Project 8: Strategy Evaluation

Nathan Riojas
nathanriojas@gatech.edu

1 INDICATOR OVERVIEW

The simple moving average (SMA) is the average of data over a subset of time or lookback period, with each date containing its own subset to be used to calculate the SMA. To achieve the best results for the SMA, the lookback window must be optimized. The reason for this is that the SMA can lag. The SMA needs to be further adjusted to create signals for any algorithm trader. This is done by dividing the price at the current date in question by the SMA at that date. This yields the price per simple moving average (PPSMA). Ideally, when no lag exists, a PPSMA greater than 1 indicates a sell signal because the price would be expected lower to move towards the average. Conversely, a buy signal would occur when the PPSMA is less than 1 and the stock is expected to increase in price for the same logic.

Bollinger Band Percentage (BBP) quantifies a stock price as a percentage relative to upper and lower Bollinger Bands – an indicator that shows how many standard deviations a stock is away from the SMA. With BBP calculated, a value of 1 or greater signifies that the stock is equal to or greater than the value of the top band and a value of 0 or lower shows that the stock is equal to or less than the lower band value. Optimizing this requires optimizing the lookback period for the Bollinger Bands, the SMA, and standard deviation, since all of these are what formulate the BBP. This is again to combat extraneous lag. Additionally, BBP should be optimized for the point at which the signal should occur. In the absence of lag, buying should be done at a value less than or equal to 0 since the return to the average is expected imminently. Conversely, selling should be done at a value of 1 or greater for the same logic.

Momentum is the percent change of a stock price over a given rolling period. When momentum is negative, it reflects a downward trend in a stock price. When it is positive, it reflects an upward trend. Furthermore, momentum can be used to determine just how steep the trend is occurring. At a large value of 0.5 or -0.5, it is apparent that a stock is rapidly increasing or decreasing, respectively. As a technical indicator, investors may use momentum to buy or sell according to the trend reflected. It is important to note that a threshold at which the momentum is occurring must be established. This is what was optimized for in the manual strategy explicitly. Additionally, the lookback window for momentum must also be optimized to prevent lag.

2 MANUAL STRATEGY

The stock used throughout this and all subsequent experiments was JPM. Unless otherwise noted, all experiments use a commission of 9.95 and an impact of 0.005. The manual strategy is evaluated with a lookback window of 14 days.

2.1 Benchmark

The benchmark strategy that is used is simply purchasing the maximum position (selling or shorting) of 1000 shares of JPM stock on the first day and holding until the end of the period in question. The hypothesis is that the manual strategy will best this strategy's cumulative return.

2.2 Indicator Rules

The heart of the manual strategy is the development of a combined set of rules for the chosen indicators that dictate when to buy and sell. These rules are as follows:

Sell: PPSMA > 0.95, BBP > 0.85, Momentum < -0.15

Buy: PPSMA < 0.95, BBP < 0.15, Momentum < 0.15

Thus, when all sell conditions are met, the trader will short the maximum number of shares. And when all buy conditions are met, the trader will long the maximum number of shares. For PPSMA, the buy signal used was a value less than 0.95. For the sell signal, the value used was a value greater than 0.95. Values less than one were chosen to compensate for lag due to the relatively large lookback of 14 days, so that the strategy would not wait for a value of one to occur and act too late. For BBP, a similar thought process was introduced. The value used to determine a sell signal was a BBP greater than 0.85 and the value for the buy signal that was used was a BBP less than 0.15. Again, this was done to compensate for lag and ensure that the signal was caught before the actual price moved too close to the average, thereby minimizing the profit to be made. For momentum, the value used to indicate a continued decrease, or a sell signal was -0.15 or lower and the value used for buy signals was 0.15 or lower. A relatively low momentum threshold was chosen to establish a sustainable trend and give confirmation that the trends expected from BBP and SMA values were occurring. It is worth noting that these values were tuned in order to achieve greater cumulative return than the benchmark for the in sample period through trial and error.

