Project 8 : Strategy Evaluation

ML4T FALL 2024

Huida David Shi

hshi320@gatech.edu

8.1.1 Indicator & Strategies Overview

The three indicators used in analysis are

- Bollinger Bands Percent
- Stochastic Oscillator
- SMA20

SMA20 calculates the 20 day moving average. This smoothes price action and de-value outliers. Bollinger Bands Percent calculates price relative to bollinger bands. This shows how much price differs from SMA using standard deviations. Stochastic Oscillator depicts momentum by comparing closing price to price range over a period.

The purpose of these indicators is with the combination of some rules we can create a manual trading strategy based on indicator values. We will compare the performance with a Q-learning based strategy learner. There will be some rules and constraints used on both strategies to keep things simple and consistent. The insample dates is 01-01-08 to 12/31/09. Out of sample test is 01-01-10 to 12-31-11. In addition the only allowed positions are +(1000) long, -1000 (short), or 0 (netural).

8.2.1 Manual Strategy

The goal of the manual strategy is to use the current price to derive the indicators, then use a combination of rules on indicator values and result in actions.

Constraints & Rules

Buy Signal when all 3 conditions are met

- Price is below SMA20
- Bollinger Band Percent is lower than 15
- Scholastic Oscillator is lower than 15

Sell Signal when all 3 conditions are met

- Price is above SMA20
- Bollinger Band Percent is higher than 85
- Scholastic Oscillator is lower than 85

By combining indicators to determine actions, we have more improved contextual analysis compared to using a single indicator. Also using multiple indicators mitigates the chance of false signals. The buy signal constraints signifies that the stock is oversold, when all three indicators agree, there is more confidence. Sell signal constraints indicator values all agree the stock is overbought.

We will create a benchmark strategy to show how well or poor the manual strategy did in comparison. The benchmark strategy is long 1000 shares from the first day until the last of the analysis date. In addition the portfolio value for both strategies will be normalized for easier comparison.

8.2.2 *FIGURE 1*

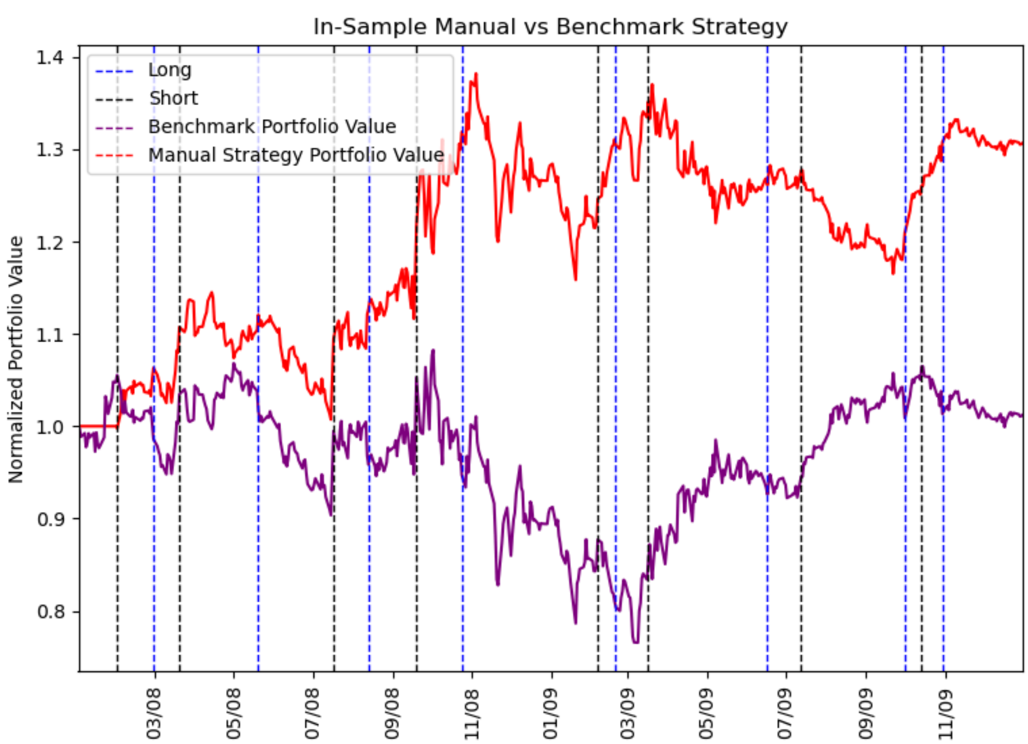

Figure shows 01/08 - 12/09 JPM Manual vs Benchmark portfolio value. Black and Blue vertical lines denote buy and sell positions of manual strategy. (insample_strategy_orders.png)

