immediately previous day. We calculated the return for both 2019 and 2020, as shown in Table 2 and 3.

| Date | NDX | ATVI | ADBE | AMD | ALXN | ALGN | GOOGL |
|---|---|---|---|---|---|---|---|
| 2019-01-03 | -0.033602 | -0.035509 | -0.039498 | -0.094530 | 0.022030 | -0.085791 | -0.027696 |
| 2019-01-04 | 0.044824 | 0.039903 | 0.048632 | 0.114370 | 0.057779 | 0.010445 | 0.051294 |
| 2019-01-07 | 0.010211 | 0.028196 | 0.013573 | 0.082632 | 0.018302 | 0.017192 | -0.001994 |
| 2019-01-08 | 0.009802 | 0.030309 | 0.014918 | 0.008751 | 0.006207 | 0.015954 | 0.008783 |
| 2019-01-09 | 0.007454 | 0.017210 | 0.011819 | -0.026988 | 0.012430 | 0.038196 | -0.003427 |

Table 2. An example of the return of each stock in 2019

| Date | NDX | ATVI | ADBE | AMD | ALXN | ALGN | GOOGL |
|---|---|---|---|---|---|---|---|
| 1/3/20 | -0.008827 | 0.000341 | -0.007834 | -0.010183 | -0.013260 | -0.011421 | -0.005231 |
| 1/6/20 | 0.006211 | 0.018238 | 0.005726 | -0.004321 | 0.001598 | 0.019398 | 0.026654 |
| 1/7/20 | -0.000234 | 0.010043 | -0.000959 | -0.002893 | 0.002533 | -0.009864 | -0.001932 |
| 1/8/20 | 0.007452 | -0.007623 | 0.013438 | -0.008705 | 0.016191 | 0.010386 | 0.007118 |
| 1/9/20 | 0.008669 | -0.009018 | 0.007636 | 0.023834 | 0.019893 | 0.036853 | 0.010498 |

Table 3. An example of the return of each stock in 2020

1. Select 5 stocks

We started by selecting 5 stocks, and the optimal selection of the funds are: 'LBTYK', 'MXIM', 'MSFT', 'VRTX', 'XEL' with weights of 0.04886175, 0.21038806, 0.58035198, 0.07119022, 0.089208, respectively. In 2019, the difference between the total return of these stocks and the return of index 'NDX' is around 0.789; in 2020, the difference is 1.112.

2. Select more stocks.

We also selected 10, 20, 30,...,100 stocks and compared their performance. The comparison metric is the absolute difference between the return of the portfolio and that

of the index. Therefore, the smaller the number shows the portfolio better tracks the index price.

Overall, as the number of stocks chosen increases, the performance also increases which makes sense as we are able to model the fund with more stocks, as shown in Table 4 and Figure 1. This phenomenon holds true even for the 2020 Stock Selection dataset as the loss is decreasing with respect to the number of stocks. However, there are exceptions in both data sets. In 2019, the performance of choosing 60 stocks was worse than choosing only 50 stocks. In 2020, there are diminishing returns after choosing 30 stocks as the loss stays stagnant and it even increases between m = 50 through m = 70. As a result, an optimal number of stocks would be to choose 30 stocks to model the NASDAQ-100 index.

To compare the portfolio performance across 2019 and 2020, we found that the 2019 performance is always better than in 2020. This makes sense because the return of each stock varies from 2019 to 2020, and we only used the 2019 data as the training data set to create the portfolio.


Figure 1. Portfolio performance using weights calculation after stock selection

## Method 2: Only used Weight Calculation - Mixed Integer Programing

In this method, we completely ignored the stock selection Integer programming problem and formulated an MIP problem that constrains the number of non-zero weights to be an integer. In order to do this, we also introduced a set of binary variables $y_i$, $y_2$, ..., $y_n$ and added some constraints that force $w_i = 0$ if $y_i = 0$. As a result, the objective function and constraints are:

$$\min_{w} \sum_{t=1}^{T} d_t$$
$$s.t. \sum_{i=1}^{n} w_i = 1$$
$$\sum_{i=1}^{n} y_i = m$$
$$w_i - M y_i \leq 0 \ for \ i = 1, 2, \ldots, n$$
$$d_t + \sum_{i=1}^{n} w_i r_{it} \geq q_t \ for \ t = 1, 2, \ldots, T$$
$$d_t - \sum_{i=1}^{n} w_i r_{it} \geq -q_t \ for \ t = 1, 2, \ldots, T$$
$$w_i \geq 0$$

The smallest value for M here would be 1 since the largest value that $w_i$ can have is 1 given that the sum of the weights must be 1. We limited this method to roughly 10 hours of runtime due to the fact that we had 2T+n+2 constraints and 2n+T decision variables (T = time period and n = total number of stocks).