2.3 In Sample Strategy Evaluation

As can be seen by figure 1, the manual strategy outperformed the benchmark strategy especially considering the relatively significant difference in ending values. However, it is not perfect. The reason being that an ideal strategy would always have a portfolio value greater than the benchmark.

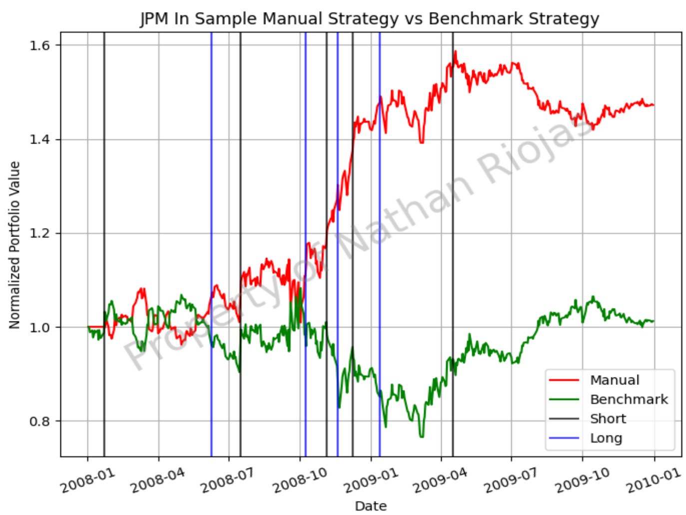

Figure 1—Portfolio value of benchmark and manual strategies over in sample period

Based on the graph, there are certain periods of time when the stock undergoes significant volatility. This rapid volatility exploits the lag weakness of the manual strategy, which appears to struggle to keep up with the fluctuation. However, it is also worth noting that its trading does not start until 14 days in, the chosen lookback time, to wait for data to be present. This could induce a sort of learning curve into the strategy, where the data is still catching up to the strategy and thus producing another form of lag. There is indeed fluctuation through the entire in sample period, and it does seem that the manual strategy is able to catch up eventually, and by the end does indeed best the benchmark.

2.4 Out of Sample Strategy Evaluation

Figure 2 compares the out of sample performance of the manual strategy vs the benchmark strategy. Considering only the beginning and ending portfolio values, the manual strategy out of sample can also be considered successful, just not as successful as during the in sample range. Delving deeper, the first portion of the graph is indicative of the considerable amount of lag the manual strategy experiences. This means that the strategy does not execute trades at ideal points in certain critical moments which would increase the portfolio value constantly.

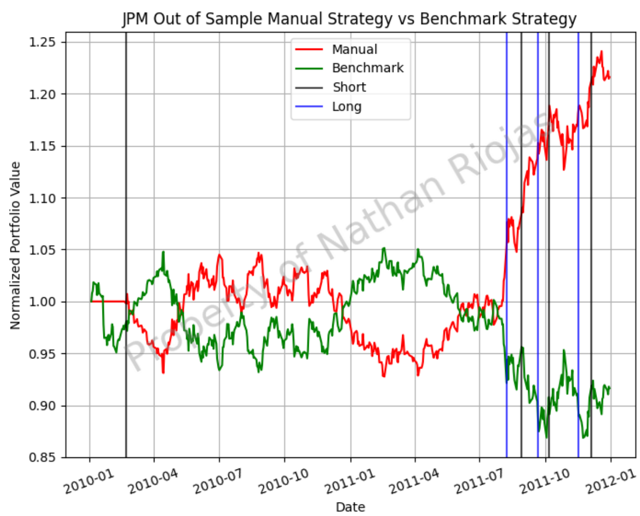

Figure 2 — Portfolio value of benchmark and manual strategies over out of sample period

An obvious explanation for this is that the strategy was tuned for the in sample data, not the out of sample, so it only has one data set to benefit from. Also, just like before, the trading waits 14 days, and a lag could be introduced in the data here again. However, by the end of the range, the manual strategy is consistently increasing and its dips do not mimic the severity of the fluctuations in the benchmark. This means that the strategy is more or less optimizing itself by this point. The takeaway is that this strategy just requires some time to do so in a meaningful and consistent way, since the second halves of Figures 1 and 2 show consistent growth the portfolio.