8.2.3 *FIGURE 2*

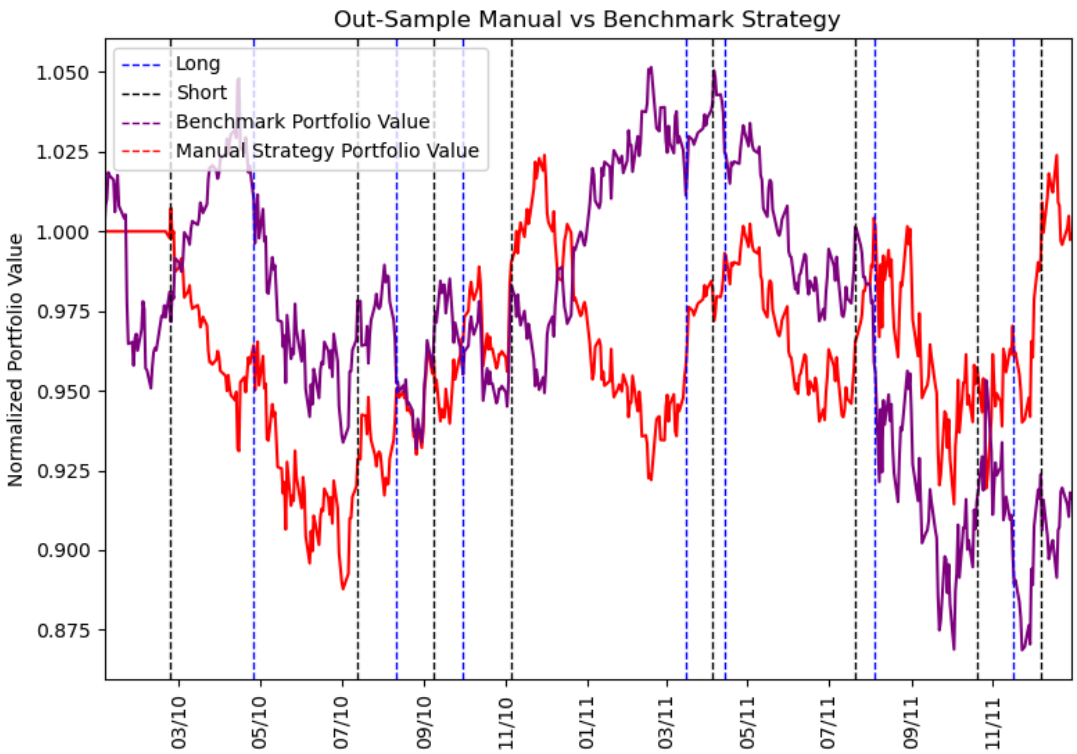

Figure shows 01/10 - 12/11 JPM Manual vs Benchmark portfolio value. Black and Blue vertical lines denote buy and sell positions of manual strategy. (out_sample_strategy_orders.png)

In sample graph shows the manual strategy has an amazing performance with more than 30% return. However the same performance does not translate to out of sample. The reason is since the manual strategy has static constraints and does not learn, the model performance will be independent between years. Another reason for the difference in performance is due to one year having better price action with less outliers for false signals.

8.2.4 *FIGURE 3*

In sample metrics:

Strategy	Cumulative Return	Standard Deviation Daily Returns	Average Daily Returns
Manual Strategy Portfolio	0.306811	0.012912	0.000614
Benchmark Portfolio	0.012325	0.017041	0.000169

(In Sample Strategy_Metric_Table.csv)

8.2.5 ***FIGURE 4***

Out sample metrics:

Strategy	Cumulative Return	Standard Deviation Daily Returns	Average Daily Returns
Manual Strategy Portfolio	-0.001062	0.008062	3.00E-05
Benchmark Portfolio	-0.083579	0.0085	-0.000137

(Out Sample Strategy_Metric_Table.csv)

Although the out of sample manual strategy did not perform as well as in sample, it did outperform the benchmark.