### 1. Select 5, 10, 20, ... 100 stocks.

We also selected 5, 10, 20, 30,...,100 stocks for the second method and compared their performance. The comparison metric is the absolute difference between the return of the portfolio and that of the index. Therefore, the smaller the number shows the portfolio better tracks the index price.

Overall, as the number of stocks chosen increases, the performance also increases which makes sense as we are able to model the fund with more stocks, as shown in Table 5 and Figure 2. This phenomenon holds true even for the 2020 Stock Selection dataset as the loss is decreasing with respect to the number of stocks. However, diminishing returns start to settle in after 40 stocks which is larger compared to the method one where diminishing returns settled in after 30 stocks. This could be due to the fact that we are not using a correlation matrix that captures which stocks are closely related to one

another and hence needs more stocks in the fund to model NASDAQ-100 index adequately.

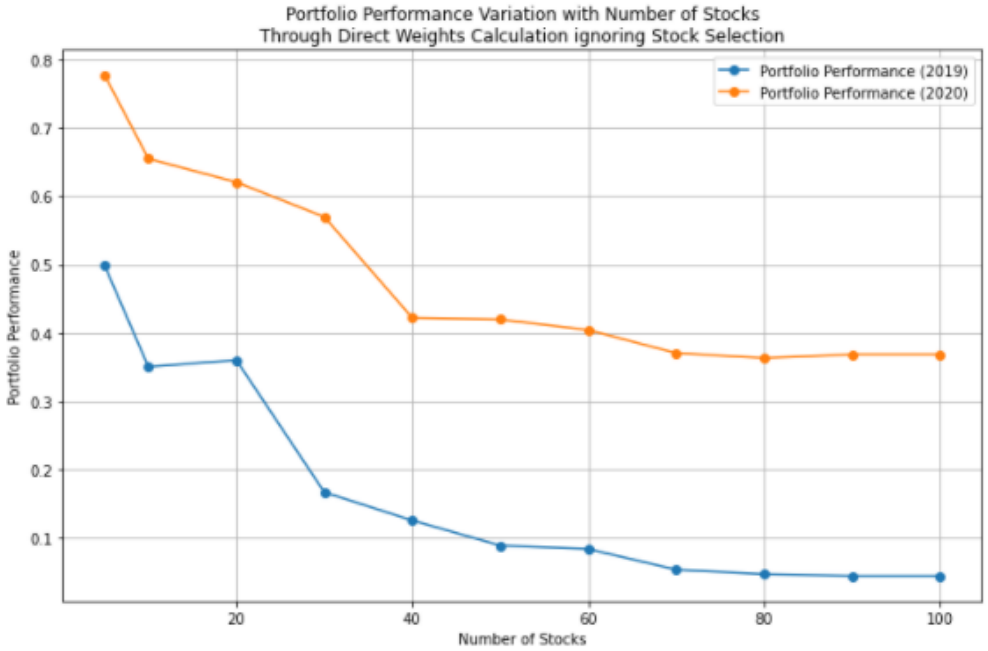


Figure 2. Portfolio performance using direct weight calculation

## Comparison between two methods

| | No of Stocks | 2019 Performance with Stock Selection | 2020 Performance with Stock Selection | 2019 Performance without Stock Selection | 2020 Performance without Stock Selection |
|---|---|---|---|---|---|
| **0** | 5.0 | 0.789178 | 1.112437 | 0.499259 | 0.777362 |
| **1** | 10.0 | 0.686533 | 1.097709 | 0.350523 | 0.654725 |
| **2** | 20.0 | 0.478836 | 0.899598 | 0.360332 | 0.620463 |
| **3** | 30.0 | 0.418015 | 0.769110 | 0.167172 | 0.569660 |
| **4** | 40.0 | 0.367439 | 0.788335 | 0.125930 | 0.422072 |
| **5** | 50.0 | 0.334010 | 0.773216 | 0.089318 | 0.419716 |
| **6** | 60.0 | 0.343788 | 1.166438 | 0.084161 | 0.404375 |
| **7** | 70.0 | 0.168587 | 0.545744 | 0.053993 | 0.370585 |
