CS 7646 Project 8: Strategy Evaluation

Nick DiNapoli
ndinapoli6@gatech.edu

***Abstract*—**

In this work, I implement two trading strategies using four technical indicators and compare their performance. One strategy is completely manual and the other develops the trading rules using machine learning (ML) – specifically, using the reinforcement learning algorithm, Q learning. For this analysis I use JPMorgan Chase & Co. (JPM) historical daily adjusted close data. I train the algorithm using data from the time period January 1, 2008 to December 31, 2009 (in-sample) and on top of evaluating the strategies over this window, I evaluate both over the period January 1, 2010 to December 31, 2011 (out-of sample).

1 INDICATOR OVERVIEW

For this analysis, I use four technical indicators to build the manual and reinforcement learning strategies: the 50-day simple moving average (SMA), momentum, %B indicator, and the true strength index (TSI).

For the SMA, a few implementations were considered for using it as an indicator, the raw value, Price/SMA ratio, death cross, golden cross, and where the SMA crosses the stock price (1). Ultimately, I chose the last option as it served as a great indication of trend reversals on the exemplar data. Therefore, when the stock price of the previous day was below the SMA but the current day's price is above that of the SMA, a sell is indicated representing a trend reversal. Likewise, when the stock price passes through the SMA from the other side (above to below) a buy and indicated. For this indicator, a few things could be tuned for optimization. For example, one could adjust the spread of the number of days between the "'cross" but that is essentially just a shift in the SMA and could be achieved in a similar fashion by adjusting the N-day SMA. Intuitively, I chose to indicate a buy/sell on the actual day of the cross.

For the momentum indicator, some considerations using it as an indicator were examining where the value passes up or down through the 0-line, indicating bullish and bearish times(2). For the training in-sample period, I noticed this happens way too often compared to the real trends in the data so I chose a thresholding approach. I only indicated a buy/sell when the momentum hit more extreme values, specifically, $\pm.2$. Using these values ensured strong on-going trends and/or trend reversals. In this analysis, I used momentum as an indication of an upcoming trend reversal, i.e. when the momentum gets too high, indicate a sell. Using momentum in this way allowed me to tune both the upper and lower limits to achieve large cumulative returns during the in-sample period.

John Bollinger claimed that an uptrend begins when the %B passes 0.8 (can be interpreted as overbought) and a downtrend is identified when %B falls below 0.2 (can be interpreted as oversold) (3). Other strategies for using the %B as an indicator include combining it with other indicators. I chose a similar thresholding method to my implementation of the momentum indicator and John Bollinger's hypothesis. However, I once again found that those initial values for thresholds were not optimal for this in-sample period and were too loose for this data. The thresholds needed to be tuned to maximize return and after a parameter search I decided on more stringent values of 1.1 and -.15. Counterintuitively, like the last two indicators, using %B as a trend reversal indicator proved to be more valuable so when too high of a value is achieved, a sell is indicated and vice versa.

Lastly, I chose to analyze the TSI. According to (4), positive territory means the bulls are more in control of the asset and (5) claims that a string of positive price changes results in high positive readings and strong upwards momentum. It is also common to note that trend/price reversals happen around values of ±20. A few implementations of the TSI were considered such as, the 0-line crossing, its divergence with stock price, combining it with another indicator, and other signal line crossovers. I decided on using the latter and like the other indicators (except SMA) I tuned upper and lower threshold values. Finally, I landed right back where I started to values of ±20. Again, I used the TSI as a trend reversal indicator and sold the asset when the TSI was too high.

2 MANUAL STRATEGY

Creating and designing the technical indicators as in the previous section is just one part to using them to develop an entire strategy. There are several approaches to designing a manual strategy and ones that I considered were: a ranking system – iteratively check the indicator values for each day in a specific order and as soon as the first one indicates a buy/sell take that action, a buy/sell biased system – i.e. if the system is buy-biased, if any indicator indicates a buy, execute a buy, and a voting system – sum up/take into consideration the buy/sell/hold indications from all 4 indicators weighted evenly. Ultimately, I decided to implement a unique version of the voting system. I developed this strategy under the constraints that I could not have more than 1000 shares long or 1000 shares short but that I could trade 2000 shares at a time so long as these holding conditions are met. After developing this strategy, I compare the value of a portfolio based on the manual strategy and a benchmark strategy where one buys 1000 shares on day 1 and holds them for the entire time period. For this analysis and each strategy, I assume that I start with $100,000, trades have a commission of $9.95 and an impact of .5%. My unique strategy is outlined in the following pseudo code:

1. Calculate all 4 indicator values over the time period
2. Create trades DataFrame to store trading amounts
3. Iterate through the time series data
    (a) Initialize total votes to 0 (hold)
    (b) Query indicator values and check if they meet the buy/sell criteria laid out in section 1
    (c) If indicator value triggers a buy, add 1 to the total votes
    (d) If indicator value triggers a sell, subtract 1 from the total votes
    (e) If the total number of votes is 1, a "‘weak buy" is triggered and buy 1000 shares if the holdings is less than or equal to 0. Update trades DataFrame and current holdings
    (f) If the total number of votes is greater than 1, a "‘strong buy" is triggered and buy 2000 shares if the holdings is -1000 and buy 1000 shares if the holdings is 0. Update trades DataFrame and current holdings
    (g) If the total number of votes is -1, a "‘weak sell" is triggered and sell 1000 shares if the holdings is greater than or equal to 0. Update trades DataFrame and current holdings

(h) If the total number of votes is less than -1, a "'strong sell" is triggered and sell 2000 shares if the holdings is 1000 and sell 1000 shares if the holdings is 0. Update trades DataFrame and current holdings

4. Return a dataframe of each day in the time period with the number of shares traded on that day (positive for buy, negative for sell, zero for hold)

I decided on the voting system to not rely too heavily on any one indicator. Using this algorithm, a hold value occurs when either no indicator gets triggered or when exactly two are buys and two are sells. In that instance, with that level of disagreement, I don't trust a buy or sell. I also decided to implement the strong/weak buy/sell system because if always maximizing the trade amount (± 2000), consecutive days could never contain a trade which puts more emphasis on the very first day an indicator is triggered. Also, I employ the voting system because if two completely separate indicators are both indicating buys, I can really trust that as a true great day to buy and not a false positive.

To verify that my manual strategy is in fact an effective one, I compare its performance to the benchmark strategy over the in-sample period for which it was optimized and also apply both to an out-of-sample time period to further evaluate its ability to generalized well to unseen data. Figure 1 shows the portfolio values for the manual strategy versus the benchmark strategy over the in-sample period for JPM. Additionally, long and short entry entry points are included. As it can be seen qualitatively, the manual strategy I developed outperforms the benchmark by a significant margin over the in-sample period as expected. Table 1 provides the quantitative results for this experiment for further comparison. Figure 2 provides the exact same experiment but over the out-of-sample time period for which the manual strategy was not optimized. Once again, long and short entry points are indicated. Table 2 summarizes the results of this in a quantitative manner.

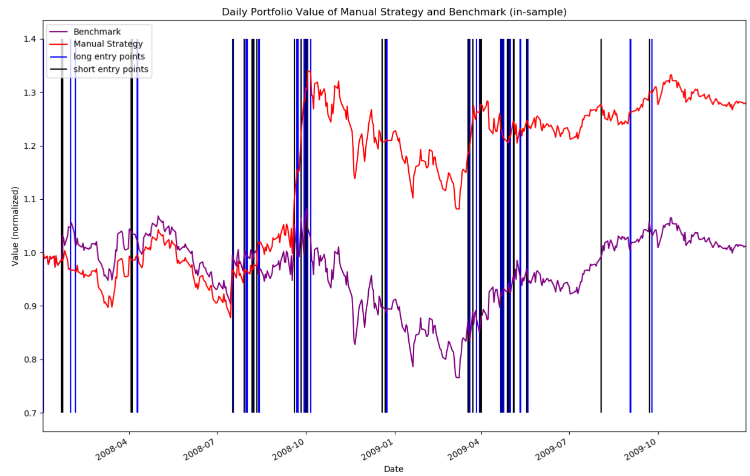

Figure 1—Portfolio value of my manual strategy versus the benchmark strategy for JPM over the in-sample period. The manual strategy significantly outperforms the benchmark as expected.

Performance Metric	Manual Strategy	Benchmark
Cumulative return	0.280218	0.012324
Standard deviation of daily returns	0.052093	0.017024
Mean of daily returns	0.001393	0.000168

Table 1—Performance of Manual strategy compared to the benchmark strategy for the in-sample period.

