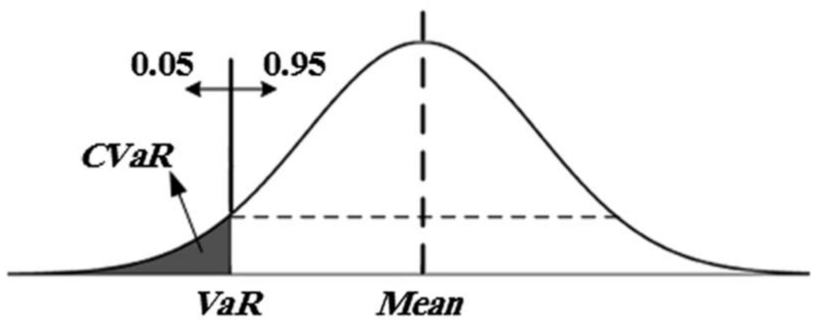
Optimization
Project 1 – Linear Programming

Deliverables

One python file and one pdf file, submitted to Canvas. Your report should go into some detail about how you solved the problem, include some graphs that explain your results, and include relevant code chunks in the final output. 66% of your grade will be based on whether you get the problem right or not, the remaining 34% will be based on the quality of the presentation of your analysis. We will re-run your code with a new data set. If you don't get the right answer or your python file doesn't run, we will go through your code and give partial credit accordingly. The easier it is to read your code the easier it is for us to understand what you're doing, so use a lot of comments in your code!

Problem Overview

In portfolio optimization, a popular way to measure and minimize risk is to use Value-at-Risk (VaR), which quantifies the maximum loss expected over a specific time period at a given confidence level. While intuitive, VaR has undesirable mathematical characteristics such as a lack of subadditivity and convexity. As an alternative measure of risk, Conditional Value-at-Risk (CVaR), also called Mean Excess Loss, Expected Shortfall, or Tail VaR, which is the conditional expectation of losses above VaR, is known to have better properties than VaR.



For example, in the image above the distribution of a portfolio's return may follow the normal distribution. The 95% VaR represents the value above which 95% of returns will occur and below which 5% of returns will occur. The cVaR represents the average value of all returns below the VaR. Sometimes VaR and cVaR are calculated with respect to "losses" instead of "returns," so the image would just be mirrored in that case because losses are just the negative of returns. Here we call this percentage β and different levels of β correspond to different measures of risk.

In this project, we will build a linear program that builds a portfolio of stocks that minimizes the cVaR risk of our portfolio and calculate VaR simultaneously. We will determine portfolio allocations using 2019 stock return data and examine this portfolio using 2020 stock return data. We will study how the choice of confidence level β (like the 95% above) affects portfolio composition and risk.

Conditional Value-at-Risk (CVaR) Model

Let $\boldsymbol{x}$ be decision variables such that $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ where x_j represents the weight in our portfolio of each stock. Let $\boldsymbol{y}$ denote the random vector of returns of the stocks over a chosen horizon, which is independent of $\boldsymbol{x}$ and has probability density $p(\boldsymbol{y})$. The return on a portfolio $\boldsymbol{x}$ is the sum of the returns on the individual stocks in the portfolio, multiplied by the weights x_j. The portfolio's return is given therefore by

$$x_1 y_1 + \cdots + x_n y_n \; = \; \boldsymbol{x}^T \boldsymbol{y}$$

and the portfolio's loss is given by $-\boldsymbol{x}^T \boldsymbol{y}$.

For a given set of portfolio weights, $\boldsymbol{x}$, it can be shown that the β-CVaR of the portfolio is calculated as

$$F_\beta(\boldsymbol{x}) = \min_{\alpha} \alpha + (1-\beta)^{-1} \int_{\boldsymbol{y} \in \mathbb{R}^n} [-\boldsymbol{x}^T \boldsymbol{y} - \alpha]^+ \, p(\boldsymbol{y}) \, d\boldsymbol{y}$$

where $[t]^+ = max(t, 0)$. We will eventually treat β as a hyperparameter and examine the out-of-sample performance based on our choice of β.

In practice, we approximate the integral by sampling the probability distribution in $\boldsymbol{y}$ by using real returns data. Suppose we have a collection of returns vectors $\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_q$. Here each vector $\boldsymbol{y}_k$ represents the returns of several stocks over one time period. Then, the approximation of $F_\beta(\boldsymbol{x})$ is

$$\tilde{F}_\beta(\boldsymbol{x}) = \min_{\alpha} \; \alpha + \frac{1}{(1-\beta)q} \sum_{k=1}^{q} [-\boldsymbol{x}^T \boldsymbol{y}_k - \alpha]^+$$

