Machine Learning for Portfolio Optimization Proposal

Team

Stefan Knapik [2]

Renee Zhuang [1]

Mentors

Prof. Mark Fleming [2]

Prof. Wing Kam Liu [2]

Prof. Zhengtao Gan [2]

Chanwook Park [2]

1. Adlai E. Stevenson High School
2. Northwestern University

Summer 2021 Mechanistic Data Science Course – Northwestern University

July 9, 2021

Portfolio Selection

Objective

The goal of this project is to create a model that can evaluate a list of assets to generate an optimized asset distribution within a financial portfolio to maximize expected returns for a given level of risk exposure (aka the Sharpe ratio). Users will be able to input a list of assets into an application and in return, the app will determine the optimal distribution of these assets in a portfolio.

Approach overview

The Sharpe ratio is used to gauge the return per risk of a portfolio or asset, and it is defined as the expected excess return over volatility. Larger Sharpe ratios are preferred by investors because they signify a larger potential for returns with a lesser exposure to risk. We plan to maximize the Sharpe ratio of a portfolio through gradient ascent. By using a neural network to output asset weights within a portfolio, and then using the asset weights and historical asset returns to find historical net portfolio returns, we can derive the Sharpe ratio of the portfolio and its derivatives with respect to the neural network parameters with each iteration of the gradient ascent.

Multimodal data generation and collection

Financial data is readily available from Yahoo Finance and Google Finance. The python library 'yfinance' enables us to fetch historical market data from the Yahoo Finance API.

Feature engineering and dimension reduction

We can use a neural network to extract features and underlying nonlinear patterns from the input asset closing price histories. In the literature, features extracted from deep learning models have performed better than traditional hand crafted features. The objective function (the Sharpe ratio) for gradient ascent is calculated as the mean of historical returns divided by the standard deviation of historical returns.

Regression and classification

We take N assets (stocks and/or ETFs) as inputs. We will use their closing prices for the past K trading days, and calculate their returns for each day. The input to a neural network is made by concatenating information from all assets so it either ends up being of size (K,N) or (K,N,2) depending on whether we only use returns, or use both returns and prices. The hidden layers in the neural network can be tried with different architectures, although Long Short Term Memory appears the most promising. Output values from the neural network are N portfolio weights. We can normalize the raw values of portfolio weights using a Softmax function so that they are all between 0 and 1 and they all sum to 1. We will then use these normalized portfolio weights and the historical return data to compute the realized portfolio returns for each of the K days in the lookback window. From the realized returns we can derive the Sharpe ratio (our objective function) and calculate the gradients of the Sharpe ratio with respect to the model parameters. We then use gradient ascent to update the parameters iteratively and arrive at asset weights within a portfolio such that the overall Sharpe ratio is maximized.

System and design

This model can be used by individual or institutional investors to create diverse and efficient portfolios to help achieve a higher return per risk. An ideal selection of asset proportions is crucial to a portfolio's success and diversification. Machine learning models can be used to inform these decisions.

A limitation of the Sharpe ratio is that it assumes returns are normally distributed. A variation called the Sortino ratio only factors in downside risk by using the downward deviation instead of the standard deviation of returns. The idea behind this is that upside volatility is good for the investor and therefore should not be used in the calculation of risk. Future extensions of this project could maximize the Sortino ratio or any other differentiable parameter.

Expectation

We hope to finish the above within the next 5 weeks, and perhaps continue on to the following project extension if we have extra time.

Stock Prediction (possible project extension)

Objective

The goal of the project extension is to predict which stocks out of a list will outperform their peers for an upcoming period of time. Users will input a list of assets and they will get out a (binary?) parameter for each asset indicating predicted over or under performance. Assets predicted to overperform could then be used as inputs to the above portfolio selection model.

Feature engineering and dimension reduction

We hand pick features to extract after identifying and eliminating redundant features using a correlation matrix. The selected features will span different groups of fundamental data and technical indicators outlined below.

Fundamental Data
- Size: market cap
- Value: price to earnings ratio
- Quality: earnings per share growth, earnings variability
- Profitability: earnings per share, dividend yield, operating margin, free cash flow/market cap
- Growth: asset growth, sales growth

Technical Indicators
- Momentum: momentum for 1, 6, or 12 months
- Moving Averages: log(price/moving average) for 50, 100, or 200 days
- Risk: beta, volatility for 1, 6, or 12 months
- Short Term Reversal: relative strength indices, log(price/bollinger bands)
- Trading Volume: volume

Reduced order modeling and regression

We can use any of a variety of machine learning algorithms to analyze the data (regularized logistic regression and principal component analysis, recurrent or LSTM neural networks, random forest or gradient boosting). LSTM neural networks and gradient boosting seem to be the most promising after reviewing the literature.