CASE 2: Portfolio Optimization Asset Return Prediction

Introduction

In this case, you are a portfolio analyst tasked with allocating a fund over a ten-year time horizon across six stocks your research team has selected. To aid you, your company has provided you with ten years of historical price data for each stock. Your goal is to develop an algorithm to construct and rebalance a portfolio aimed at maximizing returns while simultaneously minimizing returns variance. The trading begins the day after the last day of data provided.

You will be provided with a CSV/dataframe consisting of intraday prices, and you are given 30 ticks per day. New days start at dataframe indices 0, 30, 60 etc and the intraday ticks are evenly distributed throughout the trading day.

You will be given 5 years worth of financial information (i.e. 1300 days) and be tested based on your performance on the next 6 months of data (i.e. 130 days).

Education

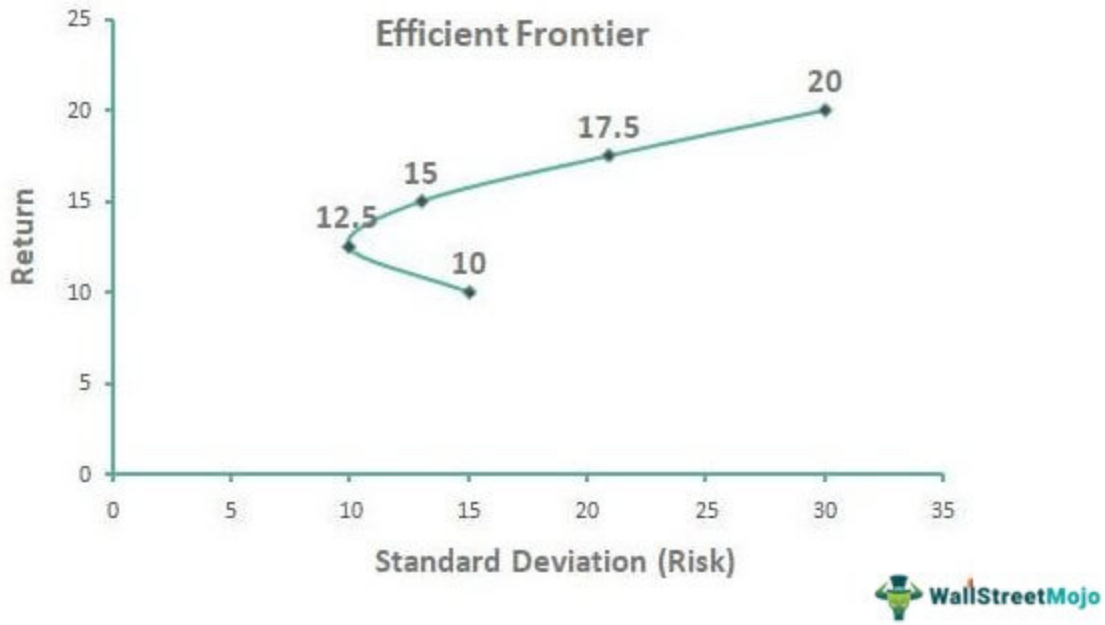
Equal Weight Portfolio

The naive portfolio allocation is giving each stock in your portfolio a weighting of 1/n, assuming n distinct assets. This method assumes that your portfolio's assets aren't correlated with one another, and has many downsides. However, it's relatively straightforward to implement.

We strongly recommend implementing this strategy in your initial explorations of the question. Use this method as a benchmark for future testing; if your new strategies are being outperformed by an allocation of 1/n, we recommend further iterations.

Markowitz

In 1952, Harry Markowitz of the University of Chicago published "Portfolio Selection," which formalized this basic intuition and formed the basis of modern portfolio theory (MPT). Markowitz conceived of an efficient frontier of portfolios that maximize return given a certain level of risk (or alternatively minimize risk given a certain desired return). Along this efficient frontier, the actual portfolio chosen is based on the individual investor's level of risk aversion (high risk aversion = low level of risk, and vice versa). In the absence of exact knowledge on risk aversion, portfolio managers can choose an efficient frontier portfolio based on the specific objectives given to them by their clients or firms.



Obviously, in order to implement a Markowitz portfolio, investors need some estimate of expected return and an estimate of risk. Additionally, we need some way to estimate the correlations, or covariances, between the returns of the different assets. Intuitively, a portfolio where all asset returns are highly correlated should have more risk than one with the same weights and risks for each individual asset but where all the asset returns are uncorrelated due to the fact that in the former case, one asset losing a large portion of its value means the entire portfolio likely will while this is not true in the latter case. Indeed, we see that the variance of returns in a portfolio (a measure of risk) is given by

$$\begin{bmatrix} w_1 & \cdots & w_n \end{bmatrix} \begin{bmatrix} \mathrm{Var}(r_1) & \mathrm{Cov}(r_1, r_2) & \cdots & \mathrm{Cov}(r_1, r_n) \\ \mathrm{Cov}(r_2, r_1) & \mathrm{Var}(r_2) & \cdots & \mathrm{Cov}(r_2, r_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(r_n, r_1) & \mathrm{Cov}(r_n, r_2) & \cdots & \mathrm{Var}(r_n) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix}$$

where w_i refers to the weight of the ith asset in the portfolio, r_i refers to the return of that asset, and n refers to the number of assets in the portfolio. While we can reasonably estimate variances and covariances of asset returns based on historical data, empirically, historical returns have not been a strong predictor of future expected return and large covariance matrix estimates tend to be numerically unstable, leading to practical difficulties in implementing a Markowitz portfolio. Since the development of MPT, investors and researchers have looked for new ways of dealing with this problem.

