Machine Learning for Trading: Strategy Evaluation

Luke Pratt
lpratt30@gatech.edu

Abstract — A manual stock trading strategy using the standard deviations of technical indicators was developed and compared to buy and hold. Using the same indicators, a reinforcement learning (Q-Learner) strategy was created. The behavior of the Q-Learner was then characterized across hyperparameter tuning. Finally, the Q-Learner was demonstrated to result in higher profits for both in-sample and out-of-sample data for the provided dataset.

1 INTRODUCTION

Actively trading stocks comes from a belief that weak efficient market hypothesis is false. If weak EMH is false, it would mean that it is possible for traders to profit from trading based on technical indicators. This paper does not seek to explore the validity of weak EMH. However, it does make the implicit assumption, valid or not, that weak EMH is false, and then seeks to develop a trading strategy that exploits technical indicators for profit.

Two strategies are developed using five technical indicators and then compared. First, a manual strategy is developed that decides the trading stances of "Long", "Short", or "Cash" based on a combination of the indicators and their standard deviations. Then, the same technical indicators are used to train a Q-Learner to decide the same trading stances. Finally, their performances are compared.

1.1 Indicator Overview

Bollinger Bands are the first of five indicators chosen (Investopedia. *Bollinger Band*). Bollinger Bands are rolling price value standard deviations calculated for +2 Sigma and -2 Sigma. For this experiment, 20 days was used for the rolling calculation.

Bollinger Bands were modified into the Bollinger percentage, which is the stock price's relative position between the top and bottom bands. A 0% value represents the lower band, 100% represents the top band, and values outside of that range represent exceeding the bands by X%. This value is more programmatically friendly than the bands for the manual strategy and the single value is more readily handled by the Q-Learner. It is calculated by:

$$BB_{Percentage} = 100 \times (price - lowBand) / (highBand - lowBand)$$

Equation 1 *- Bollinger Percentage*

The next indicator was the Price Percentage Oscillator (Investopedia. *Percentage Price Oscillator (PPO))*. The PPO is the relative value of a shorter term EWM (exponentially weighted moving average) to a longer term EWM. Here, a period of 12 days is used for short-term, and 26 for long term.

$$PPO = 100 \times (rollingEMA(12) - rollingEMA(26)) / rollingEMA(26)$$

Equation 2 - *PPO*

The third indicator was Simple Moving Average. SMA is similar in use to Bollinger Bands in that it is used to indentify discrepances of price to the specified time frame. A rolling period of 14 days is used. Additionally, SMA was modified to be relative to the price of the stock so that normalized values were returned.

$$Relative\ SMA = price/RolingMean(14)$$

Equation 3 *– Relative SMA*

The fourth indicator was Stocastic Oscillaton. (Investopedia. *Stochastic Oscillator*). Stochastic Oscillation identifies percentage shifts in a stock's price relative to the extremes over a time period. It is a highly volativle indicator as implied by its name. Here, a period of 14 days is used.

$$k = 100 \times (price - rollingMin(14) / (rollingMax(14) - rollingMin(14))$$

Equation 4 *–Stochastic Oscillator*

The final indicator was the Rate of Change (Investopedia. *Price Rate of Change Indicator (ROC)).* ROC is similar to k but less focused on extremes and not quite as volatile. ROC is defined as the relative price difference of a stock to its value over a specified period of time; here 12 days was used.

$$ROC = 100 \times (price - priceBefore(12)) / (priceBefore(12))$$

Equation 5 *– ROC*

For all indicators used, the time-period values were kept constant for both the Manual Trader and the Q-Learner. There is opportunitity for finding higher performing trading strategies by also optimizing for the time periods used by the indicators. In these experiments, the focus is on the way in which the indicators are combined and discretized.

1.2 Experiment setup

The same data and assumptions are used for both the manual and the Q-Learner strategies. The stock JPM (J.P Morgan) is traded. For both strategies, buy-and-hold benchmarks performance.

