Project 8 – Strategy Evaluation
CS7646

Wei en Chen
wchen658@gatech.edu

1 INTRODUCTION

Machining learning is widely used for a lot of applications, including financial markets. In this paper, we will explore the performance of manual and reinforced learning strategies using three selected technical indicators. Manual strategy will be analyzed with in-sample and out-of-sample performance. It is very likely the out-of-sample performance will not do as good as in-sample performance because predicting stock prices would need a lot more indicators than three. As for comparing manual strategy and strategy learner (machine learning) performance, it's likely strategy learner would outperform manual strategy because it's easier for computers to find correlations between indicators and stock prices when humans find it harder to do. There will be further analysis on the effect of impact on strategy learner performance. From a thought experiment, increasing impact will likely diminish the cumulative return. This is because number of trades needs to decrease to reduce cost. Reduced number of trades will not be able to accumulate returns when stock prices frequently go up and down.

2 INDICATOR OVERVIEW

Three technical indicators are utilized in this project. The rolling windows in the indicators are based on typical values, and not further optimized [1][2][3][4].

2.1 Bollinger Band Percentage

BBP uses SMA with certain window. For both manual strategy and strategy learner approaches, the window is 20 days. In manual strategy, the hand tuned parameters are the upper and lower thresholds to trigger buy/sell signals. The sell signal is when BBP crosses over 0.8 line and trends lower. The buy signal is when BBP crosses over 0.0 line and trends higher.

2.2 Relative Strength Index

RSI uses a rolling average for calculation. In both manual strategy and strategy learner, the window is 14 days. There are hand tuned parameters in manual strategy to trigger buy/sell signals. It is a sell signal when RSI crosses over 0.75 and trends lower, while it is a buy signal when RSI crosses over 0.23 and trends higher.

2.3 Exponential MACD Oscillator

EMAOsct has 3 rolling averages and 1 factor. In both strategies, the first one is 12 days, to calculate the first EMA. The second one is 26 days, to calculate the second EMA. The third one is 9 days, to calculate the EMA of the difference of the first two EMAs. The last factor is 0.1, used as a smoothing constant. There are hand tuned parameters in manual strategy to trigger buy/sell signals. It is a sell signal when EMAOsct crosses over 0.34 and trends lower, while it is a buy signal when EMAOsct crosses over -0.62 and trends higher.

3 MANUAL STRATEGY

3.1 Input Parameters

It's important to lay out the parameters.

Starting value: $100,000. Stock: JPM. Position: +1000, 0, -1000 shares. In-sample: 2008-01-01 to 2009-12-31. Out-of-sample: 2010-01-01 to 2011-12-31. Impact: 0.005. Commission: $9.95.

3.2 Discussion

To combine all three technical indicators to generate buy/sell signals, I first tuned the threshold parameters for each indicator separately to get decent return over the in-sample period. There is a functionality to assign weight distribution for each indicator. I tested combinations of weights and found that each indicator is equally important so the weight is the same among them. To trigger the buy/sell signal, the manual algorithm just needs any of the three indicators to be triggered with thresholds indicated in previous section. Each buy/sell signal is move position by 1000 shares in increment. This still required some minor fine tuning of the thresholds of each indicator to squeeze more cumulative returns. I believe this is a simple and effective strategy since indicators try to provide when stock price is about to take a change. Using increment of 1000 shares also smoothens the volatility of portfolio values. Figure 1 below illustrates the performance between manual strategy and benchmark (buying 1000 shares on first day and hold it) during in-sample period. Blue/black vertical lines indicate long/short entry points. Manual strategy made a series of good decisions during the stock market crash in late 2008 and early 2009, and beat the benchmark at the end of the period. Table 1 below summarizes the key performance comparison.

