Project 8: Strategy Evaluation Report

Daniel Crawford
dcrawford46@gatech.edu

1 TYPOGRAPHY

The stock market is one of the greatest puzzles yet to be solved. But it also holds the greatest prize for those able to so. There have been many who have claimed some strategy for profiting more than could be expected from the market, and a few who are hyper-successful in their strategy. As the advent of Machine Learning, gives way to an age AI, we see that it is time to use these methods to try to crack the case of the Stock Market. Here we will first establish a set of predictors which will set up a framework for indicating when to buy or sell a stock. Second, we will manually select the parameters that go with those predictors and see how well our strategy does. Finally, we will hand over the problem to an optimizer engine, that will find the optimal value of those parameters. Note that we will be using return as our measure of success.

2 INDICATORS

Carrying over the indicators and descriptions used in Project 6, I will be using the following indicators:

2.1 Simple Moving Average (SMA)

We start with the simplest moving average, SMA. This is simply the average over a given window of time:

$$SMA = \frac{p_1+p_2+...+p_n}{n},$$

where p_i is the price at time i and n is the window size. The SMA captures the general trend of the stock's price and as a result helps to smooth out volatility that can come with high degree of granularity. The larger the window, the more that this effect is evident. As such, the SMA provides an indication of what the movement of the stock maybe.

2.2 Exponential Weighted Moving Average (EWMA)

The next natural step after SMA would be to find a way to focus on more recent values than previous values, when trying to predict. The assumption is that

recent values will have more of an affect. To do this, we will implement EMA, which accomplishes the goal by:

$$EMA_t = s(n)p_t + \big(1 - s(n)\big)EMA_{t-1},$$

$$where\ s(n)\ = \frac{2}{1+n}, which\ is\ a\ smoothing\ factor$$

This factor allows for more flexibility in the indicator, by adjusting the current price, p_t according to the most recent prices. The smoothing factor in our case is a shown, and is a common one. This is analogous to the window size in SMA, and set how wide we may want to keep out windows. The n reflects the number of days in our window.

2.3 Double Exponential Moving Average (DEMA)

To eliminate lag in EWMA, let us extend the notion of EWMA to DEMA, which, through the doubling, removes the period of lag:

$$DEMA\ =\ 2\,EMA_n\ -\ EMA(EMA_n)$$

As described by this formula, we double the EMA of our window of n periods, and then calculate the EMA of that. That is, if we look at EMA as focusing on the recent periods of time, we calculate that value, but then remove the more recent more recent volatility, which gets rid of some lag. The n again reflects the number of days in our window.

2.4 Triple Exponential Moving Average (TEMA)

We will further extend our notions of exponential moving averages in such a way as to create a TEMA. TEMA also takes out the lag, like DEMA, but has the added benefit of indicating a upward movement of the price when that same price is above the TEMA, as shown here. What is more, that TEMA can follow the price better than the previous discussed, indicating that lag is reduced. As a result, we can use TEMA to indicate short term results, and can work well for those seeking out smaller, short term, advantages.

2.5 Quadruple Exponential Moving Average (QEMA)

Venturing farther into the realm of abstraction, we find QEMA, which is given by:

$$QEMA = 5 * EMA_n - 10 * EMA(EMA_n) + 10 * EMA(EMA(EMA_n)) - 5 * EMA(EMA(EMA(EMA_n))) + EMA(EMA(EMA(EMA(EMA_n))))$$

We get a bit more abstract here, but Project 6 suggests to us that QEMA does not provide as much smoothing as the other indicators, but does appear to indicate a trend with relation between different windows.

2.6 Indicator Parameters

The parameters for each of these indicators is rather straight forward: It is the number of days in the window that we looking in. For each class of indicator, we will have a Short (S) Window and a Long (L) Window. When ever it is the case the moving average of the short window goes above the moving average of the longer window, it is a sign that the asset is on the rise, and it would be an opportune time to buy, symmetrically opposite for selling.