In-Sample data is taken as the period between January 1st, 2008, to December 31st, 2009. Out-of-Sample data is taken between January 1st, 2010, to December 31st, 2011. To simplify the experiments, the adjusted close value is used for all calculations, although in reality it is not possible to precisely trade adjusted close.

Commission is $9.95 and a market impact of 0.5% is assumed (unless stated as otherwise). For every trade, the commission and the cost of the market impact is reduced from the cash balance and impact always works against the trade.

The trader always starts with a value of $100,000. It is assumed the trader can use debt and hold a negative cash value at a 0% interest rate. Only discrete trading positions are allowed. The trader may be in the position of holding 1,000, 0, or -1,000 shares at any given point in time. Therefore, each strategy only points to long, cash, or short, and does not return a suggested purchase size. The trader can make trades of the size of 2,000, 1,000, -1,000, and 2,000, so long as it stays within the bounds of +1,000 and -1,000 shares held. Short positions are assumed to cost the same to enter as long positions.

1.3 Manual strategy

While researching technical indicators, one finding was that many of the indicators were stable variations of how the price has changed relative to the 2–3-week period, while other indicators caught extremes and had much higher levels of volatility. Thus, the first idea behind the manual strategy became that the stable trend may be combined into 1 indicator to determine higher level confidence shifts from the trend, while the more volatile indicators may be less reliable but occasionally point to higher profit trades.

All indicators had higher positive values when the price was higher relative to previous behavior. All the indicator values were first standardized such that they now represented sigma variations on the same scaling. To combine them to fit the hypothesis, the equations were then:

$$Volatile\ Confidence = today(K) + today(ROC)$$

$$Stable\ Confidence = today(BB\%) + today(PPO) + today(SMA)$$

Equation 6 *– Manual Indicator Combinations*

The next idea behind the trading strategy was that a stock may be determined to be overbought or oversold based on technical indicators. Using the combined indicators, "overbought confidence" and "oversold confidence" were calculated. During trials of in-sample data, it was also found that the strategy performance could be improved by reducing the weighting of the volatile indicators and rebalancing towards the more stable ones.

$$Overbought\ Confidence = \frac{4}{3} Stable\ Confidence + \frac{1}{2} Volatile\ Confidence$$

$$Oversold\ Confidence = 0.0 - Overbought\ Confidence$$

Equation 7 *– Overbought and Oversold confidences*

Finally, the overbought and oversold confidences were used to decide to enter long, short, or cash positions. The principle was that an overbought stock should be shorted, an oversold stock should be bought, and that any time a position is taken, there is at some point a return to normalcy at which the trader should exit the trade and take profits. A return to normalcy is defined as the point the absolute value of overbought confidence is equal or less than the exit factor.

However, there is still the question of what the best value is to enter a trade and what is the best value to exit a trade. Iteration was used to find the best performing in-sample confidences and then exit points. To reduce overfitting and increase the program's run speed, a sparse iteration space of 4 linspace confidences between 1.5 and 6.0 was taken for 5 linspace exit factors of -1.0 to 6.0. The combination of values that resulted in the highest final portfolio value was taken to be the best. These values were then manually set as an attribute inside of the manual strategy such that the iteration does not need to occur on each execution.

Shown below in Figure 1 and 2 are the results on the In-Sample and Out-Of-Sample trading. Blue represents long, black indicates short, and green dashed represents switching to cash.

