Project proposal

In the final capstone project, I chose to design a stock trading program based on the Q learning.

1. Domain background

Stock market has always been volatile and generally varying every now and then. However, people never stop to seek to understand the patterns of trading that allow us to predict future returns of the market.
In this project, I will try to train an automatic stock trader that searches optimal returns using Q-learning.

2. Problem statement & Solution statement

Traditional Q-learning works with an ϵ-greedy exploration strategy and experience replay. However, in stock trading problems, the huge number of possible price combinations will make the search for optimal strategy almost impossible.

Then I got my inspiration from an article of MIT that synthesizes the CNN network and Q learning, using the deep learning to learn Q. I would like to follow the ideas of the article and combine the deep leaning techniques with the reinforcement learning. Thus I plan to predict the values of Q first using neural network, then with the estimated Q value we can do Q learning. The simulation would be displayed several times.

3. Data sets and inputs

The data I plan to use consists of the daily prices of 500 stocks that constitute the S&P 500 index, between the year 2016 and 2021. All price data are downloaded from Yahoo finance. Every chosen stock has the following trading information: Open, High, Low, Close, Volume and Adjusted Close. In addition to it, the prices of SPY will also be one of the inputs of our model, as a performance benchmark of our portfolio.
I will also do some transformations of the raw data, calculating the bollinger bands of multiple window.

4. Evaluation metrics & Benchmarks

Several metrics would be used to achieve my goals. To evaluate the model efficiency, the most direct indicator is the portfolio’s return over that of SPY. Thus we would calculate the total returns of the two during the same period of time. As the ETF can simulate the trends of market index with little tracking errors, I believe the metric will allow me to have an insight in whether my program is really outperforming the market.

The second metric will be the winning rate, measuring the proportion of good trades in all the trades that I make. In the model, I will identify a trade that brings positive returns as a hit. Alternatively, I will call it a miss. Consequently, % of good trades can be calculated as: hits/(hits+miss). The winning rate can show the model's ability to select the right timing.

Finally, I also develop a random trader aiming at controlling bias. The process is similar to a Monte-Carlo simulation. The average return of one hundred random traders will be considered as a proxy of the performance of traders that act randomly, without following any rules. If the Q learning work well, my portfolio should have returns that are above the mean portfolio return plus the variance of the random agent.

5. Project Outline

I will start with processing the data set containing all the daily prices, doing some basic data cleaning work and split it into training and test set. Later, I will randomly choose a certain time horizon. During the time frame, I will set an initial strategy for my model, including the timing to buy, sell or hold stocks. I will also calculate the losses and rewards of the actions so as to train my model to learn trading well. The process will be repeated multiple times in assorted periods while keeping the returns of SPY as a performance benchmark.

Finally, after the model is trained, I will use the data from 2021 to test whether it is useful in out-of-sample situations.

With the results above, I also plan to summarize the performance of different hyper parameters and analyze the reasons.