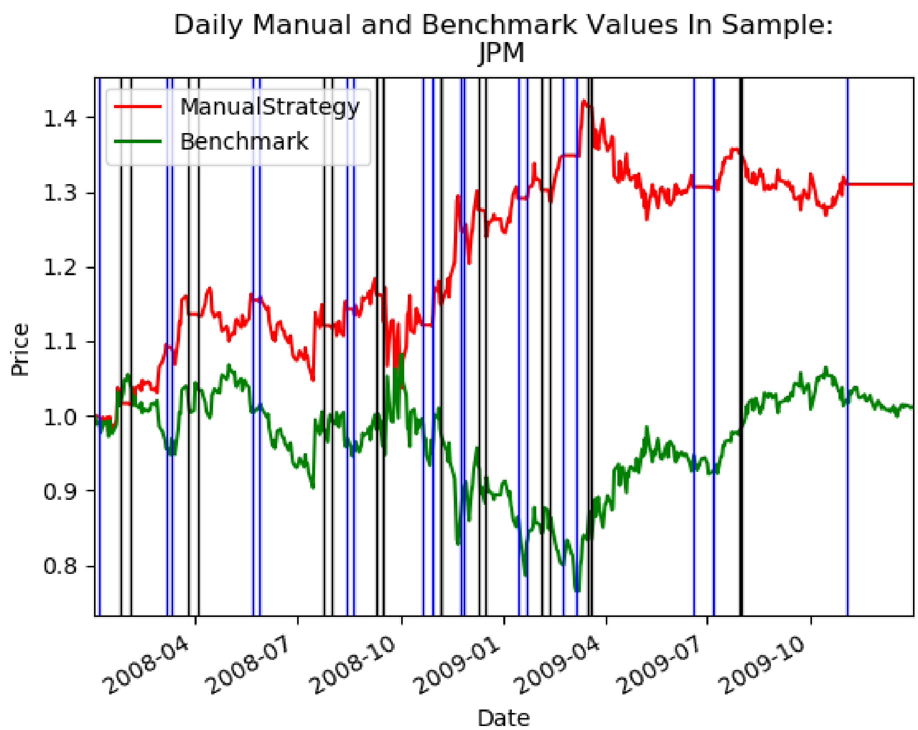


Figure 1 — Performance comparison between manual strategy and benchmark, JPM, 2008-01-01 to 2009-12-31.

Table 1 — In-sample comparison between manual and benchmark performance.

	Sharpe Ratio	**Cum. Return**	**Std Dev**	**Avg DR**	**Final Value**
Manual	0.7839	0.3100	0.0124	0.0006	$131003.85
Benchmark	0.1572	0.0123	0.0170	0.0002	$101027.70

For out-of-sample performance comparison, it was hypothesized that manual strategy wouldn't do as well because the threshold fine tuning occurred on the in-sample period only. In addition, the in-sample period is not on variety of stocks and longer period of time to generalize threshold parameters. Figure 2 below illustrates the out-of-sample (2010-01-01 to 2011-12-31) performance between manual and benchmark.

