Project 8 Strategy Evaluation Report

Boyuan Liu
bryanliu@gatech.edu

1. INTRODUCTION

In this project, machine learning techniques is explored and implemented in stock market trading strategy. A Q-learning strategy learner was trained using historical trading data and compared to a manual strategy and a benchmark. The impact rate is a crucial factor in large volume trading. The influence of impact rate on the Q-learning strategy learner is also explored in this project.

2. INDICATOR OVERVIEW

Three indicators were selected for both the manual strategy and the strategy learner: momentum, Relative Strength Index(RSI), and Bollinger Band Percentage(BBP). The equations for calculating each indicator is shown below:

Bollinger Band Percentage:

$$upper\ band = SMA[t] + 2 \times std$$
$$lower\ band = SMA[t] - 2 \times std$$
$$BBP = \frac{C_p - lower\ band}{upper\ band - lower\ band}$$

Relative Strength Index:

$$RS = \frac{Average\ Gain}{Average\ Loss}$$
$$RSI = 1 + \frac{1}{1 + RS}$$

Momentum:

$$Momentum = \frac{C_p[t]}{C_p[t - window]} - 1$$

All three indicators were calculated for each trading day with a 14-day look back window. The manual strategy uses a rule-based algorithm to generate holding positions for each training day. The strategy learner uses these indicators to generate states for its Q-table.

3. MANUAL STRATEGY

A long/short/nothing signal generator is implemented in the manual strategy. The signal starts at 0 and it is updated based on the following rules:

BBP enters upper band: signal + 1
BBP exits lower band: signal - 1
RSI > 0.5: signal + 1
RSI <= 0.5: signal - 1
momentum > 0.05: signal + 1
momentum < 0: signal - 1

The manual strategy trader will enter a long position if the signal is greater or equal to 1. It with hold nothing yes the signal is between -1 and 0, end it will enter a short position if the signal is below -1. The logic behind this trading rule is that if more indicators are showing a long signal, then enter a long position. And enter a short position if the indicators are showing a strong short signal. I chose the parameter values based on the theory of those indicators. The performance of the manual strategy trader is compared to the benchmark in Figure 1 below. Vertical blue lines indicate the trader enters a long position and vertical black lines indicate the trader enters a short position.

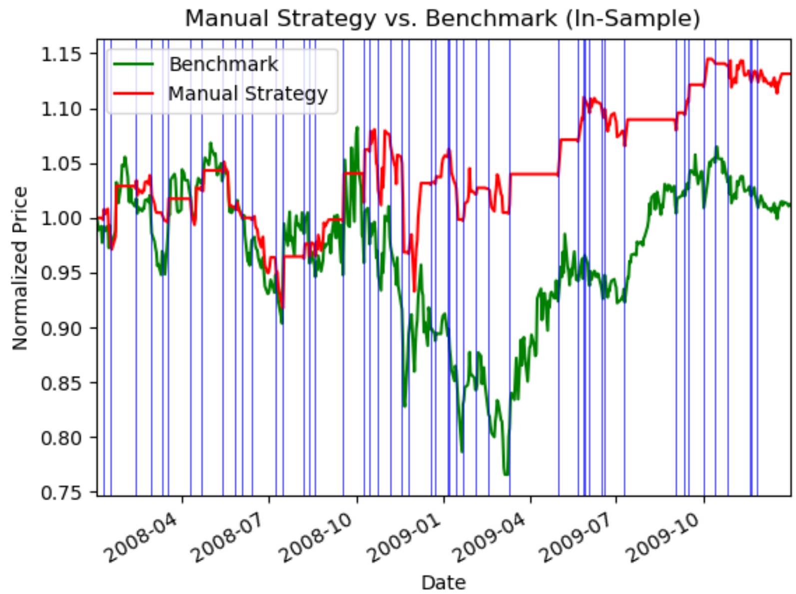

Figure 1-Manual Strategy vs. Benchmark for in-sample data.

As it shows in Figure 1, the manual strategy trader follows the benchmark closely at the beginning of the in-sample period. However, the manual strategy was able to avoid a sharp decline in the stock price and climbed up as the stock price raised. The manual strategy appears to not generate any short positions in this trading period.

The manual strategy trader used the same rule-based algorithm to generate long/short/nothing position signals for out-of-simple data. The out-of-simple performance of the manual strategy trader is compared to the benchmark in Figure 2 below.