8.3.1 Strategy Learner

The strategy learner applies Q learning based reinforcement learning to identify and execute trades based on market conditions. The original q learner was a rewards based bot trying to reach a goal on a map. To turn it into a trading problem we represent buy,sell and hold as 3 different actions. 1= Buy, 2= Sell, 0= Hold. For each action the bot is rewarded. In this instance reward is denoted as daily portfolio return. The q learner will optimize actions based on rewards and account for discounted future rewards as well. We then represent indicator values as different states in the q table. In the beginning we have random exploration to fill the q table with states, with each iteration we decay the randomness. Eventually the q table will converge and in every state the q learner can query past states to determine the best action.

Hyperparameter tuning can change how the bot acts or learns. Tuning the parameters is a key factor in optimizing the performance of the q learner.

Parameter definitions:

- num_states = (10 x 10 x 10) =1000 states (bin ^ actions)
- num_actions=3 (buy, sell, hold)
- alpha=0.19 (learning rate, higher the number, more weight on each iteration of learn
- gamma=0.90 (discount rate on future rewards, higher the number, high priority on future rewards)
- rar=0.91, (random action rate for exploration, higher the number, higher probability of randomness)
- radr=0.99, (decay rate of random action exploration, higher the number, higher decay)

8.4.1 Experiment 1

The purpose of this experiment is to compare Qlearn, Manual & Benchmark strategy. My hypothesis is that Qlearn strategy learners should have better performance than the other two strategies. I believe q learner have more flexibility to dynamically adapt to market conditions. The fixed constraints in the manual strategy are too rigid and may perform badly in different market conditions. The manual strategy relies on indicator thresholds which may be not reliable due to how subjective it is. For example, I have the constraints BB% < 15 as oversold indicating a buy signal. However someone else might think BB% <25 low enough for their risk tolerance. Comparing the counterpart q learner, tuning hyperparameters have a lot more benefits than changing threshold numbers. The parameters provide ways to balance risk and reward while quite arbitrary to calculate in manual strategy.

8.4.2 *FIGURE 1*

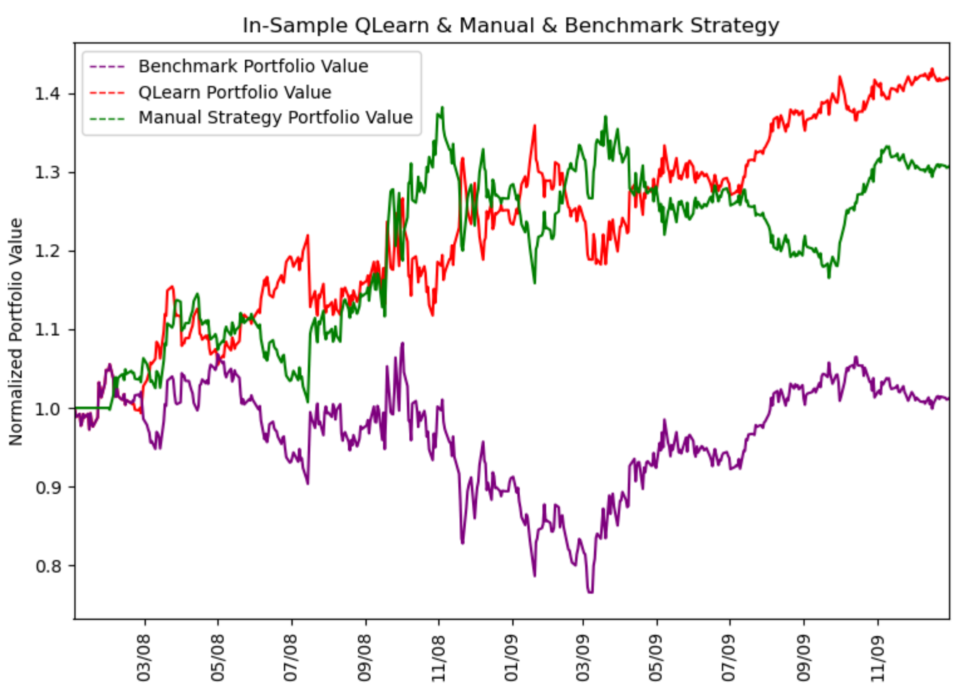

Figure shows 01/08 - 12/09 JPM QLearn vs Manual vs Benchmark Strategy portfolio value (in_sample_strategy.png)