If *MA$_S$ > *MA$_L$ then LONG

If *MA$_S$ < *MA$_L$ then SHORT

3 MANUAL STRATEGY

3.1 Combining Indicators

Because we have multiple indicators and each one has its own strengths and weaknesses, we will implement a majority-based system. That is, whenever it is the case that a majority (at least 3) of the indicators, suggests an action (Long or short) we will take that action. If only two or one say to, we will not. This allows for flexibility and will hopefully generate some insight.

So we now have a system where we enter a position when the majority of indicators suggest Long, and exit when Majority say Short. I think that this will be an effective strategy because the different moving average windows capture different things and can be sued to weight different aspects of the price. Things like lag and smoothing are important, and by combining them into a majority will allow us to take all of those features into consideration.

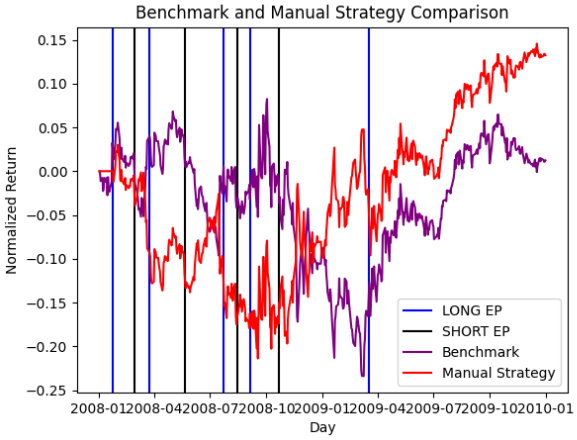
I chose the majority (3+) of the indicators because I thought that this would be a strong enough signal, but not one that would cause either an unreasonable

number of trades to occur, or be over run by a single indicator switching. Further having a simple majority helps balance between longing and shorting a stock. If we were not doing a simple majority, and instead need 4 to vote long, and 1 to vote short to go short, we would be experiencing a great deal of shorting, which may not be an optimal balance and lead to an optimal result.

Also, and important reason for the choice of these features is their popularity. The moving average family of indicators are widely used and written about for their effectiveness and simplicity. I think that adding on this voting layer adds a bit more complexity, but will have positive results.

3.2 Testing Manual Strategy

To test this strategy and see the results, we used the given data of JPM Stock from the in-sample period of 1/1/2008 – 12/21/2009, to apply our strategy to, and comparing that to the benchmark of just buying 1000 shares (max long position) and holding them, we generate the following plot:

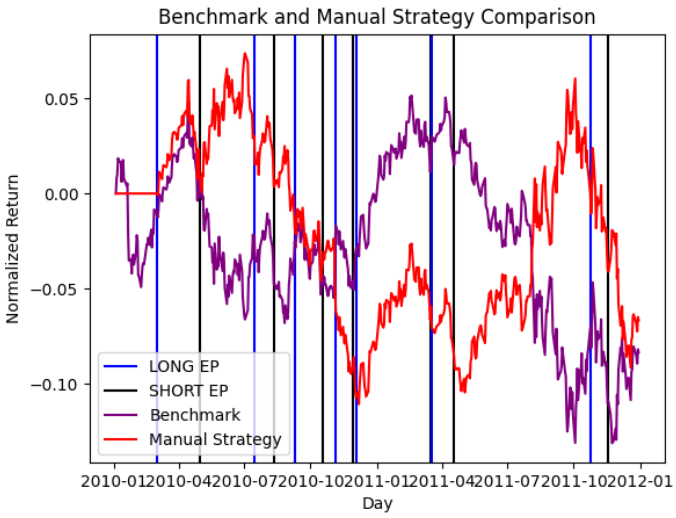


This plot shows not only the cumulative returns of the Manual Strategy compared to the benchmark, but the entry (long) and exit (short) points of the stock.

Examining this plot, we see that for about the first year, the manual strategy was being outperformed by the benchmark. However, at the end of the in-sample period, it appears the cumulative return for the manual strategy was greater than that of the benchmark. This seems to be due to the fact that the indicator method was able to pick up on the downtrend of the stock's value just short of half-way through the simulation, and take a short position, and then quickly pick up on the uptrend. This appeared to be effect for the in -ample test.