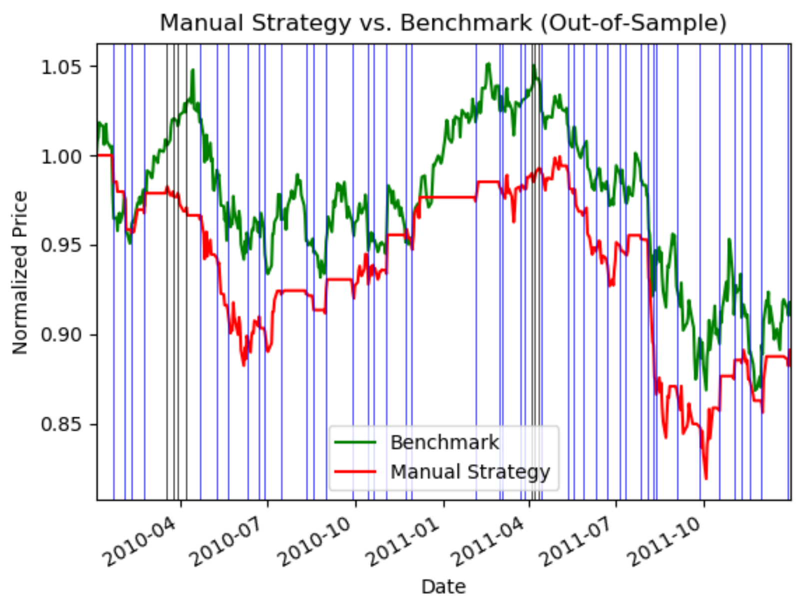

Figure 2-Manual Strategy vs. Benchmark for out-of-sample data.

For the out-of-sample period, the manual strategy did not perform as well as the benchmark overall. The cumulative return of the manual strategy was below the benchmark throughout most of the period. A table is created to quantitatively compare the performers of the manual strategy and the benchmark.

In-Sample	**Manual Strategy**	**Benchmark**
Cumulative return	0.1314	0.0123
STDEV	0.0517	0.0644
Mean	1.0464	0.9681

Out-of-Sample	**Manual Strategy**	**Benchmark**
Cumulative return	-0.1088	-0.0836
STDEV	0.0436	0.0418
Mean	0.9380	0.9742

Table 1-Manual Strategy vs. Benchmark for in-sample and out-of-sample data.

For the in-sample period, the manual strategy has a higher cumulative return and a lower standard deviation. This is because the manual strategy was able to avoid the sharp decline and therefore did not fluctuate as much as the benchmark. For the out-of-sample period, the manual strategy has a very close standard deviation to the benchmark because it followed closely with the benchmark throughout the period and therefore has a similar cumulative return as well.

4. STRATEGY LEARNER

I used a Q-learning model in my strategy learner. The strategy learner is composed of two main pieces: add_evidence and testPolicy. The add_evidence function first creates an instance of the Q-learner. It then calculates the three indicators for each trading day using the indicators.py file. The indicators were bucketized into 10 groups based on its value and mapped to a 1-10 value accordingly. Therefore the Q-learner has a state size of 1000. The function iterates through each trading day to update the Q-table. A positions DataFrame is created to track each update. The model is considered converged if the positions DataFrame does not update anymore. The testPolicy function uses the same discretizing method to convert indicators to Q-table states. It queries the Q-table using the states and generates trading actions based on the results.

The Q-learner has a state size of 1000 and action size of 3 (long/short/nothing). It uses a random action rate of 0.5 and a random action discount rate of 0.99. Initially, I used a learning rate alpha of 0.2 and discount rate gamma of 0.9. After some exploration, I realized that a higher learning rate and lower discount rate can improve the efficiency of the training phase and did not harm the performance of the learner. The learning rate alpha and discount rate gamma currently settle at 0.4 and 0.7 respectively. The Dyna option is not enabled because there wasn't significant performance improvement and it was slowing down the training phase.

5. EXPERIMENT 1

The goal of this experiment is to compare the performance of the strategy learner to the manual strategy and the benchmark. The portfolio values for the benchmark and manual strategy were calculated using the same method mentioned in previous sections. The portfolio value of the strategy learner was obtained by query the trained Q-table and execute trades accordingly. The comparison of strategy learner, manual strategy and benchmark is shown in Figure 3.

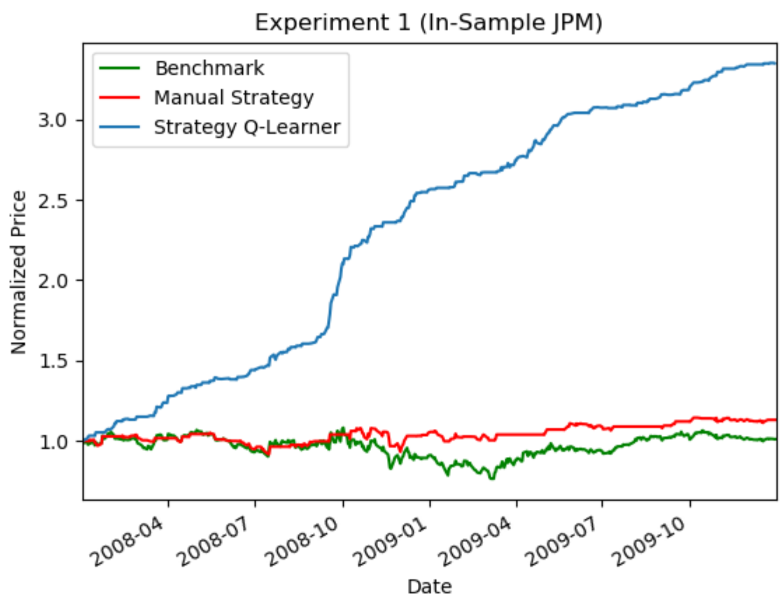

Figure 3-Comparison of strategy learner, manual strategy and benchmark.

