Stock Portfolio Optimization using MDPs and Q-Learning

Joshua Chang, Troy Shen

CS221 Artificial Intelligence, Department of Computer Science, Stanford University

Stanford Computer Science

Abstract

Investing firms increasingly apply artificial intelligence techniques to optimize stock portfolios. By utilizing a state-based approach with Q-Learning, our model takes in input data (daily stock opening and closing prices), applies a feature vector, and returns an optimal portfolio within a given time frame. Although the stock market responds to many complex economic forces beyond the scope of our work, our research sheds light on how state-based models can combine external knowledge of the stock market in a feature vector with a reward function to improve portfolio optimization.

Stanford University

Background

The challenge of stock portfolio optimization is a trillion-dollar industry with complex variables. To engage this challenge, top performing quant firms and hedge funds have increasingly used reinforcement learning algorithms to create optimal portfolios.

Problem

Given five years of daily trading data of S&P 500 stocks, build a portfolio whose returns exceed the average movement of the S&P 500 in a given time frame.

Dataset and Features

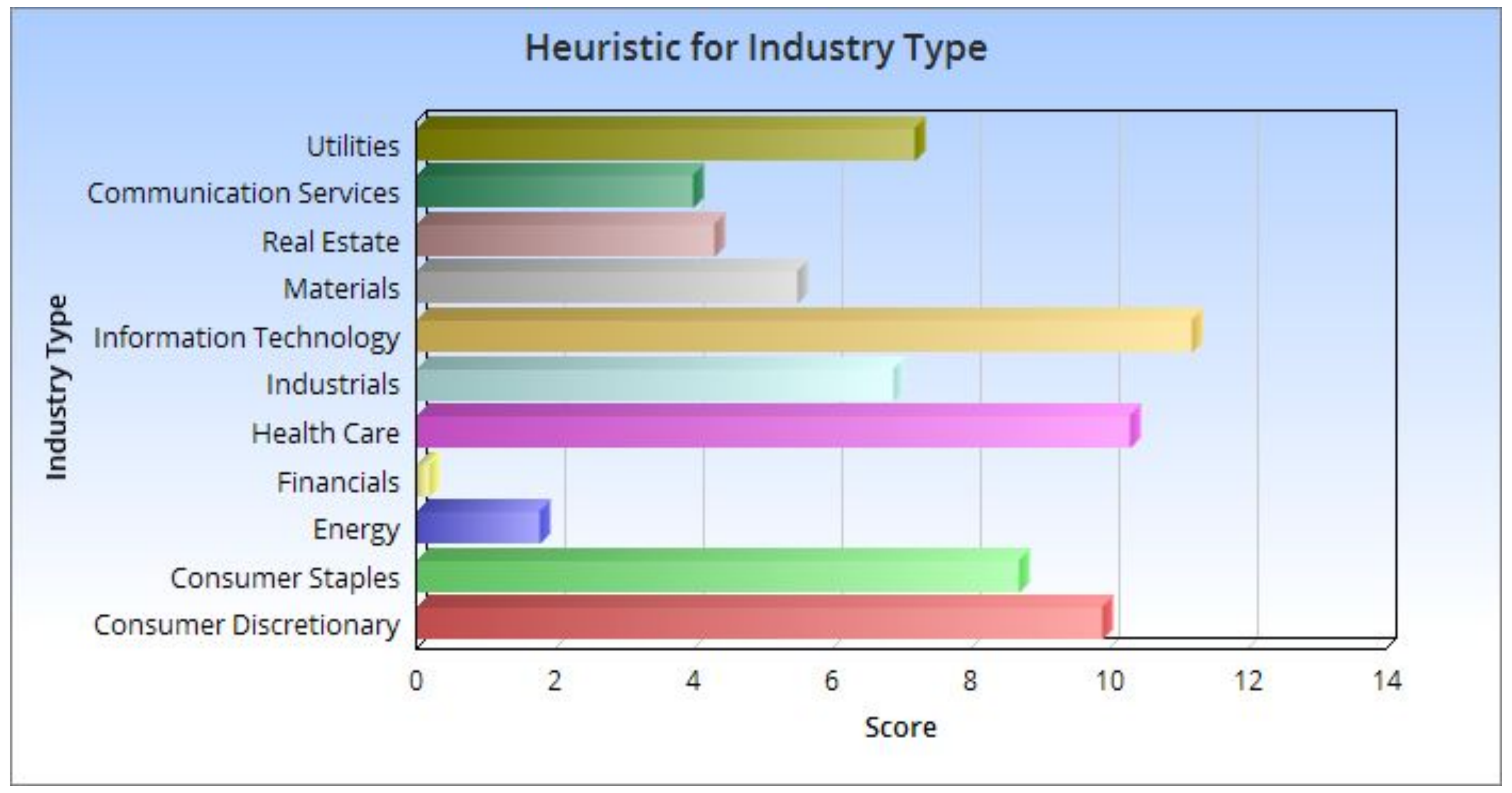
We use five years of daily trading data between February 8, 2013 and July 7, 2018. The trading data includes opening and closing prices, volume, and highs and lows of each stock. Our two primary features are: industry type of each stock weighted to favor more profitable industries in the past decade; standard deviation of returns of the portfolio minimized to favor portfolio diversification.