A downside of the Markowitz approach is that it works best on longer time horizons. Intraday price movements are often incredibly

volatile and unpredictable- Markowitz based implementations are best used for trading at a 'macro' level rather than reallocating on a daily level.

Risk Parity Allocation (RPA)

One approach to portfolio allocation is to simply ignore expected returns when determining how to allocate assets in a portfolio. The most naive way of doing this would be to simply find the portfolio with the lowest predicted risk, but this would simply be to invest in a risk free asset, and the return offered by such a portfolio would not entice many investors. What we want instead is a way to have a portfolio full of risky assets where more weight is given to those with lower risk. This can be done by ensuring that every asset's risk contribution to the portfolio is equal, meaning that less risky assets have more weight than riskier ones in order to equalize the risk contribution. Here, risk contribution is calculated as

$$\frac{w_i(\sum \mathbf{w})_i}{\sqrt{\mathbf{w}^T \sum \mathbf{w}}}$$

where w is the column vector of weights, and ∑ is the variance-covariance matrix from the previous equation (not a summation of w). In this calculation, assets that are uncorrelated with the rest of the portfolio contribute less to risk than correlated assets with the same risk level, which is intuitive since a drop in the value of an uncorrelated asset does not contribute as much to overall losses for the portfolio as would a correlated asset. As it has been shown that asset volatilities and correlations are relatively stable over time, historical risk contribution can be used as a reliable estimate for risk contribution.

Case Specifications and Rules

The exchange trading platform will **not** be used for this case. Teams are expected to develop their strategies using our Python stub code and submit their code before the competition.

We will run each competitor's portfolio allocation algorithm on a test dataset with data generated by the same process as the data you are given; **this test dataset will immediately follow the period in the training dataset.**

There will be one round, the results of which will be computed prior to the competition and played back during the competition as if unfolding in real time. As such, you must submit your final code to the case writers beforehand.

In each timestep, asset prices will be provided and teams will submit portfolio allocations among the available assets for that period. **Your algorithm should submit a new vector of weights every day.** These allocations will be in the form of weights on each stock: weights can be positive, negative, or zero in each timestep, but **your weights must be in the range [-1, 1]. Note the absolute magnitudes of your weights don't matter if the relative magnitudes are the same since we are grading on Sharpe. We only place limits for simplicity of grading. For more details ask on Ed.** We assume there are no exchange fees or bid-ask spreads in the market and no liquidity concerns to deal with in allocating your portfolio.

You may use any packages (and any programming language) to study the training data we will provide, but the submitted portfolio allocation code must be in Python and will be restricted in dependencies. The environment used to run submitted competitor code will be Python 3.10 and will only have the NumPy, pandas, scikit-learn, and SciPy packages installed (alongside base Python). Although advanced and/or complex machine learning techniques are interesting to study and are valuable to learn, they are not the focus of this case and are not required for the purposes of solving this case. Deep quantitative research lies not in fancy methods, but smart applications of simple ones.

We strongly advise that you test your submission using a similar environment on your local machine before submitting your final code; submitted code that does not compile or that fails to run for any time step will be disqualified for this case and the team that submitted it will receive 0 points. Before final submission, there will be an opportunity to test if your code compiles and runs properly (note that the Sharpes yielded from this test will not be indicative of what your actual Sharpe ratio will be)

Scoring

Teams will be ranked based on their annual **Sharpe ratio** over the 5 year test period on the return percent series of the portfolio. Normally, we calculate the Sharpe ratio based on the excess return series rather than just the return series, but for our purposes we will assume that the risk free rate means relatively constant and can be ignored.

Case Materials/Data and Code Submission

Python stub code and training data will be released along with the case packet and additional supplementary resources through the UChicago Trading Competition Ed.

We are requiring the final code for this case to be submitted by **11:59 PM CST on Thursday, April 10th**. Note that this is different from Case 1, as we will be computing the results of this round prior to the competition. Code submitted past this deadline will not be accepted, and we reserve the right to disqualify any competitors who submit incomplete code or miss this deadline. Again, we strongly advise that you test your submission in a Python 3.10 environment with only NumPy, pandas, scikit-learn, and SciPy installed before submitting your final code.

Miscellaneous Tips

1. Analyze returns, not prices. Prices of stocks tend to be non-stationary processes, but returns are generally stationary. Analyzing returns series will be more fruitful for your strategies than analyzing price series.
2. Don't test strategies on the same data you train them on. Strategies will likely perform well on data your model has already seen - what's relevant is how well the strategy performs on data the model has not yet seen. You should not necessarily expect that your strategy will perform as well out-of-sample as it will in-sample; holding out a portion of your training data to test on (or running any other procedure to test on new data) is strongly advisable to get a more accurate sense of how successful your strategy will be.
3. Daily movement and intraday price changes are often two very different processes. Make sure you understand what the dynamics of each of these are- portfolio optimization is fundamentally about trading off short-term volatility with long-term predictability.

Questions

For questions regarding Case 2, please post in the UChicago Trading Competition Ed in the "case2" folder.