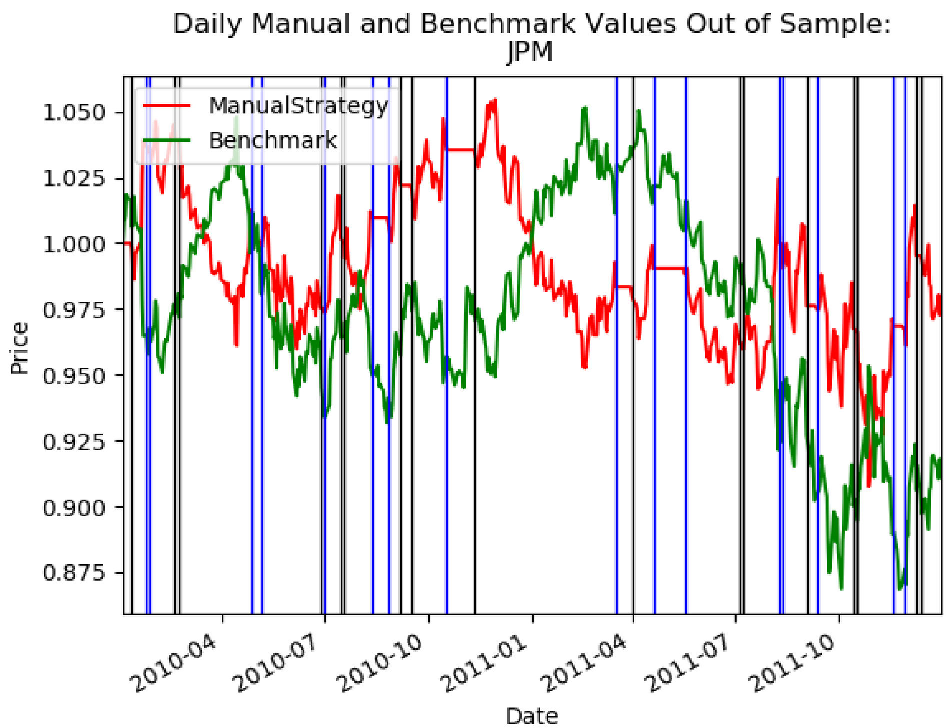


Figure 2 — Performance comparison between manual strategy and benchmark, JPM, 2010-01-01 to 2011-12-31.

As hypothesized, there are mixes of good and back decisions which canceled out the gain. At the end the return is still greater than the benchmark, but it is far from a decent trading algorithm. Table 2 below summarizes the key metrics for comparison between manual and benchmark.

Table 2 — In-sample comparison between manual and benchmark performance.

	Sharpe Ratio	Cum. Return	Std Dev	Avg DR	Final Value
Manual	-0.0481	-0.0256	0.0076	-2.2973	$ 97438.95
Benchmark	-0.2567	-0.0836	0.0085	-0.0001	$ 91445.70

In both in-sample and out-of-sample, manual strategy did perform better than benchmark. It just performed not as great during out-of-sample period from the reasons discussed above. Threshold fine tuning on in-sample only, not enough in-sample for generalization, and more sophisticated manual algorithm may be able to perform better.

4 STRATEGY LEARNER

I used Q learner as strategy learner to recognize buy/sell signals from the same three indicators. For Q learner, which is a type or reinforcement learner, to learn, it needs states, action, and reward.

The states are the discretized values of the indicators. For BBP, most values are between 0 and 1. The bin boundaries are [0.1, 0.2…0.9]. Discretized BBP then goes from 0 to 9. The same setting is applied to RSI since most of RSI values range between 0 and 1. For EMAOsct, most values range from -1 to 1. The bin boundaries are [-0.8, -0.6…0.6, 0.8]. The discretized EMAOsct is then from 0 to 9. To sum all three discretized indicators, discretized BBP is multiplied by 100, and discretized RSI is multiplied by 10. This will results in total of 999 states in the Q table. I did not include any other factor as states because considering there are only ~500 days in-sample for training, 999 states is already likely going to overfit. Adding more to the states is essentially pointless.

There are only three actions, buy, sell, and do nothing. In Q table, sell is 0, do nothing is 1, and buy is 2. They are only converted to -1, 0, and 1 in the learning algorithm for convenience.

For reward, it is the stock price change ratio from current day when action is applied, to the next trading day. If the action is buy or sell, an impact is subtracted from the reward.

Indicators need rolling averages so additional stock prices before in-sample were taken to calculate indicator values starting on the first day of in-sample period. First possible trade can only happen on the first day of in-sample. There are also some hyperparameters to tune. Alpha=0.2 (learning rate), gamma=0.7 (future reward discount rate), rar=0.1 (random action rate), radr=0.99 (random action decay rate), epochs=25 (max number of training iterations over in-sample period), converge=3 (min number epochs of unchanged Q table before exiting learning algorithm). To arrive at these values, arbitrary ones were assigned first to run. Each hyperparameter was adjusted manually to see the effect on speed to convergence and performance in portfolio return. Lower alpha does provide more stable convergence so 0.2 was chosen. Higher gamma allows faster convergence, so 0.7 was chosen. Rar and radr of 0.1 and 0.99 are chosen so there are not too many random actions to swing the final return too much while maintaining some decay rate to explore. Epochs and converge of 25 and 3 are chosen to ensure the algorithm will not time out, and if the results converge early, it can exit early to save time.

5 EXPERIMENT 1

In this experiment, it compares the performance among strategy learner (Q learner), manual strategy, and benchmark.

5.1 Input Parameters

The in-sample period is 2008-01-01 to 2009-12-31. Stock is JPM. Impact is 0.005. Commission is $9.95. Starting value is $100,000. Positions -1000, 0, +1000 shares.