2.5 Performance Summary

Table 1 presents the values of the cumulative return, standard deviation, and mean of the manual strategy along with the benchmark results for the same metrics. As an additional consideration, these metrics are also presented for various lookback windows to show justification for the lookback window of choice. To begin, all lookback windows of the manual strategy yield a greater cumulative return than the benchmark strategy, with the chosen 14 day window outperforming the cumulative return 48 times over. This is to be expected since the manual strategies use the knowledge of prior data to make decisions to improve portfolio performance whereas the benchmark does not.

Metric	In Sample				Out of Sample			
LB	10	20	14	BM	10	20	14	BM
CR	0.283600	0.337900	0.471900	0.012320	0.171300	0.198300	0.216030	-0.083570
Std	0.013520	0.012840	0.013010	0.017040	0.007480	0.007351	0.007380	0.008500
Mean	0.000586	0.000659	0.000852	0.000168	0.000342	0.000386	0.000416	-0.000137

Table 1 — Performance metrics of manual strategy and benchmark strategy in sample and out of sample with various lookback windows

Additionally, the standard deviation of all manual strategies is less than the benchmark standard deviation. This is because the benchmark is at the mercy of market fluctuation with regards to portfolio value, so it can be highly volatile at certain points. This manual strategy most likely experiences a smaller standard deviation since it slowly grows the portfolio and any losses do not cause too great of losses on specific days.

Much like the cumulative return, since the manual strategy actively seeks to implement rules to enter long and short positions as ideally as possible, the mean for each manual strategy is greater than the mean of the benchmark portfolio.

Furthermore, when comparing in sample to out of sample manual strategies, it can be seen that the out of sample values for cumulative return and mean are not as great as those in sample and that is to be expected since this was tuned for the in sample data. So the in sample performance is indeed better than out of sample.

3 STRATEGY LEARNER

3.1 Framing the Problem

The strategy learner used for this paper was implemented using a Random Forest learner – a Bag Learner of Random Tree learners. With any tree learner it is crucial to frame the data to be used in a manner that separates it into features and results. The results being what is to be optimized for. This is obviously the cumulative return. The features are the variables that together ideally produce some sort of correlation to the results. In this problem, the features are the indicators chosen. Furthermore, the bag learner is used to create more data by simply slicing the training data in random ways, or bags. This exposes the learner to various new subsets of data.

From there, using the in sample date range, x training data and y training data are generated, and subsequently input into the initialized bag learner of random tree learners. This builds the tree. Next, the tree is queried with an array of prices generated from out of sample data to create a dataframe of trades to be returned.

3.2 Hyperparameters/Tuning Values

The specific hyperparameters that are input into the bag learner and (by passing) random tree learner itself are the leaf size and the number of bags. The leaf size specified in this strategy learner was 8 due to some analysis from Project 3. It was found that leaf sizes under 8 are generally where overfitting begins. Conversely, the higher the leaf size, the larger the root mean square error becomes when working with bag learners. Finally, the smaller the leaf size, the faster the random tree training time becomes. The number of bags chosen was 20 because the amount of data the in sample data range provides contained only around 500 data points and it was desired to mimic a larger dataset for accuracy. Larger than this number could have slowed the training time, which was not desired. Additional parameters tuned for the actual strategy learner itself were the lookback window, testing window, and minimum cumulative return required. The lookback window is the window the indicators used. The testing window, although training window would probably be a better name, was the number of days the training data would peer into the future stock price to populate the y training data based on cumulative return. The minimum cumulative return was the minimum cumulative return the testing window should provide to signal to short or long a stock (i.e. populate the y train data with a -1 or 1). The lookback window and testing window were set to 5. This provided a shorter lag period and a relatively short time to look into the future in order to train the strategy learner to quickly act on indicator signals. The minimum cumulative return was set to 3% which may seem small but was large enough that a large enough amount of small trades accurately assessed could provide large profit.