As shown on the graph above, the Qlearn portfolio value did indeed perform better in comparison. Q learning's biggest advantage is the ability to learn from past experience in comparison to the manual strategy which requires human knowledge on thresholds. As the market condition changes the thresholds have to be changed manually, the q learner will learn dynamically. You can argue you need knowledge to tune the parameter, however there are other maximizer functions that can help test many parameters and pick the best one (gridsearchcv).

8.4.3 *FIGURE 2*

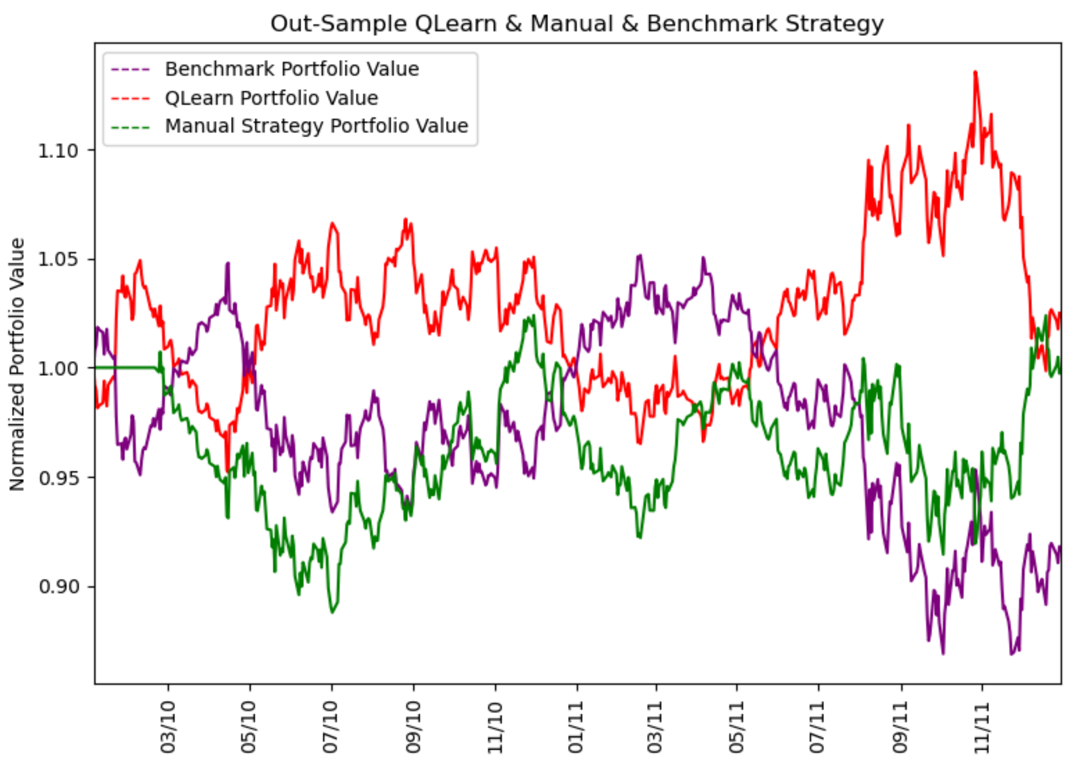

Figure shows 01/10 - 12/11 JPM QLearn vs Manual vs Benchmark Strategy portfolio value (out_sample_strategy.png)

Graph is showing the q learner has beat benchmark but showing poor out of sample performance. This indicates that there is a possibility we are overfitting on in sample data. Some steps to mitigate overfitting are decreasing the decay of randomness or starting with a higher probability of randomness. We can also decrease the learning rate to slow down the convergence.