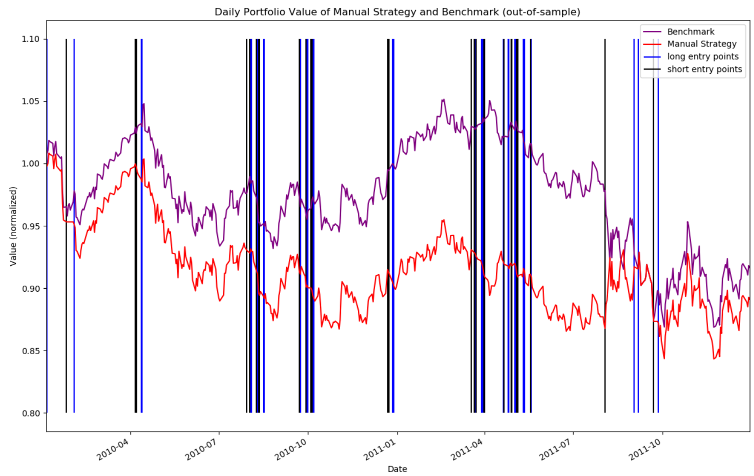

Figure 2—Portfolio value of my manual strategy versus the benchmark strategy for JPM over the out-of-sample period. Now, the manual strategy does not outperform the benchmark.

Performance Metric	Manual Strategy	Benchmark
Cumulative return	-0.109070	-0.083579
Standard deviation of daily returns	0.022592	0.008492
Mean of daily returns	-0.000198	-0.000137

Table 2—Performance of Manual strategy compared to the benchmark strategy for the out-of-sample period.

It is clear from Figure 1 and Table 1 that the manual strategy significantly outperforms the benchmark strategy as expected. But interestingly, looking at Figure 2 and Table 2 shows that the manual strategy cannot generalize well to the out-of-sample data compared to the simple benchmark which can be an indication that it is overfit to the in-sample data.

3 STRATEGY LEARNER

I now turn to creating, implementing, evaluating, and analyzing a machine learning algorithm to the same data presented in the previous section to see whether a learning algorithm can inf fact beat a human. As mentioned, I use the Q learning algorithm, a reinforcement learning model. I trained the algorithm using two passes over the in-sample period and then apply the model to both the in and out-of-sample periods and compare the performance to the manual strategy and the benchmark strategy. In order to use Q learning, and for fair comparison, I calculate the values for the exact same four indicators as previously mentioned and discretize their values by binning them into equal sized bins of 10. Once the values for each day are discretized, I create a unique state for each day over the period by concatenating the bin digits. In this way, there are 10,000 possible states for this Q learner. This strategy learner has three possible actions: buy, sell, and hold (do nothing) just as the previous two strategies. Lastly, for the reinforcement algorithm to function, I use daily average return as the reward for taking a specific action. For example, if the model takes a specific action after being in a particular state, I update the trades DataFrame and compute the daily returns and assign the reward as the return of the next day. In this way, if a bad action was taken after being in a particular state, the reward is a large negative number and if a good action was taken the reward is a larger positive number. Similar to the manual strategy, the constraints on buying and selling such that holdings never exceed ± 1000 is still in place but now anytime a buy/sell is indicated I maximize the number of shares traded instead of holding back as in the manual strategy. The model then outputs a final trades DataFrame which I use to compare performance. To compare the performance of the strategy utilizing reinforcement learning, I conduct two unique experiments.

4 EXPERIMENT 1

In this experiment, I compare the daily portfolio values for the strategy learner compared to the manual strategy and benchmark for both the in and out-of-sample time periods. I hypothesize that the ML-based strategy should outperform my manual strategy by a good margin but that due to the difficult task of generalizing well to unseen data, it may struggle. I would still expect the ML-algorithm to outperform a human if designed and implemented properly.