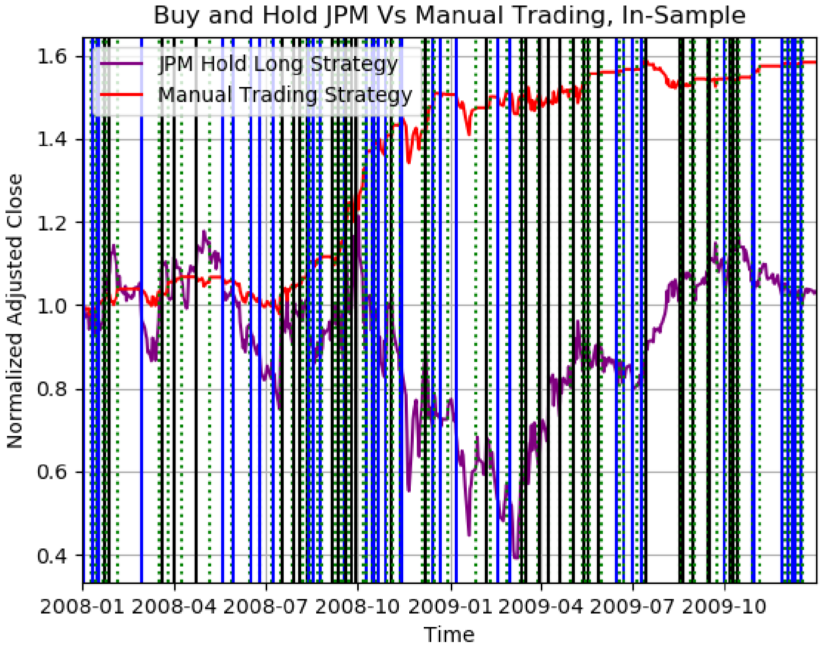

Figure 1 *- In-Sample Manual Trader*

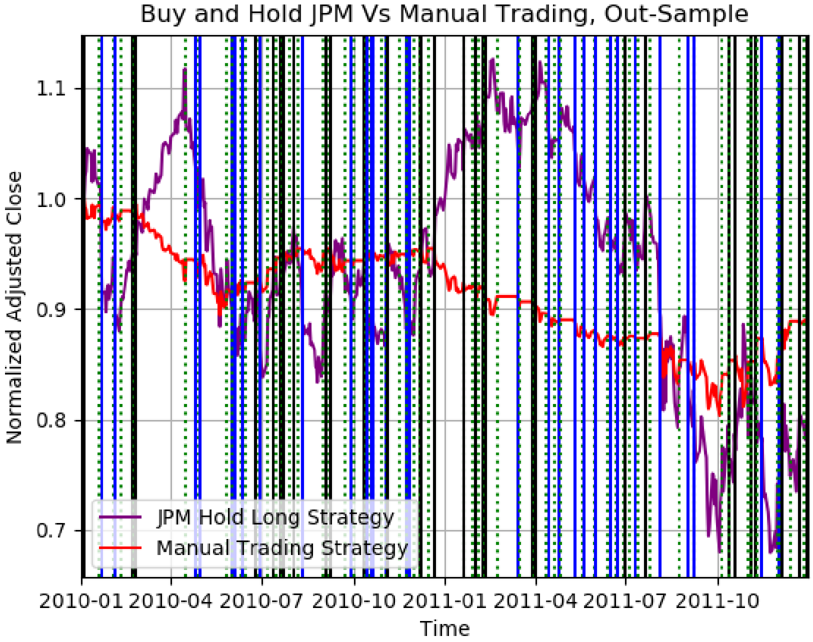

Figure 2 *– Out-Of-Sample Manual Trader*

The results of the in-sample and out-out-sample trading are summarized in the below table 1:

Set	Strategy	AVG Daily Return	STD Daily Return	Cumulative Return
In-Sample	Manual	.095765%	.955648%	58.23110%
In-Sample	Buy-and-Hold	.139566%	5.21445%	2.30000%
Out-Of-Sample	Manual	-.026367%	.724809%	-11.01989%
Out-Of-Sample	Buy-and-Hold	-.019846%	2.26141%	-8.33999%

Table 1 *- Manual Strategy Performance*

Both strategies managed to outperform the benchmark buy-and-hold strategy. However, the in-sample performance was far superior relative to the benchmark. This suggests that the manual trader did have some overfitting in its implementation in spite of a sparse iteration space for optimization. Perhaps the rebalancing of the volatile confidences contributed to the overfitting, or perhaps the overfitting could be reduced by increasing the required confidence factor to enter a trade.