All other hyperparameters belong to Q learner and manual strategy are fixed. The in-sample data is fed into each of the strategies. For Q learner, it will go through the learning cycle first from `add_evidence()`, and spit out a series of trades as a dataframe through `testPolicy()`. Manual strategy already has the hyperparameters tuned in previous section so it will simply generate the trades in `testPolicy()`.

5.2 Hypothesis

The advantage of reinforced learning is it can correlate its action based on rewards without explicit labels on actions. Since there are in total of 999 possible states, it can easily figure out very good actions to maximize portfolio returns. It is likely Q learner will outperform manual and benchmark strategies by a huge amount.

5.3 Discussion

Figure 3 below is generated from 3 strategies mentioned above. As expected, Q learner was able to find patterns between stock price movement and indicator values.

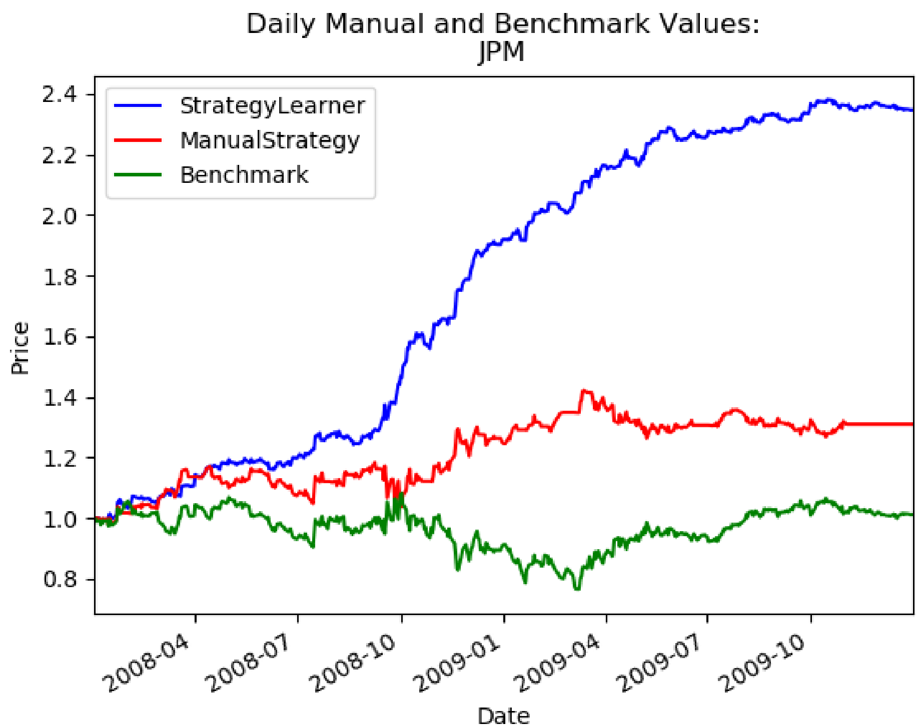


Figure 3 — Performance comparison among Q learner, manual strategy, and benchmark, JPM, 2010-01-01 to 2011-12-31.

Looking at the chart in details, one can find that the highest return per unit time (steepest positive slope) with Q learner occurs when benchmark shows huge volatility in 2008-10. This makes good sense because Q learner learns based on given rewards. The actions will skew towards days with higher rewards. Q learner outperforming other strategies should be able to repeat with the same in-sample data because Q learner is trained by the same data. If the Q learner is trained from scratch every time with the same in-sample period, it would result in slightly different results due to random actions built into the Q learner, but it should still outperform other strategies.

6 EXPERIMENT 2

In this experiment, it compares the effect of impact on cumulative returns when using Q learner algorithm.

6.1 Input Parameters

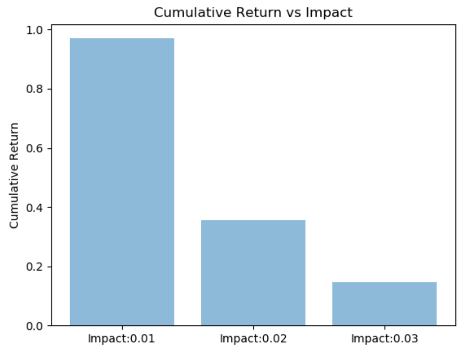
The in-sample period is 2008-01-01 to 2009-12-31. Stock is JPM. Commission is $0.00. Starting value is $100,000. Positions -1000, 0, +1000 shares. For impact, it takes values of 0.01, 0.02, and 0.03, so there will be three runs of training Q learner with three different impacts. All other hyperparameters in Q learner are the same as experiment 1.