To get this result, recall that parameters for the indicators must be used. After manually tweaking and testing, I used S = 13, and L = 130. (For EACH individual

indicator) This means that the window for all of the 'small' windows of the moving averages was 13 and all of the 'large' windows were 130 days. I wanted to have them be a large order of magnitude in difference, and started looking around the suggested 12/100 and the 12/50 which I used in project 6, but found that 13/130 worked well.



However, this success may just be due to that time period, so below is an out-sample comparison of the same benchmark (+1000 and hold) and Manual Strategy (13/130).

We see that there are distinct periods when one strategy is performing better than the other, but the manual strategy ends up with a slightly larger cumulative return than the benchmark.

I think that the differences observed in both the in and out sample period are due to the Manual Strategies ability to capture when the stock price was increasing or decreasing in value and adjust accordingly. Recall that this is indeed the goal of the moving averages. By using a balanced majority, I was able to capture more smoothly, and more robust to lag, the times that the value was fluctuating, and from that the strategy was able to make a better decision than the benchmark.

To quantify these results consider the table below which shows various statistics. We know that the cumulative return comes out high in both samples for the manual strategy. But, interestingly, in sample the standard deviations of daily returns is the same. However, in out sample, the standard deviation is lower for the manual strategy.

For both samples the mean daily return of the benchmark is the same, but different for the manual strategies. This is due to me tweaking only looking at the in sample period. By virtue of the sensitivity of the mean, the mean of daily returns is negative for the manual strategy on the out sample.

Portfolio	Cumulative Return	St. Dev. of Daily Returns	Mean of Daily Returns
In Sample Benchmark	1.0123	0.017	0.000168
In Sample Manual Strategy	1.1323	0.0173	0.000395
Out Sample Benchmark	0.9166	0.017	0.000168
Out Sample Manual Strategy	0.9335	0.0078	-0.000106

4 STRATEGY LEARNER

4.1 Framing the Strategy Learner

Originally, I had planned to use a QLearner, but after much testing found not only convergence difficult, but reasoned that the problem may be better suited to an optimizer solution. Therefore, I utilized an OptimizeLearner to optimize the parameters for my indicator method, which, recall, were the short look-back window (S) and the long look-back window (L).

Another realization I had, was to make these different for *each* individual moving average. As such, I have 10 parameters, 2 for each indicator. My simple majority strategy will stay the same for each indicator voting on position.

Because the optimizer method that I used uses a numerical estimation of the gradient and the step sizes are rather small, I normalized the parameters from 0 to 1, which scale from 1 to 201. I chose to cap at two hundred because the simulation

run was only about 700 days long, (with only about 500 trading days) and I did not want the window to be unusably large.

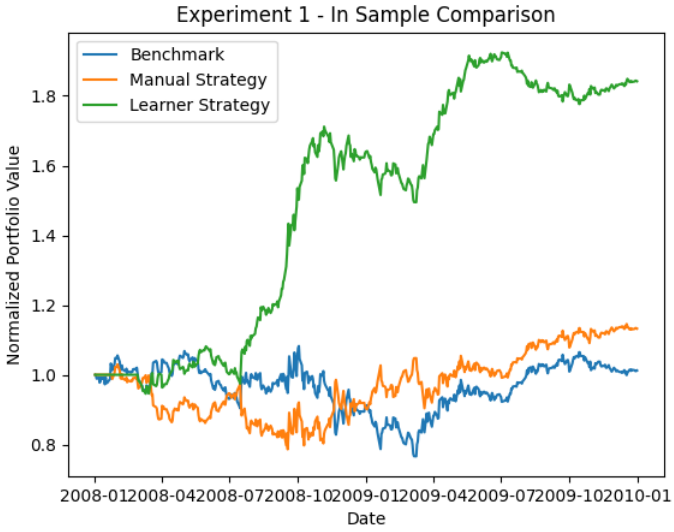
The hyperparameters I used for my optimizer are as follows:

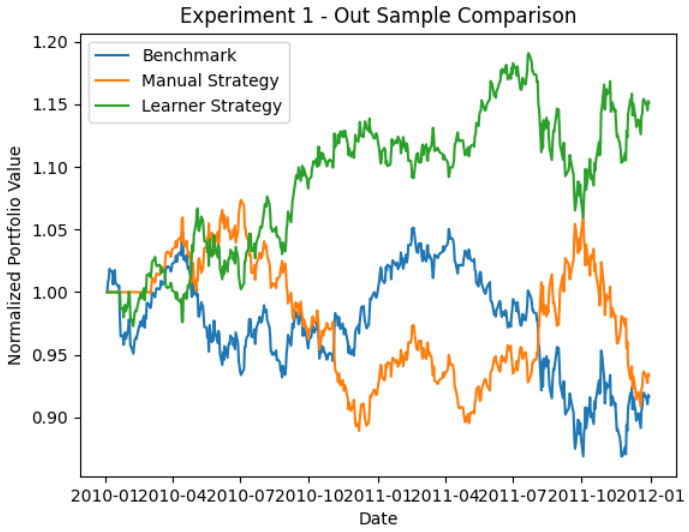
- Objective: I opted to not use the Sharp Ratio, as I did not want to look into optimizing (decreasing) volatility. Because this learner could be built into an auto trader, and not have to be done manually, I would accept higher volatility for greater return.
- Method: I used Sequential Least Squares Programming. I used it in project 6 to good effect, and it was able to converge within the time frame. This method iterates over possible solutions for non-linear constrained problems.
- I set a very small tolerance for convergence (1e-20)
- I increased the max iterations to 1e8, to allow the algorithm time to move the windows of the indicators according to the gradient.

Again, to standardize the data to be used, I scaled the constraints of the values from 0 to 1 to go from 1 to 201. This allows the windows to better fluctuate in sizes. (Whenever I ran the indicators on the windows, I rounded to the nearest integer, which allows for the operation to be valid.)

5 EXPERIMENT 1

For this experiment, I compared the performance (measured as normalized value) of the benchmark (hold +1000 from start), manual strategy (with small window S = 13 and large window L = 130) , and learned strategy (described above) on the in sample period and out-sample period. (Note that only in sample data was used to train the learned strategy. The plots are shown here:





We see that on the in-sample data, the manual strategy slightly out performs the benchmark, as to be expected form tinkering with is, as discussed previously. The learner strategy almost doubles the value of the bench mark strategy however. This suggested that the optimizer was rather powerful in finding the best windows to use for the in-sample time period. (Though dis so with 50-15- orders, compared to the dozen or so in the manual strategy.)

We see that the learner strategy, which applies the same windows, again does marginally better in terms of portfolio value than the manual, which in turn out performs the benchmark. (Though note the different axes scales.)

In both cases, it appears that the windows that the learner found were able to capture trends that occurred early, and take advantage of these shifts to create a large advantage. It does look like the learner had trouble in the second half of the out sample simulation, but was able to recoup some of the value lost.

A natural question would be as to the values of the window sizes that the learner found. These are provides on the table on the side. An obvious feature is that for three of these, the S is larger than the L. This is certainly not customary, as it goes against the