1.4 Q-Learner

For the Q-Learner, it was able to take 3 possible actions: cash, long, or short. A wrapper script for the Q-Learner handled making trades as allowed by the rules of the experiment when the Q-Learner wanted to enter or maintain a position. Daily rewards were passed back to the Q-Learner, being the difference in value of the portfolio from the current day to the previous day. The Q-Learner iterated through the in-sample data a minimum number of times, and then past that point would consider itself converged in the prior final portfolio value equaled to that of the current iteration. To reduce the search space of the Q-Learner, holdings were not kept as a part of its search space; it was agnostic about taking a position based on its current position.

The the Q-Learner used all 5 indicators, with the K and ROC indicators being combined. Therefore it uses the same information available to the manual trader. The tradeoff between using more indicators is that a richer more nuanced decision-making process can be created, but there is a sparser search space from which to learn to make actions.

The Q-Learner was given a space of possible states created by the indicators. The number of possible states for the Q-Learner was the number of possible combinations of indicators. For every indicator, the wrapper standardized and then discretized the values to reduce the number of possible states to avoid a learning space that was too sparse (overfitting). The value being standardized meant that each discrete value had significance in terms of the indicator's standard deviation from normal.

To discretize the standardized values, NumPy array operations were used. Negative values were taken out by adding the absolute value of the minimum to the array. The delta between discrete points was found by dividing the range of values by the number of desired discrete states minus one. Then, each point was divided by the delta. Finally, the values were rounded and casted as integers.

For simplicity, the number of discrete states for each indicator was the same. However, there is possible value in doing hyperparameter turning on which individual indicators receive how many discrete values, as well as how many indicators are combined in total (not to mention the periods used for calculating each indicator). Additionally, the method of discretization offers some possible exploration. For example, to create discrete values that focus on extremes of standardized values, discrete values intermediate the extremes can be folded into the extremes.

4 discrete values for each indicator was settled upon after a few iterative trials for a combination of in-sample performance and reduced overfitting. With 4 indicators having 4 possible discrete values, the total search space of the Q-Learner was thus 256, or approximately half that of the number of trading days in the in-sample period. It is thus not probable that every space will be explored at least once, but it is probable that the majority of them will be explored.

The selection of the Q-Learners initial random action selection rate and its decay rate was a tradeoff for faster execution speed, Q-Learner predictability, and final performance. As the initial random rate of actions was made larger with decay rate constant, the Q-Learner was able to explore a wider range of possible trading strategies. However, the Q-Learner would need longer to converge, and also may sanction off effective trading strategies if the rate of random action decay was low. Because, the Q-Learner may have negative experiences while trading randomly that aren't indicative of how the same trades in the same state may perform if they are being reached by a more well optimized Q-Learner. Therefore, the most effective Q-Learner for this trading experiment is the one that has a steep initial randomness rate and a low rate of decay.

However, to meet the performance requirements of training and performing in no more than 25 seconds, a relatively steep rate of decay and high initial random action rate were landed upon for faster convergence. The final values selected were:

Alpha	Gamma	Random Action Rate	RAR Decay
0.2	0.9	0.8	0.9995

Table 2 *– Q-Learner Parameter Values*

1.5 Experiment one

The performance of the Q-Learner compared to the benchmark and manual trading strategies, in-sample and out-of-sample, is shown below in figures 3 and 4:

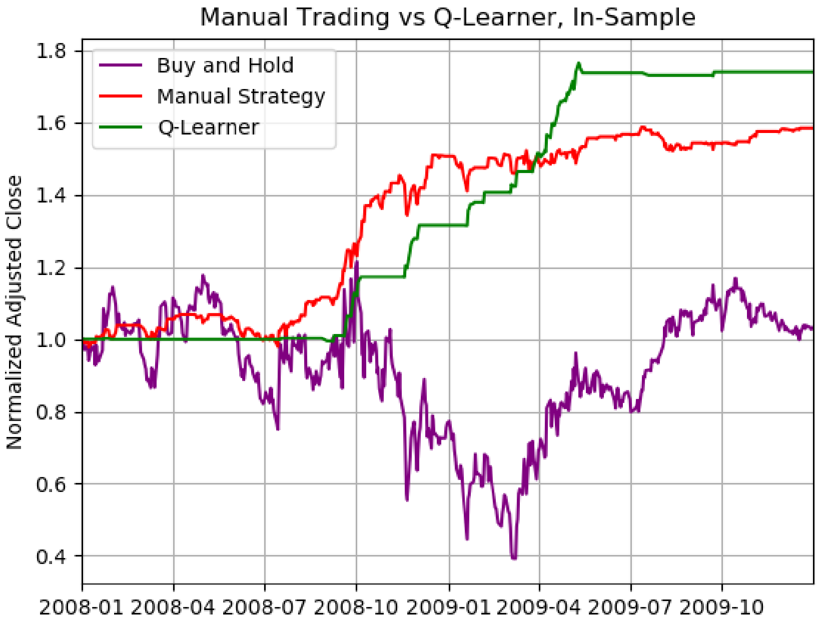

Figure 3 *– In Sample Performances*

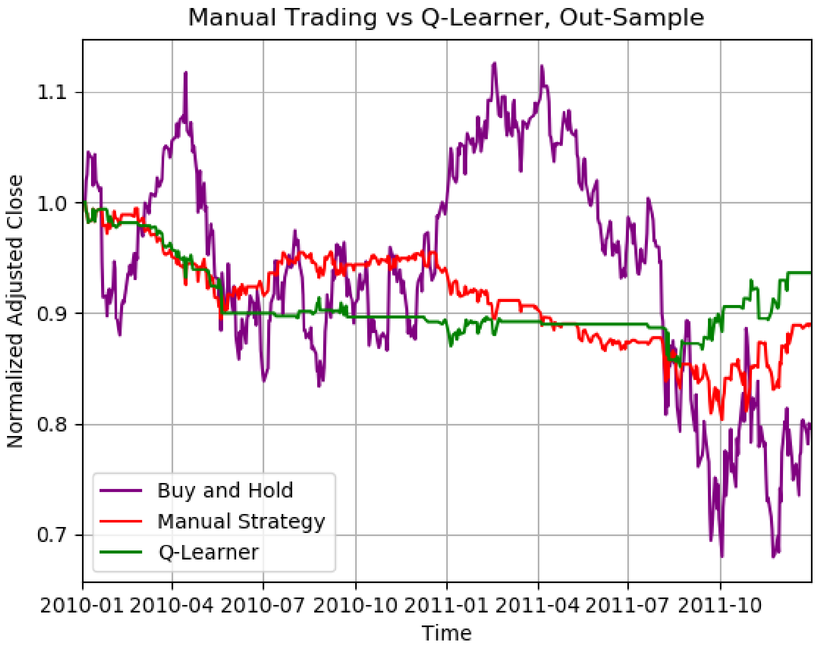

Figure 4 *– Out Of Sample Performances*

In both in-sample and out-of-sample, the Q-Learner substantially outperformed the manual trading strategy. Although the Q-Learner may be expected to overfit because it makes very specific decisions based on combinations of a state's value, while the manual learner used a simple cutoff, this was not observed. The Q-Learner gets a rich space of possible decisions which may be capturing some deeper information not available to the manual trader. Further, the Q-Learner is making decision on short-term rewards, which means it will take reactionary stances to any particular trading days rather than complex plans.

As discussed, there is still much room yet for more of the Q-Learner to be explored and improved. However, it is important to note that the performance of this Q-Learner is highly random because it has a high initial random action rate and a relatively fast random action decay rate. Some trials will result in better performance than shown above, some worse. This is entirely dependent on the initial random NumPy array, and the random decisions chosen by the Q-Learner. More consistent results of higher performance are achieved when allowed longer execution time constraints and lower random action decay.

