CS 7646 - Strategy Learner

For this assignment, I have used QLearner developed in the last assignment. The StrategyLearner class has two main methods – addEvidence() and testPolicy().

In the addEvidence method or the learning phase, we get values of our indicators – Price/SMA ratio, Momentum and Bollinger Bands %. We discretize them in a way that each value is binned to an integer in the range 0-9. For each day, for the corresponding discretized value of each indicator, we get the discretized state by combining the concatenating the three integers. During the learning phase, we also loop over each day of the in-sample period, we calculate the reward for our qlearner using the change in price value from the previous day multiplied by the total holdings. Based on the reward and discretized state for the day, we get the action using the query method from our qlearner. Based on the action returned we get the trade for the day. One thing to note here is that three actions for long, cash and nothing are considered for training our qlearner.

This is looped over and over until a maximum number of iterations is reached or a minimum number of iterations has been reached and the trades dataframe remains the same from the last iteration to the current iteration, implying that the policy has converged.

During the out-of-sample testing with testPolicy method, there are a couple of major differences compared to the learning phase in addEvidence. One is that only one iteration over the range of days is performed. Second, we only make use of querysetstate based on the q table generated during the learning phase. The rewards and actions and consequent trades are generated the same way as during the learning phase. This method returns the trades dataframe for the out-of-sample training period.

Experiment 1

Comparing the normalized portfolio values from trades generated by Manual Strategy and Strategy learner with the benchmark, we get the plot below.

Figure 1 Comparing Manual Strategy with Strategy Learner

For this experiment we use in sample date range of January 1, 2008 to December 31, 2009 with JPM. We have used a look back period of 30 days for our Strategy Learner. We have also used marketsimcode from previous assignments to generate portfolio values from trades. While we perform better than benchmark and manual strategy with out q-learner during several periods within in-sample, there is a significant drop in our portfolio value towards the end of 2008.

This anomaly can be explained by a short look back period and unexpected drop in prices for JPM.

As we increase the maximum number of iterations possible and the look back period, we get better results as can be seen below from the plot generated for look back period of 200 days and max iterations of 1000.

Figure 2 Better performance with increased lookback period

Experiment 2

Experiments performed for this part have their code experiment2.py file. We keep the parameters same as that for experiment 1, except we introduce market impact for both manual strategy and our q learning strategy.

We get the following plot for portfolio values generated compared to the benchmark for the same initial values of lookback period and max iterations in experiment 1. We can observe the difference in trading strategies through the normalized portfolio values plotted in the graphs below.

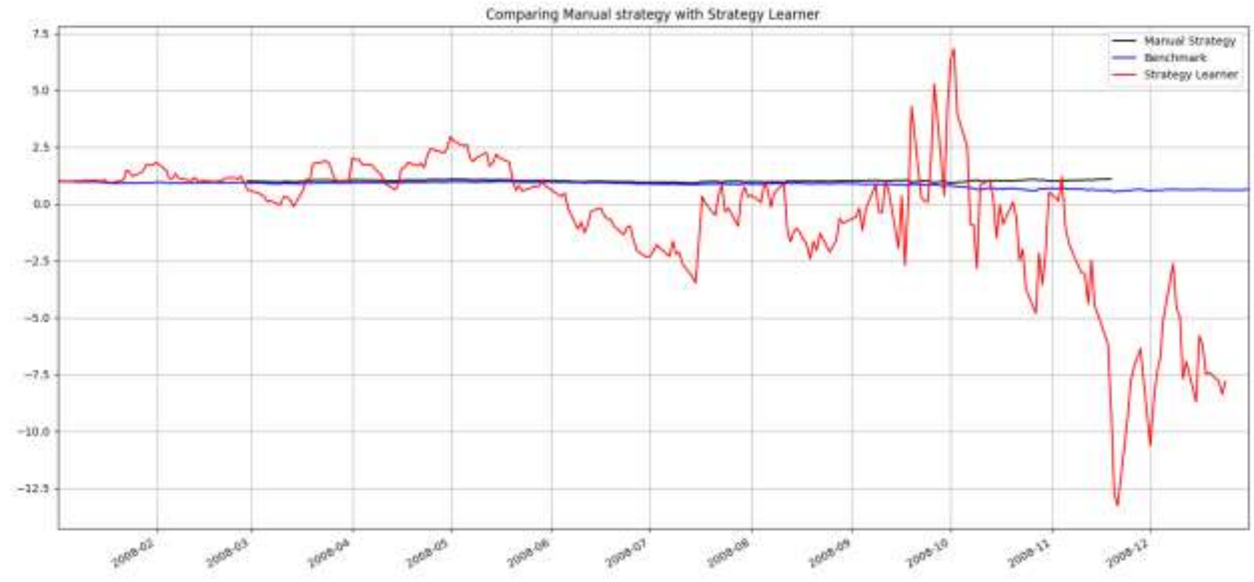

Figure 3 Manual Strategy vs Strategy Learner with Market Impact

With market impact into the picture, the major change is that we could get a negative reward for trading with minor percentage changes in returns.

For results we have used two metrics – sharpe ratio and cumulative returns. The calculated values are given below

Sharpe Ratio Strategy Learner:-0.355321326445

Sharpe Ratio Manual Strategy:0.582100741754

Sharpe Ratio Benchmark:-0.881725328898

Cum Return Strategy Learner:-12.8939552288

Cum Return Manual Strategy:0.0994029303283

Cum Return Benchmark:-0.362391222096