The strategy learner performed exceptionally well because it was testing on the in-sample data that it previously trained on. It knows which action has the highest reward by querying the Q-table. I expect the strategy learner to always perform well on in-sample data as long as the training phase converges. This also means that there could be some level of over-fitting in the Q-learner model.

6. EXPERIMENT 2

The goal of this experiment is to study the influence of the impact rate. The impact rate could have significant influence when trading at large volumes. A higher impact rate could result in a lower cumulative return. It could also lead to a fewer number of trades. To further study its influence three strategy learners were trained with different impact rate. The results are plotted in Figure 4.

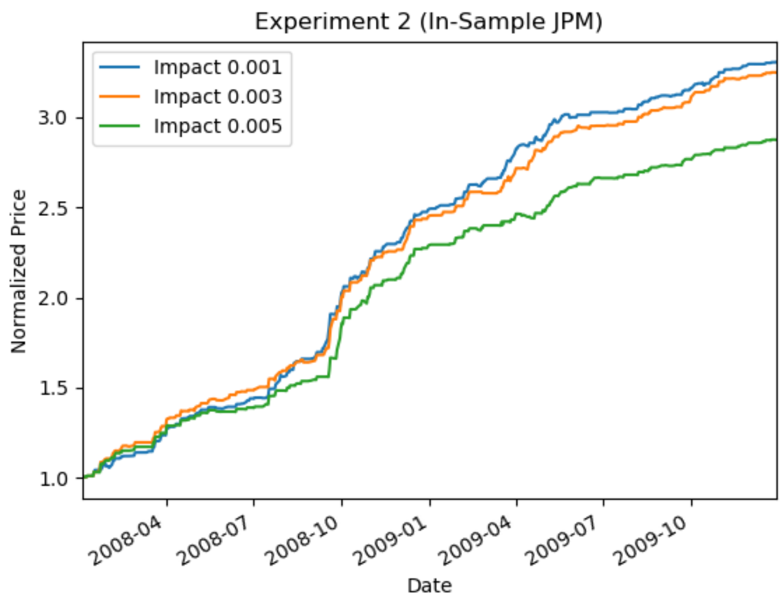

Figure 4-Portfolio value at different impact rate

As it shows in the figure, a higher impact rate resulted in a lower portfolio value. This is because every time a trade is executed, a higher impact rate would result in a higher penalty.

A higher impact rate could also lead to a fewer total number of trades. The total number of trades for the three strategy learner were shown in Figure 5 below.

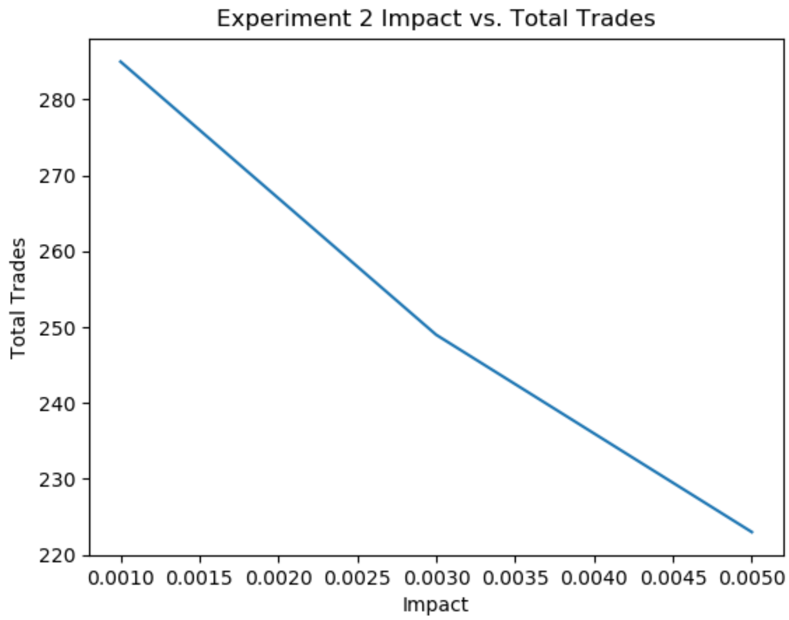

Figure 5-Impact rate vs. Total Trades

As it shows in the figure, strategy learners with a higher impact rate executed less trades throughout the period. This is because sometimes the are only small changes of stock price. If the change in stock price is lower than the impact rate, the trader would lose money by executing any trade. Therefore it is not surprising to see that the higher impact rate resulted in a lower total number of trades.