1.6 Experiment two

In the next experiment, the behavior of the Q-Learner's performance was viewed as the impact was varied. A linspace of 12 impacts between 0, the impact used in the initial training of the Q-Learner (.5%), and a value of twice that used in training (.1%) the Q-Learner was used. The Q-Learner is trained on each value and then executes. The expectation is that the final portfolio value will decrease with impact, while the profit made per trade will also decrease.

As shown in the blow figure 5, the value of the final profit did indeed decrease with the increasing impact. There is variation point-point due to the high Q-Learner randomness as discussed earlier, but the trend is overall a linear decrease.

Shown below in figure 6 is the average profits made per trade. As shown, the profits per trade were volatile but on the whole did not *substantially* decrease as impact decreased. This indicates that the Q-Learner is compensating for the increased cost to make a trade by making fewer trades of higher confidence, else it would also be faced with a similar linearly decreasing value of profits per trade.

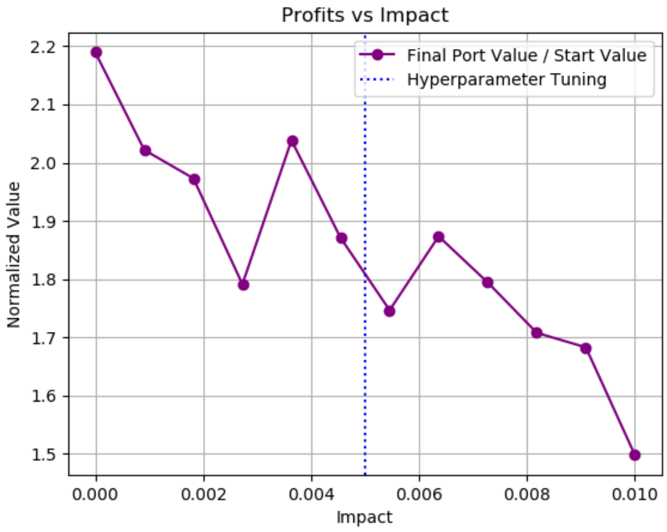

Figure 5 *– Q-Learner: Final Profits vs Impact*

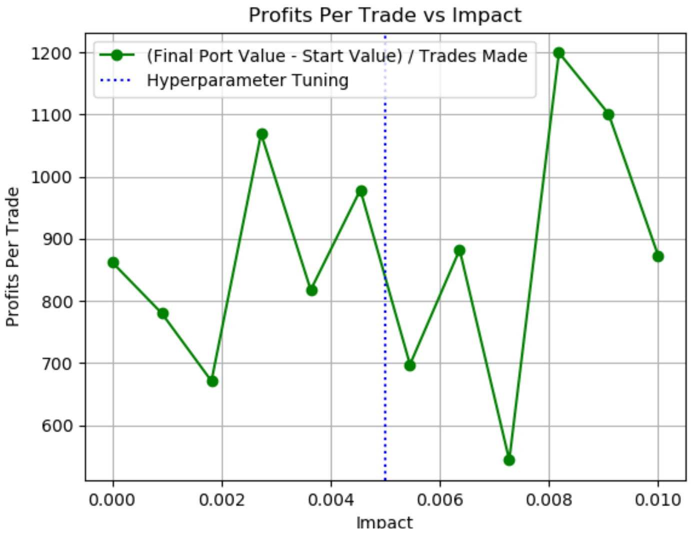

Figure 6 *– Q-Learner: Profits per Trade vs Impact*

REFERENCES

1. Hayes, A. *Bollinger Band*. Investopedia. https://www.investopedia.com/terms/b/bollingerbands.asp
2. Mitchell, C. *Price Percentage Oscillator (PPO)*. Investopedia. https://www.investopedia.com/terms/p/ppo.asp
3. Mitchell, C. *Price Rate of Change Indicator (ROC*). Investopedia. https://www.investopedia.com/terms/p/pricerateofchange.asp
4. Hayes, A. *Stochastic Oscillator*. Investopedia. https://www.investopedia.com/terms/s/stochasticoscillator.asp