This objective is nonlinear due to the $[\cdot]^+$ terms. We can make this linear by introducing auxiliary variables u_k, such that $u_k \geq 0$ and $u_k \geq -\boldsymbol{x}^T \boldsymbol{y}_k - \alpha$; as in

$$\tilde{F}_\beta(\boldsymbol{x}) = \min_{\alpha, u} \; \alpha + \frac{1}{(1-\beta)q} \sum_{k=1}^{q} u_k$$

such that

$$u_k \geq -\boldsymbol{x}^T \boldsymbol{y}_k - \alpha$$

$$u_k \geq 0$$

Our goal is to find the portfolio of stocks, $\boldsymbol{x}$, such that $\tilde{F}_\beta(\boldsymbol{x})$ is as small as possible. To do this all we must do is set $\boldsymbol{x}$ to be more decision variables in the problem. We further constrain $\boldsymbol{x}$ to satisfy

1. The portfolio weights should sum to 1 and be nonnegative.

2. Only portfolios that can be expected to return at least a given amount R will be admitted: $\boldsymbol{x}^T \overline{\boldsymbol{y}} \geq R$, where $\overline{\boldsymbol{y}}$ is the vector of mean returns of each stock, $\boldsymbol{y}$.

Specifics

1) Attached to this project are two csv files with stock price data (you need to calculate the percentage return yourself) from 100 stocks in 2019 and 2020. We are considering 2019 – 2020 because in 2019 the world was relatively normal and then in 2020 the covid pandemic hit and the market went a little crazy, making this an excellent case study for the effectiveness of risk management. The NDX column is the Nasdaq index, and the other columns consist of the stocks that make up the index – the NDX should not be considered as part of your portfolio. You will use the data from 2019 to build a vector of portfolio weights, $\boldsymbol{x}$, and then see how that portfolio would have performed in 2020.

2) Find the portfolio that minimizes the daily average β-CVaR using the 2019 data. Use β=0.95 and R=0.02%. If you keep using the 2019 portfolio in 2020, what is the daily average β-CVaR in 2020? Is it a good idea to stick with the same portfolio across different years? Briefly justify using the differences between in-sample and out-of-sample daily average CVaR, and comment on potential non-stationarity.
    a. To calculate the cVaR using 2020 data and the 2019 portfolio you can re-evaluate $\tilde{F}_\beta(\boldsymbol{x})$ using the ***x*** from 2019, and the ***y'***s from 2020.
    b. What is the cVaR of the NDX index over these time periods?
3) Rerun your portfolio model using the 2019 data with β=0.90 and β=0.99. How do different β values affect the portfolio allocation?
4) Your team proposes a conservative risk management approach: instead of minimizing the average risk over time, focus on minimizing the maximum monthly β-CVaR across the year. By controlling the single worst month's tail loss, the portfolio will be better protected against extreme adverse conditions. Run your model again using the 2019 data with this objective. Compare with Part 2.
5) After some research, your boss has requested that the portfolio be updated every month instead of keeping the same allocation all year. The reason is that market conditions and asset performance can vary significantly from month to month—for example, seasonal effects, earnings announcements, and macroeconomic events may shift the relative attractiveness of different assets. To address this, you need to run the CVaR optimization model in each month in 2020. You will obtain a sequence of portfolios. Please evaluate the average daily CVaR, its variation and minimum and maximum monthly CVaR and then compare with Part 2. Is it worthwhile to reoptimize across months?
    a. For each portfolio just use 1 year worth of daily returns. For example, to pick your January 2020 portfolio you should use data from January 2019 – December 2019. To pick your February 2020 portfolio you should use data from February 2019 – January 2020.
6) A stable portfolio is defined as a monthly allocation such that, for each instrument, the change in portfolio weight from one month to the next is no more than 5 percentage points (0.05 in weight). Is the sequence of monthly portfolios you found stable? If it is not stable, you do not need to solve a new optimization model. Instead, describe how you might add constraints to the CVaR optimization model to enforce stability in portfolio weights over time.
7) Write a pdf file that summarizes your analysis with graphs, text, and code chunks. Pretend like this is a report you will deliver to your boss making a recommendation on how to construct low-risk portfolios. You will be graded on the quality of the presentation of results and justification.
8) The first few lines of the python file, in the first code chunk at the beginning, should include pd.read_csv call that reads the csv file that has the return data. Be sure to include a noticeable comment that lets us know where you read the csv files. The template csv files fit the correct format. To be graded, we will change this pd.read_csv call to load a new csv file and re-run your code to see if you get the right answer on new data. Failure to run will automatically reduce your grade by 10 percentage points! Be sure that all your analysis in the python code file is generalized, so when we load the new csv files the output will be for the new data, instead of the template data. That means you should not hard code any numbers. Instead, you should reference variable names for your output.

Notes

- The problem formulation used in this project comes from a paper called "Optimization of Conditional Value-at-Risk" by R. Tyrrell Rockafellar and Stanislav Uryasev.