Programming Homework, CSE 440
Spring 2019, Michigan State University,

Portfolio Optimization Using Convex Optimizers in Python

Sari Saba-Sadiya
CSE 440, Spring 2019

Michigan State University
sadiyasa@msu.edu

ABSTRACT. Modern portfolio theory (MPT) was formulated by the economist Harry Markowitz at 1952, and later won him a Nobel Prize. The key concept of MPT is that by calculating the risk, expected returns, and correlations between stocks one can choose a portfolio (a group of holdings in different stocks) to maximize his expected returns. His methods, now standard at every hedge-fund business rely on optimizing complex multi variable equation solver. Hence, many economists focus on coming up with ways to convert the equations to convex optimizations problems.

KEYWORDS: Portfolio optimization, Convex functions, CVXPY.

Prelude

This document will be censored for academic honesty concerns, an uncensored version will be made available upon request after all projects are submitted. A censored text looks like this: [redacted].

The author is grateful to the many programmers who compiled their knowledge and made their great resources available on the web, the interested reader is encouraged to check out:

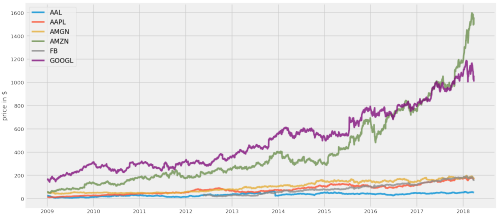
- [redacted].

- [redacted] blog post at [redacted] on how to use the quandl API.

Finally, we are grateful to the authors of [redacted] for making their data available and to [redacted] for collecting and hosting the data at the [redacted].

1. Introduction

Using a python API package such as quandl it is possible to get a table of stock prices for a given time period[1]. For instance, I can find out the symbols for S&P 500 traded companies such as Apple, Amazon, Google, Facebook, American Airlines, and Amgen Inc. (AAPL, AMZN, GOOGL, FB, AAL, and AMGN respectively). I use the function `getPortfolio` to get the closing price of the different stocks[2] and the function `plotPrices` to plot the prices.




```
stock_table.loc['2017-01-01':'2018-01-01']
```

date	AAL	AAPL	AMGN	AMZN	FB	GOOGL
2017-01-03	46.004869	114.715378	147.636176	753.67	116.86	808.01
2017-01-04	46.402320	114.586983	149.732251	757.18	118.69	807.77
2017-01-05	45.597483	115.169696	149.839993	780.45	120.67	813.02
2017-01-06	45.915443	116.453639	153.561996	795.99	123.41	825.21
2017-01-09	46.779898	117.520300	155.579713	796.92	124.90	827.18
2017-01-10	48.170973	117.638817	155.501355	795.90	124.35	826.01
2017-01-11	48.329954	118.270913	153.405280	799.02	126.09	829.86
2017-01-12	47.793396	117.777088	152.171142	813.64	126.62	829.53
2017-01-13	47.346264	117.569682	152.915543	817.14	128.34	830.94

FIG. 1: *Left: Stock closing price per date. Right: same prices plotted over time.*

However, we are more interested in the *returns*. The daily return of a stock is the ratio of change in stock price. Specifically if in the previous day the price was x and today the price of the stock is y then the return is $\frac{y-x}{x}$.

Problem:get_returns

compute the returns at the plotReturns function.

The return is useful because for every dollar you have invested in a stock

[1] To be able to use quandl API you should have key, you can get one by creating a a (free) account at https://www.quandl.com/

[2] Note, some cell values might be `NaN` if the company was not traded at that particular date

yesterday, your wealth will be dollar+return by the end of today. The output of `plotReturns` shows how the daily return changes for the different stocks[3].

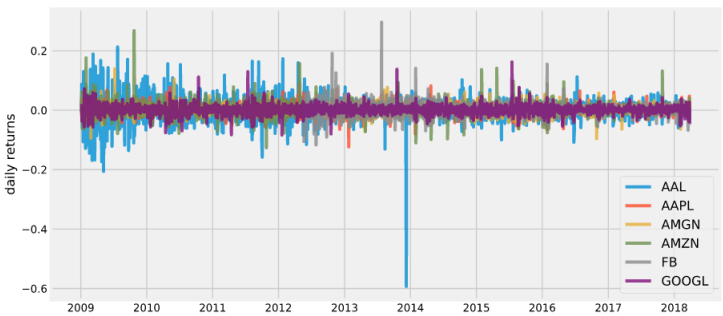


FIG. 2: *The daily returns for the different stocks.*

Another measure we are interested in is the standard deviation of the daily returns. This is a measure of risk. A high standard deviation indicates a high variation in values. Even if the expected (mean) return of a stock is high, an investor might be reluctant to buy a stock with a high standard deviation as it means that low returns are as likely as high ones.