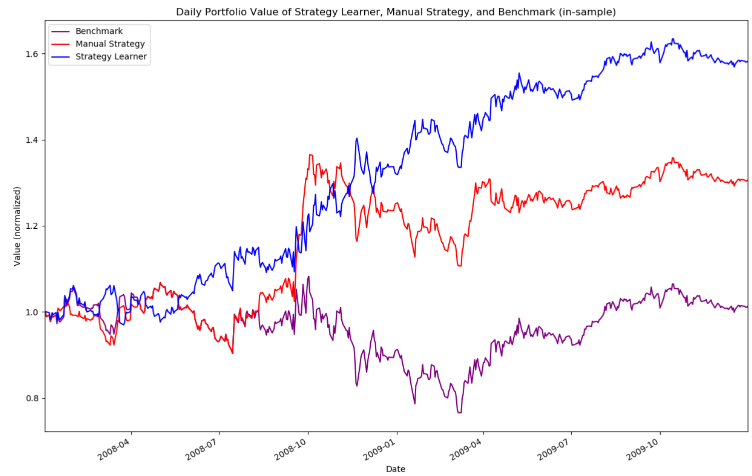

Figure 3—Portfolio value of my strategy learner, manual strategy, and the benchmark strategy for JPM over the in-sample period. The machine learning algorithm outperforms the other two by a good margin as expected.

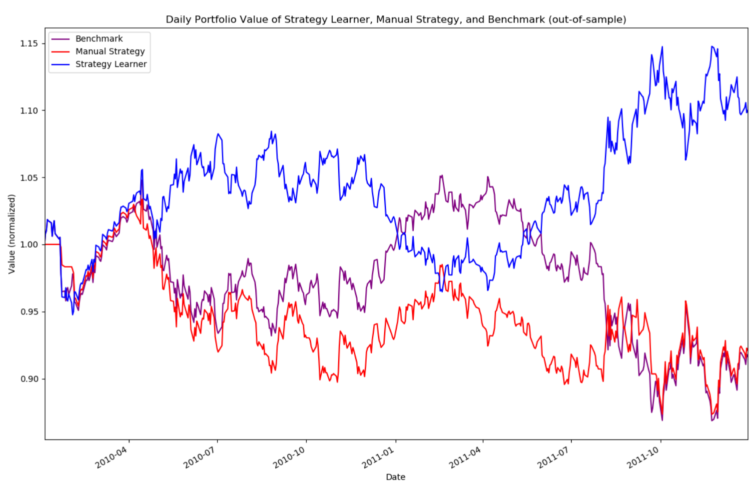

Figure 4—Portfolio value of my strategy learner, manual strategy, and the benchmark strategy for JPM over the out-of-sample period. The machine learning algorithm is resilient to the challenging unseen data and still outperforms the other strategies.

Figure 3 shows this experiment for the in-sample period and Figure 4 shows the results for the out-of-sample period for JPM. The reinforcement algorithm significantly out performs the other two strategies during the training period and interestingly is resilient to the unseen data and generalizes fairly well to the new period.

5 EXPERIMENT 2

In this experiment, I aim to discover how varying the impact trading parameter affects the performance of the Q learner during the in-sample period. Specifically, I change the impact from 0.0 to 1.0 and test 5 total values. It is worth noting that I set the commission to 0 and that the impact is the only parameter taking away from the portfolio value. Because raising the impact negatively affects the trader during every trade, I hypothesize an increase should lead to worse cumulative returns and that the algorithm will learnt to trade less as each trade negatively affects the trader compared to the previous situation where impact was 0. Table 3 summarizes the results for this experiment.

Impact	Cumulative return	Total number of trades
0.0	0.3204	5
0.25	-1.31945	9
0.50	-2.02025	6
0.75	-2.272925	5
1.0	-3.611	5

Table 3—Performance of Strategy learner when the impact parameter is varied during the in-sample period.

It is quite clear that generally speaking, the performance goes down when the impact parameter goes up which is precisely as hypothesized. the Q learner does a good job recognizing that trading doesn't have as positive as a benefit and that the trade decisions must be very strong ones for the portfolio to benefit at all. When the impact is maximized, the portfolio loses over 3X of its value. Additionally, the general trend of a fewer number of trades when impact is raised is not apparent to my surprise. However, this could mean that the algorithm still converged to a more strict policy but it does not quite appear that the learner is learning that when impact is high, the benefit to trading is simply not as good.

6 REFERENCES

1. https://www.investopedia.com/terms/s/sma.asp
2. https://www.investopedia.com/investing/momentum-and-relative-strength-index/
3. https://school.stockcharts.com/doku.php?id=technical_indicators:bollinger_band_perce
4. https://www.investopedia.com/terms/t/tsi.asp
5. https://school.stockcharts.com/doku.php?id=technical_indicators:true_strength_index