Indicator	**S**	**L**
SMA	10	1
EWMA	133	1
DEMA	200	1
TEMA	1	81
QEMA	1	200

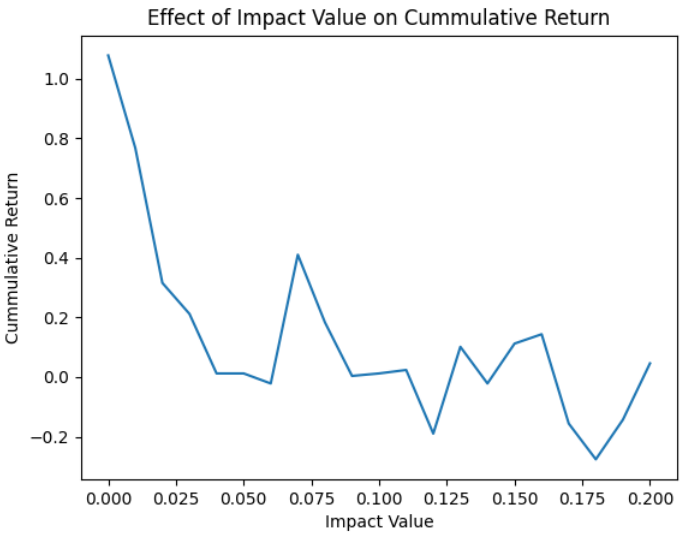
intuition laid out in the first section. Also, the window of 1, for either S or L does not combine well with intuition. But, the results are none the less effective. I think that the optimizer picking up on some small change that would be difficult to explain to a human, but obviously made sense with the date. Further, I was able to have success on multiple assets with the same configuration of optimizer so I think that the fact that the optimal solutions found are not intuitive would not warrant discount.

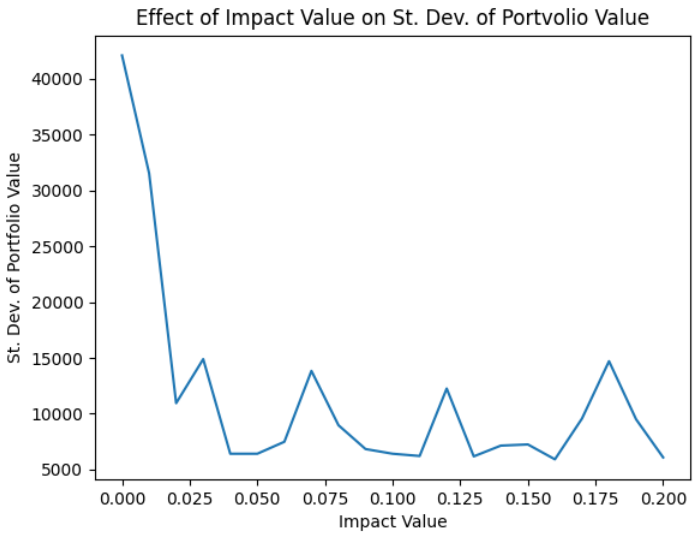
I think some of the oddity can be explained by my initialization of S to 10 and L to 20 for each individual indicator. I tried, with a little success to change the initial values of the windows, so I think it is likely that I found a local optima for these initial conditions. I would expect the same result with the same starting conditions and ones similar, but not all. However, none of the other ones I found were as successful (on the in-sample period) as 10/20.

6 EXPERIMENT 2

For experiment 2, we looked at how changing the value of impact would affect in-sample trading. To look at this I looked at the cumulative return and standard deviation of the portfolio. I chose cumulative return because it is close to the normalized value we have already been working with, and it may be considered the truest measure of success of a portfolio. I also included the standard deviation of the portfolio. I wanted to capture the volatility of the portfolio, but not with the Sharpe ratio as I do not want to roll up value and volatility into one, so I chose the standard deviation.

I would expect that with a larger impact, which is the amount the price moves against us when we buy, the learner would achieve lower cumulative return, and lower volatility. This is because as impact grows, so does the implicit price of the stock, which would mean that we are having a larger cost of the transaction decreasing our return and which would make it less beneficial to buy often, suggesting less entry and exit points, meaning less volatility (measured in standard deviation.) We are using the starting value of $100,000 , in sample time period, and simulation for impacts of 0%,1%,2% to 20% (Somewhat of an arbitrary stopping point, but 20% is quite larger for an impact. Also, note that I did not want





to show standard deviation normalized, as I wanted to emphasize the swings in value that were experience, which is accomplished by showing the raw values.)

The plots here confirm my hypothesis. We see the cumulative return does indeed decrease as the impact grows, probably due to the reasons already put forth. Also, the standard deviation shows decreasing value after the first bit, and from there levels off around 10,000 (or 0.1) This rather a large swing for lower value of impact, and suggests that the learner may be more appealing to an aggressive trader.