3.3 Discretizing Data

To create and use the learner discretization needs to occur at two points. The first is when the value of stock prices in sample is converted to y training data, and the second is when reading the queried data from the bag learner. The training method, called add_evidence in this implementation, accepted a stock to analyze along with a date range for the in sample period. The x training (features) data was developed using a dataframe containing columns of the indicator values at each day. In order to generate the y training data, a for loop iterates over all the dates within the in sample range. Within this loop, the price 5 days (specified lookback) in the future is used in conjunction with the current price to calculate the cumulative return over that period. Once the cumulative return is found, the logic for discretizing the training y data is implemented. If the cumulative return is greater than a specified minimum plus the value of impact, then a 1 is specified for that date. If it is less than the negative of the cumulative return plus the impact, a -1 is specified for that date. Otherwise, a zero is specified for that date. Since this training data is populated throughout with a possibility of only three discrete values (-1,1,0), this problem becomes best framed as a classification problem, so the random tree learner was adjusted to use the mode instead of mean when building its tree.

The testing method is found within the testPolicy method of the strategy learner. Given the date range for the out of sample data, a prices dataframe is generated and used to query the tree made within the add_evidence method. The returned value is the y test data, which must be discretized. An initial holdings variable is set to zero and a trades dataframe is created to subsequently be populated. This is done by looping over the values of the y test data. A value greater than zero is a long signal, less than zero is a short signal, and equal to zero is a cash signal. To convert this to a final trades dataframe, each signal is checked for the situation where the current holdings are not the same as the position the signal suggests.

For example, during a long signal, if the holdings are -1, the trades dataframe is given a value of 2000 to enter the maximum position of 1000 shares, and if the holdings are 0, the trades dataframe at the specified date is set to 1000. The same logic is applied when the signal is a short or cash signal.

3.4 Note On Implicit Indicator Strategy

Whereas the manual strategy tells the strategy exactly when to trade, the strategy learner is given the indicator values at each day with the specified lookback for the indicator calculation and shown the outcome of the cumulative return several days in advance. This allows the tree learners to associate specific indicator combinations with specific returns, thereby implicitly developing a strategy for indicators. Because of this, short windows are more beneficial as a larger lag will hinder the learner's learning potential.

4 EXPERIMENT 1

4.1 Summary of Experiment

The purpose of this experiment is to compare the performance of the strategy learner, manual strategy, and benchmark strategy using the in sample date range. The strategy learner is given the in sample date range to add_evidence to train the bag learner, and then the testPolicy method is used to query the same in sample date range to get a trades dataframe. This dataframe is plotted along side the benchmark and manual strategies from Section 3.3.

4.2 Hypothesis

The strategy learner should outperform both the manual and benchmark strategies. The reason is because the strategy learner is more adaptable to various situations, due to the implicit learning of the indicator relation to cumulative return as previously discussed. It is also specifically trained on this dataset.

4.3 Results

Based on Figure 3, it is evident that the strategy learner is the best of all of the strategies implemented in this paper during the in sample period, as hypothesized. One highlight is that the strategy learner outperforms the benchmark strategy throughout the entire period. Apart from around March 2008 and June 2008, the strategy learner outperforms the manual strategy the entire time.

Nevertheless, the strategy learner is to be expected to outperform these strategies every time within sample data because the strategy learner is trained on the in sample data. It also can be trained on smaller lookback windows for its indicators. This could minimize the lag it experiences compared to the lag the manual strategy is subject to.