| **8** | 80.0 | 0.147683 | 0.537323 | 0.047347 | 0.363281 |
| **9** | 90.0 | 0.053779 | 0.367790 | 0.044911 | 0.368671 |
| **10** | 100.0 | 0.044911 | 0.368682 | 0.044911 | 0.368671 |

Table 5. Performance of portfolios using stock selection and no stock selection

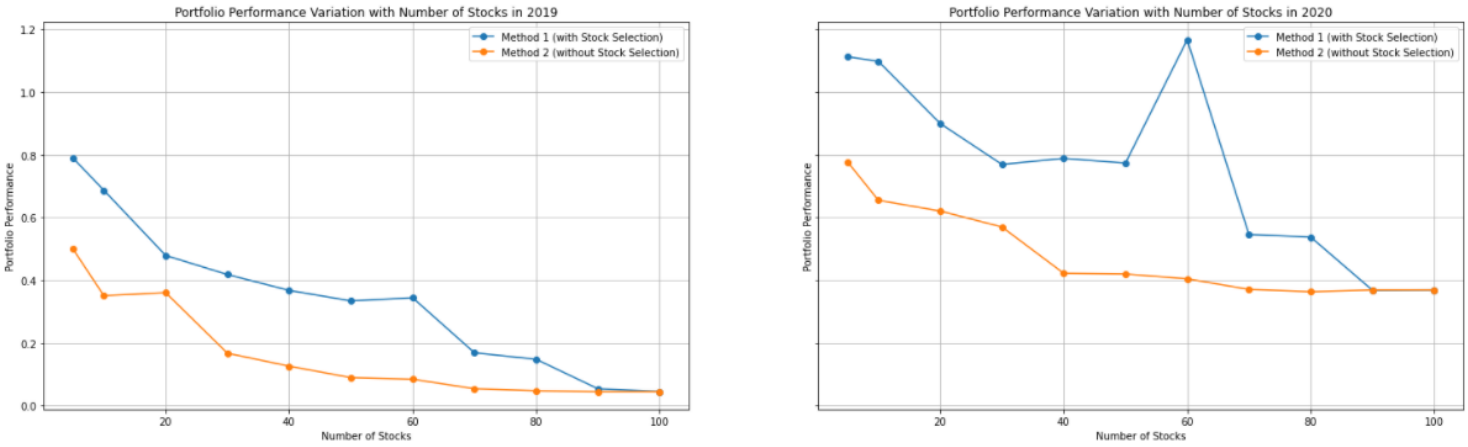


Figure 3. Comparison of the performance of two methods on the 2019 data set (left) and on the 2020 data set (right)

Method 1:

In 2019, the performance of choosing 60 stocks was worse than choosing only 50 stocks. In 2020, there are diminishing returns after choosing 30 stocks as the loss stays stagnant and it even increases between m = 50 through m = 70. As a result, an optimal number of stocks would be to choose **30 stocks** to model the NASDAQ-100 index.

Method 2:

Using this method, the performance keeps improving with an increasing number of stocks. In 2019, there is a single exception where the performance of choosing 20 stocks was worse than choosing only 10 stocks. In 2020, there are diminishing returns after choosing 40 stocks as the loss stays stagnant after. As a result, an optimal number of stocks would be to choose **40 stocks** to model the NASDAQ-100 index.

## Conclusion

It is clear from the above plot that Method 2 works better on the 2020 data. The performance is similar across both methods, however there is a lot more variance with Method 1 as the loss fluctuates with a higher magnitude. Since Method 2 is better, our final recommendation is to choose **40 stocks** to model the NASDAQ-100 index and model the optimization to directly calculate the weights without first selecting the stocks, i.e., use the **Second Method** going forward. We can improve our model by calculating the cost of adding more stocks to our portfolio and modelling the cost into our optimization constraints.