Model

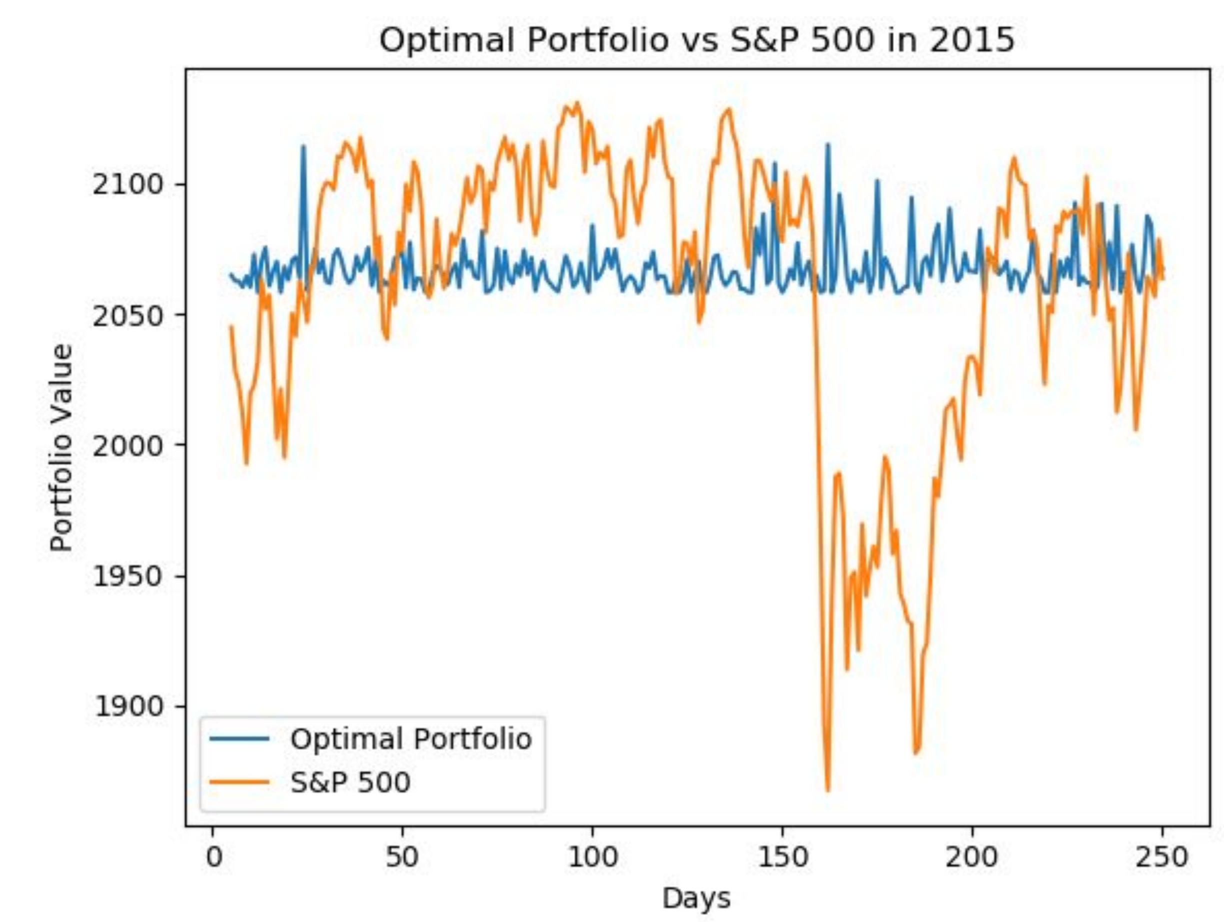
By using the standard deviation of a return of a portfolio, our feature extractor accounts for both the volatility of individual stocks in their moving period as well as the level of portfolio diversification through covariance measurements between stocks. Portfolios with lower standard deviations are favored.

$$\sigma_P = \sqrt{\sum_{i=1}^{N} w_i^2 \cdot \sigma^2(k_i) + \sum_{i=1}^{N}\sum_{i\neq j}^{N} w_i \cdot w_j \cdot \mathrm{Cov}(k_i \cdot k_j)}$$

Our feature extractor also accounts for the viability of an industry type by applying an appropriate heuristic, the average growth of S&P 500 companies by industry type in the past twelve years. Companies whose industry types are more profitable are favored by the algorithm.



Results



Discussion

Because of the difficulty of exploring a large state space, we found that we needed to implement certain constraints to run the model, such as limiting the purchase and selling of securities to a fixed proportion. We measured the performance of our model through the percent gain in our best portfolio within a time interval. We achieved a return of 1.93% on our best model in 2015, compared to the S&P's overall gain of 1.38% in that year.

Future

With more time, we would implement function approximation to generalize state space and thus allow for more exploration. We would also like to include additional features such as stock momentum (measures rates of rise and fall in price) to our feature vector.