Note, we want the standard deviation of the return, not the original price.

Finally, we are also interested with in the correlation between two stocks. If two stocks go up and down together then investing only the two of them is risky. If we invest instead in two stocks that move somewhat independently then even if one stock plummets the other is likely to remain solid.

[3] The easily visible AAL dip in December 9th 2013 coincides with a 0/1 split, now we know we did not make a mistake!

```
Problem:get_mean_returns
Problem:get_stds
Problem:corr
```

```
Compute the expected (mean) returns, the standard
deviation, and the correlation matrix for the stocks
at the portfolioWriter function.
The function write the number of stocks, the expected
return and standard deviation for each stocks, and
then the correlation matrix in table format.
```

The file *myport.in* contains the number of stocks, the return and mean standard deviation for each stock, and the correlation matrix between the stocks in a Coordinate list (COO) format.

```
Problem:ret_sdv
Problem:cov
```

```
complete the portfolioReader function to return the
number of stocks, and arrays of mean return values,
mean standard deviation values, and a covariance
matrix.
```

2. Portfolio Optimization

With the definitions complete, and the measures stored in the different portfolios understood. It is time to start investing.

2.1. No holds Bar!

Modern portfolio theory (MPT) focuses on maximizing the portfolio given the previously introduced measures (expected return, standard deviation, and correlation matrix) and some constraints on the allowed trades. Specifically we want to calculate a vector of weights $w = (w_1, w_2, \ldots, w_N)$ such that if we allocate the w_i resources to stock i then we get the sum of returns. In other words our *objective function* 'ret' can be written in terms of w and the expected returns

$expected_ret = (ret_1, ret_2, \ldots, ret_N)$:

$$ret = expected\_ret^T \times w = \sum_{i=1}^{N} ret_i w_i \tag{1}$$

A clear constraint is that the total sum of the weights should be 1. In other words, we want to find out what portions of our wealth should be invested in the stock. The constraints should be given to the *cvxpy* in a list. Since we have only one constraint we the list will have only one entry.

$$constraints = [\blacksquare] \tag{2}$$

```
Problem:ret
Problem:risk
Problem:constraint_1
Problem:objective_1
```

```
In the 'No holds bar' section, write the code for
the objective measure 'ret' and the constraint we
previously defined.
```

Now we can use *cvxpy* to solve the optimization problem. Use `prob.solve(warm_start=False)` if you have done everything correctly and with value from `portfolioReader('./myport.in')` you should get around that the expected return is around 1446.64

2.2. Long holds only!

If you print the w vector that gives you the optimum for the last section you will notice that 1. The values are in the millions 2. Some values are negative. This means that we have a *short hold* we are borrowing money by "owing" a stock so we can invest in a different stock. In contrast, a *long hold* means that we invest in a stock hoping it goes up (if confused, see investopedia for an excellent explanation). Mathematically, a short holds of stock i means that $w_i < 0$. Some investors allow only long stocks. This means that we need to add an additional constraint:

$$constraints = [\blacksquare] \tag{3}$$

Problem:constraints_long_hold

Fill in the constraints with long hold only.

Use `prob.solve(warm_start=False)` if you have done everything correctly and with value from `portfolioReader('./myport.in')` you should get around that the expected return is around 0.001654.

2.3. Risky Business!

Look at your w values, you will find that most are zero (or negative but extremely small) and only one stock is positive (specifically the one with the highest return, in our case, facebook). This is a huge risk as we are putting all our money on one company. This is where *diversification* comes into play.

MPT defines risk as $\sum_{i=1}^{N}\sum_{j=1}^{N} w_i cov(ret_i, ret_j) w_j$ where ret_i is the expected return for stock i. We previously calculated the covariance matrix $COV(ret, ret)$ and saved it in our file. We can use it to calculate the risk in matrix form:

$$risk = [\blacksquare] \tag{4}$$

Now our objective function is not only to maximize the return, but also to minimize the risk. We use the parameter γ to choose how risk averse the model should be.

$$objective = [ret - \gamma \cdot risk] \tag{5}$$

Problem:risk
Problem:objective_1

Fill in the risk and objective functions.