8.5.1 Experiment 2

The purpose of this experiment is to compare q learner performance when the impact is changed. Impact in trading is defined as market impact, typically ranging from 0.01% - 2% depending on liquidity. The higher the percentage the more your buy order will create a bigger price shift on market price. Slippage occurs when the price deviates from your action price because the order is so large it moves the price. High impact will change the q learners behavior drastically. The q learner has to optimize towards strategies with lower risk and lower returns due how much each trade affects the stock price. In addition the q learner will try to look for less trades.

8.5.2 ***FIGURE 1***

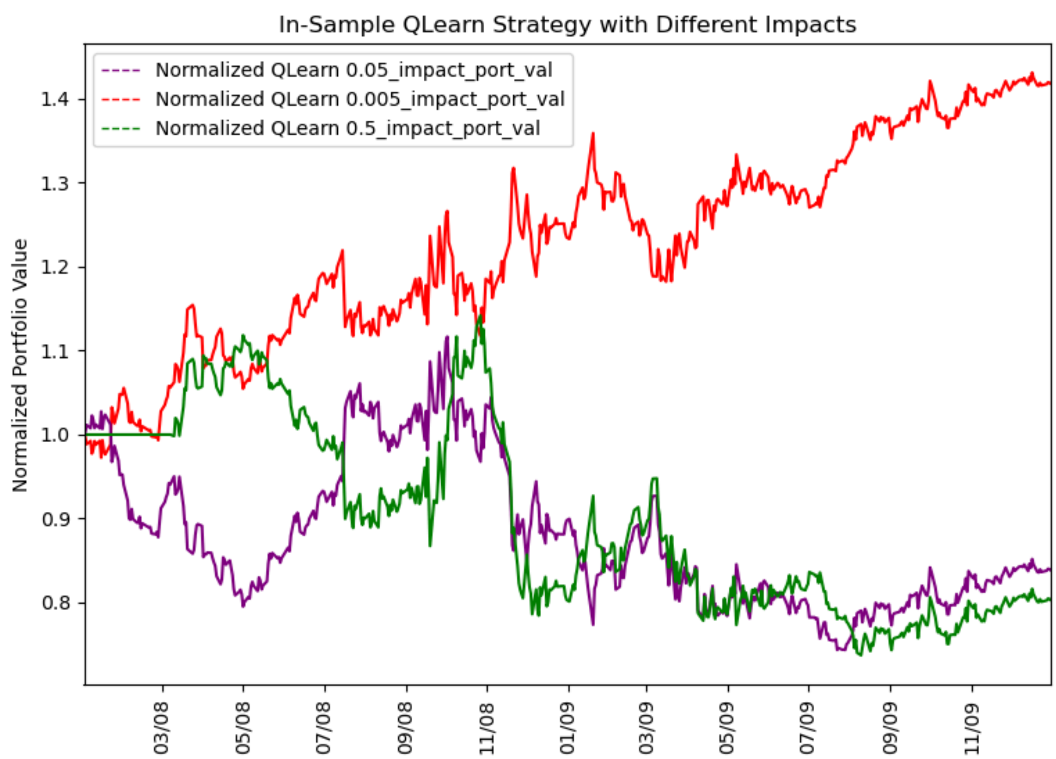

Figure shows 01/08 - 12/09 JPM portfolio value when trading at different impact levels (in_sample_strategy_impact.png)

8.5.3 ***FIGURE 2***

Strategy	Cumulative Return	Standard Dev	Average Daily Returns
0.5_impact_port_val	-0.196969	0.017338	-0.000285
0.05_impact_port_val	-0.161519	0.017595	-0.000195
0.005_impact_port_val	0.417809	0.013378	0.000782

Figure shows JPM portfolio value metrics trading at different impact levels

We ran the q learner on 3 different impact levels for the original in the sample period shown above. When the impact is high our original q learner performance is poor. The original q learner was hypertuned for low portfolio's value relative to market cap and liquidity so it may take on more trades and risk. Some improvements I would implement is to increase lookback dates or change indicators. In a case where Hyperparameter tuning may not be enough sometimes and it's best to start fresh from the drawing board.