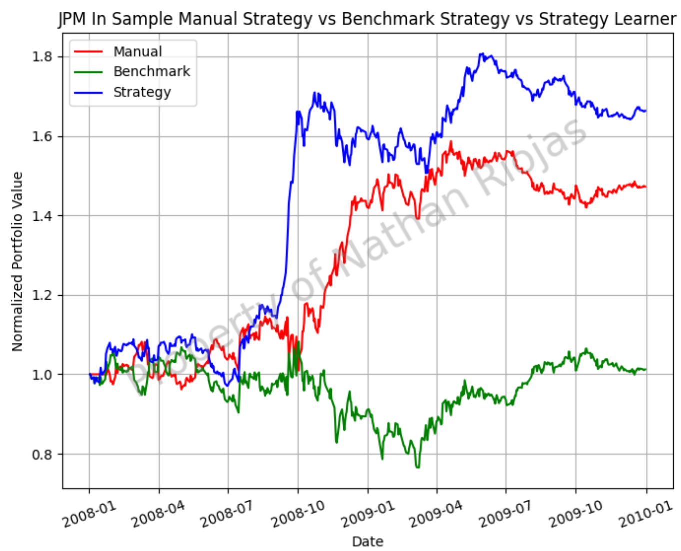

Figure 3—Comparison of benchmark strategy, manual strategy, and strategy learner performance in sample

5 EXPERIMENT 2

5.1 Summary of Experiment

The purpose of this experiment is to show the effect market impact can have on the strategy learner. To test this, five strategy learners are trained and tested using the in sample date range, each with different impact values (0, 0.005, 0.01, 0.015, and 0.02), but with a commission of 0. Their testPolicy dataframes are then used to generate a cumulative return dataframe to be plotted.

5.2 Hypothesis

The more reactive a market is to a learner's trades the more volatile it will become, and the trades performed will become less and less effective. Therefore, the portfolio values will be worse the greater the market impact.

5.3 Results

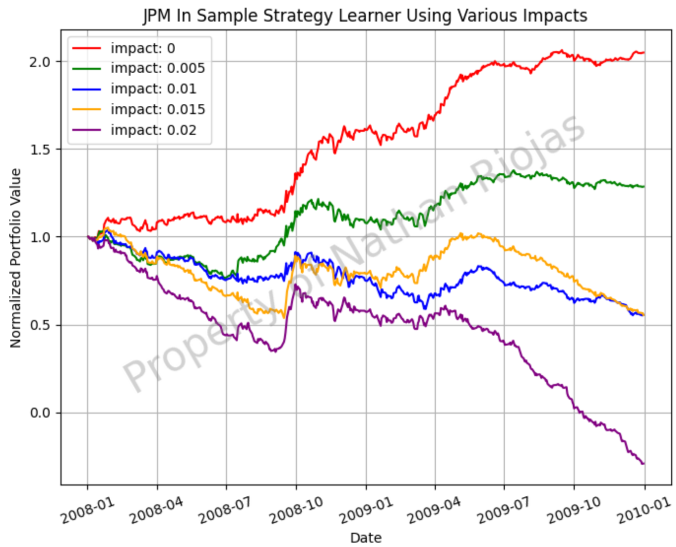

Figure 4 — In sample portfolio performance of strategy learners with varying impact values

Figure 4 shows that the hypothesis was indeed correct, because the larger the impact is, the worse the portfolio does over the in sample period. Conversely, at a value of 0, the most ideal condition, the learner does even better than in the previous experiment. Table 2 analyzes this further using two metrics, cumulative return and standard deviation.

Impact Value	**Cumulative Return**	**Standard Deviation**
0	1.0487	0.0117
0.005	0.2853	0.01449
0.01	-0.44609	0.019253
0.015	-0.4462	0.019149
0.02	-1.291	0.273065

Table 2 — Performance metrics of the strategy learner with varying impact values in sample

In reviewing the actual values of these metrics, just within just two increments of 0.005, the cumulative return has already become negative (-0.44609). This illustrates how quickly market impact can have on turning a successful strategy learner into an ineffective algorithm.