6.2 Hypothesis

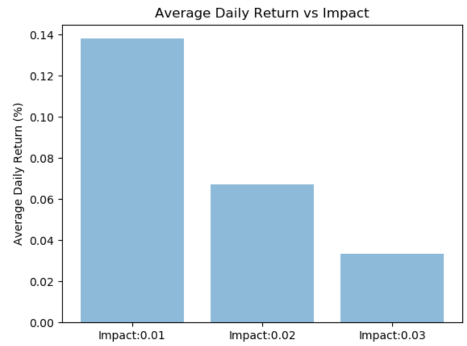
I chose to retrain Q learner each time when reward is changed because with the same number of trades (no retraining), cumulative return will simply decrease linearly with increasing impact. With retraining Q learner, rewards with higher impact is realized so few trades will be executed at days with higher gains. The cumulative returns should still decrease with increasing impact, but rate of decrease in return should slow down at higher impact.

6.3 Discussion

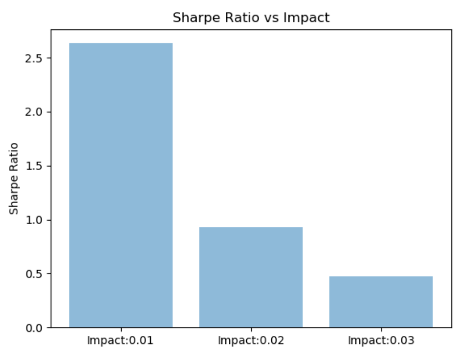
Figure 4 below shows the effect of impact with different metrics. Figure 4a is cumulative return. As expected, the return is diminishing as impact increases, and the relationship is looks more like exponential decay then linear. Figure 4b is average daily return, which takes similar shape as cumulative return.



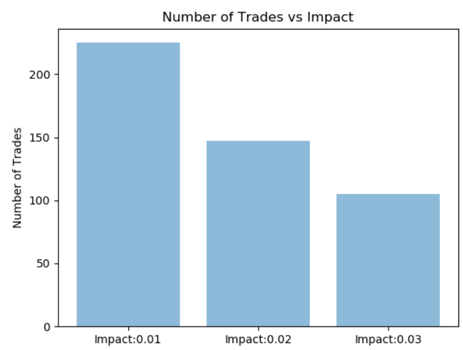
a)



b)



c)



d)

Figure 4— Effect of impact on stock portfolio using Q Learner, Stock JPM, 2010-01-01 to 2011-12-31.

Figure 4c is Sharpe Ratio. The shape is again similar to that of cumulative return. This indicates the volatility doesn't change much among three different impact values. Figure 4d is number of trades within the in-sample period. As hypothesized, as impact increases, Q learner acknowledges the reward is reduced every time a trade occurs. Number of trades is decreased to minimize loss from impact and focuses on actions with higher rewards.

7 REFERENCES

1. Bollinger Bands % B, Barchart, "URL: https://www.barchart.com/education/technical-indicators/bollinger_bands_percent", Accessed: Oct. 9, 2021
2. Relative Strength Index Modified, Barchart, "URL: https://www.barchart.com/education/technical-indicators/modified_rsi", Accessed: Oct. 9, 2021
3. Relative Strength Index (RSI), Fidelity, "URL: https://www.fidelity.com/learning-center/trading-investing/technical-analysis/technical-indicator-guide/RSI", Accessed: Oct. 9, 2021
4. MACD Oscillator Exponential, Barchart, "URL: https://www.barchart.com/education/technical-indicators/macd_oscillator_exponential", Accessed: Oct. 9, 2021
5. Pandas Documentation, https://pandas.pydata.org/docs/, accessed: 2021-11-12.
6. Numpy Documentation, https://numpy.org/doc/, accessed: 2021-11-12.