We can use the probOpt function to iterate over the γ values and then plot the *trade-off*. This curve demonstrates how the expected return changes if we accept higher risks. Note that the highest risk (the highest point of the curve) is the same solution we found at the previous step.

Additionally, we can plot the holdings (w values) for each of the optimal solutions and visualize how the portfolios change qualitatively. As can be seen

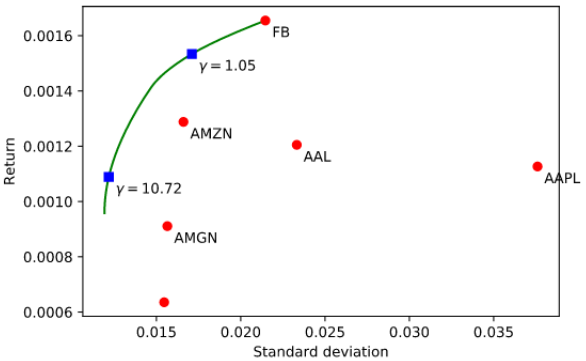


FIG. 3: *The trade off curve, lower γ values indicate riskier portfolios.*

in figure 4, the more risk averse a portfolio is, the more diversification we can expect.

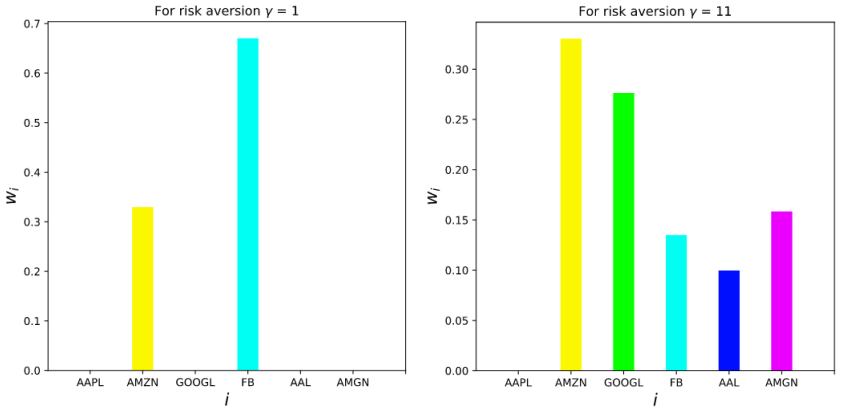


FIG. 4: *Optimal portfolios for different risk aversion*

2.4. Maximum leverage constraints

With risk factored into the picture, we might want to relax our constraints. Instead of having only long holds we will now allow some short holds but limit our

leverage. In other words we will allow the w vector to contains some negative values. Consider the following options:

- $\|w\|_1 = L_{max}$ for some L_{max}, meaning the portfolio holdings can be negative but we are not risking more than l_{max} times our wealth in total.

- $\|w\|_1 \leq L_{max}$ meaning our holding can be any combinations as long as we do not owe more than l_{max} times our wealth.

Which of these is a convex constraint? Once you find it add plug it into the constraints list instead of the long-holds only constraints.

Problem:constraints_maximum_leverage

Fill in the maximum leverage constraint.

We can now plot the trade-off curve for the different L_{max} values. As expected, the more we are allowed to leverage our wealth teh more risky the portfolio can become (the more far we can go on the x axis) and the higher returns we can expect.

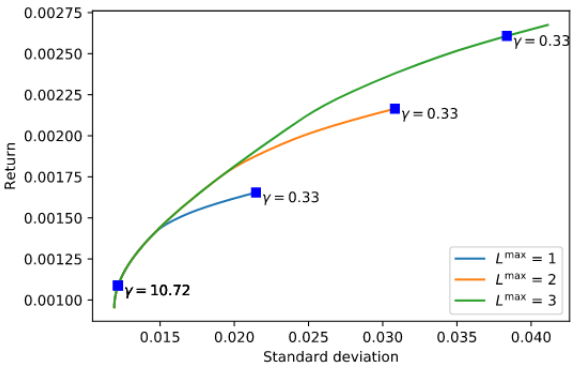


FIG. 5: *The trade off curve, lower γ values indicate riskier portfolios.*

3. Portfolio Optimization with transaction costs

In their 2007 seminal paper [2] the authors tackle the problem of portfolio optimization with transaction costs (note, the paper's notation of w and x is the opposite of the one we are using). Specifically, if we have x dollar amounts already in every stock, and w is how much we buy ($w_i > 0$) or sell ($w_i < 0$) in each stock then we have:

$$ret = expected\_ret^T \times (w+x) = \sum_{i=1}^{N} ret_i(w_i + x_i) \tag{6}$$

With the addition of transaction costs we now have a budget constraint. We want to sell enough to be able to both buy the stocks and pay the transaction costs. In other words, if $\phi(w)$ is the total transaction cost we want:

$$\phi(w) + \sum_{i=1}^{N} w_i \leq 0 \tag{7}$$

The transaction cost $\phi(w) = \sum_{i=1}^{N} \phi_i(w_i)$ is a function of the base transaction cost β and the selling commission α and is usually given by:

$$\phi_i(w_i) = \beta_i + \alpha_i \mid w_i \mid \tag{8}$$

This gives rise to a difficult non convex constraint. However, given upper bounds on w_i, for instance $\mid w_i \mid \leq LU$ then we can find the *convex envelop* (smallest containing convex function) of ϕ_i:

$$\phi_i^c(w_i) = (\frac{\beta_i}{LU} + \alpha_i) \mid w_i \mid \tag{9}$$

Which relaxes the budget constraint in the sense that it enlarges the search set. Since we enlarge the set then the optimal solution is an upper bound for the original problem.

Problem:constraints_costs
Problem:risk_costs
Problem:ret_costs

Complete the constraints, they are equation 7 with the convex relaxation and the upper and lower bounds on w_i. Also, complete the risk and return function, same as before but now with $w+x$.

The authors of [2] also came up with a method for calculating the lower bound using a different convex relaxation. This is done by iterating with the following change for the cost function function:

- For $k = 0$ solve with the same relaxed constraints as for the upper bound.
- For $k > 0$ define a new cost function:

$$\phi_i^l(w_i) = \left[\frac{\beta_i}{\mid w_i^{k-1} \mid + \delta} + \alpha_i\right] \mid w_i^k \mid \tag{10}$$

And solve with this new constraint (equation 7 with this new function).

Problem:constraints_costs_lower

Complete the constraints, they are equation 7 with the new convex relaxation.

Note that we stop the process after 50 iteration, however the original paper contains a proof of convergence. While this means that we will not find the lower bound for all the values we are still be able to to get the lower bound for the last 5 samples within 50 iterations. Plotting the two shows how the optimum values are almost identical (see figure 6). In other words, we were able to get tight upper and lower bounds for non-convex function using convex relaxations only.

4. Conclusion

Convex optimization is useful. Moreover, this seemingly mundane concept is at the very heart of cutting edge of economic theory. We hope you enjoyed

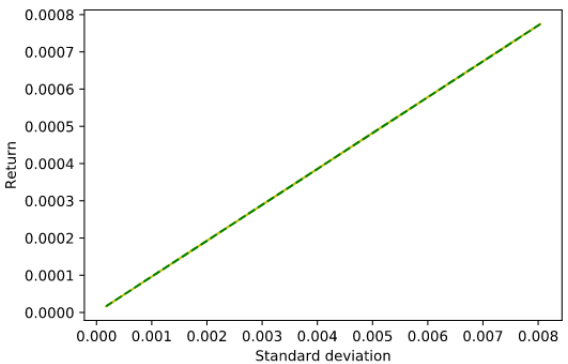


FIG. 6: *The upper bound (green) and lower bound (yellow). The average of the green-yellow curves is 2.375e-07.*

learning some of the real world applications of the material you study in class.

5. Appendix: Extra credit

Pick 10 stocks and use the long holds + risk optimization constraints and objective function to find the best weights. Send me the weights and stock symbol list, as well as the *gamma* you used. A competition will be held and the entry that gives the best returns for march 2019 will be rewarded with 4 points to the final grade (and an invite to the Robinhood app). A few caveats: The portfolio can not contain *Aerospace Defense* or *Tobacco* companies.

You will need to call the function `getPortfolio(['AAPL','AMZN',...])` with your list of stocks to get your own `stock_table`.

REFERENCES

[1] T.-J. Chang, N. Meade, J. E. Beasley, and Y. M. Sharaiha, "Heuristics for cardinality constrained portfolio optimisation," *Computers OR*, vol. 27, pp. 1271–1302, 2000.

[2] M. S. Lobo, M. Fazel, and S. Boyd, “Portfolio optimization with linear and fixed transaction costs,” *Annals OR*, vol. 152, pp. 341–365, 2007.