Project 8: Strategy Learner

Zhihua Jin
zjin80@gatech.edu

1. Describe the steps you took to frame the trading problem as a learning problem for your learner. What are your indicators? (They should be the same ones used for Manual Strategy assignment) Describe how you discretized (standardized) or otherwise adjusted your data. If not, tell us why not.

I chose to use Random Forest approach and followed the hint provided by the course site ("Summer 2019 Project 8: Strategy Learner ", 2019). As stated in the hint, I turned my previous learner to a classification-based one. The leaf_size parameter of my decision tree learner was set to 5 to avoid overfitting in in-sample period. Besides, bagging is utilized to achieve optimal return. After parameter tuning, the number of bags was adjusted to 25. The initial impact for training is set to be 0.

The three indicators I chose are:

- Price/SMA: It includes two parts: simple moving average (SMA) and the adjusted stock closing price. SMA is the moving average calculated by adding recent closing prices together and then dividing that by the corresponding time range.
- Bollinger Bands: The Bollinger bands feature value can be calculated using following formula ("Machine Learning for Trading | Udacity", 2019)

$$BB[t] = \frac{price[t] - SMA[t]}{2 \cdot std[t]}$$

- Momentum: The momentum of stock market could be calculated by taking price differences for a fixed time interval continually.

Following the same indicator code written for Manual Strategy, the window size for the price/SMA and Bollinger bands feature value are 10 days, while for the momentum is 3 days. These parameters are exactly the same in Project 6. I could assure you the figures in both reports are generated correctly using the same parameters and lookback window, despite the typo in previous report stating all three use 10-day window. I apologize for that.

The training steps taken to frame the trading problem are:

1) Use these three indicators together as input X data.
2) Calculate N-day return (my choice of N is 10 days). To standardize the data among different stocks, the future N-day returns (price[t+N]/price[t] -1) was used, instead of a normal N-day return.
3) Discretized/Classified Data: Find appropriate value for Y based on the N-day return and classify Y into three categories: +1 (LONG), -1 (SHORT) and 0 (CASH). If the Nday return exceeds a certain value, the trader classifies the day as a +1 and triggers a YBUY. Similarly, if the return is too low, then the trader classifies the day as a -1, triggers a YSELL signal and sells the stock ("Classification Trader Hints ", 2019). If neither the threshold is met, the learner classifies that day as a 0 and just does nothing. This part is more experience-based. I tried to find the ideal value for both YSELL and YBUY by observing the stock volatility and price trend, calculating the N-day returns of several stocks, etc. Tuning YBUY and YSELL is for maximizing the corresponding cumulative return. In the meantime, bag number and leaf size were also adjusted.
4) Put both X and Y into the Random Tree learning model, train with bag learner and let the learner learn the strategy. The data used is from January 1, 2008 to December 31 2009 with start value 100,000. To deal with NANs, I did not trade until after the lookback period when I have valid calculated values.

In the subsequent testing phase, for the selected date range, the indicators are calculated again. Then, using the learner generated in the training phase, it would initiate the trader, decide whether to buy/sell/hold each day according to the strategy learner and return the date frame with values representing trades for each day. Values that range from +2000 to -2000 are legal as long as the net holdings are one of the following: -1000, 0, +1000.

Experiment 1

Using exactly the same indicators that you used in manual_strategy (trade JPM), compare your manual strategy with your learning strategy in sample. You can use the same impact (.005) as was used for Project 6 or use 0 for both. Be sure to add in an author method.

For experiment 1, I compared this machine learning strategy to the manual strategy I devised in Project 6 against benchmark. The impact for the comparison is 0 for both, and there is no transaction fee. Date range from January 1, 2008 to December 31 2009 is the same used for the training, so it is an in-sample test. The traded symbol for this experiment is JPM. The leaf size is 5 and the bagging is 25. Other parameters are the same as mentioned in the first part of this report. The trading rules also remain the same.

My hypothesis is that both manual strategy and strategy learner outperform the benchmark significantly in in-sample period. Besides, strategy learner performs better than manual strategy. To demonstrate, the normalized portfolio value using different strategies are plotted in this time range and shown in Figure 1.

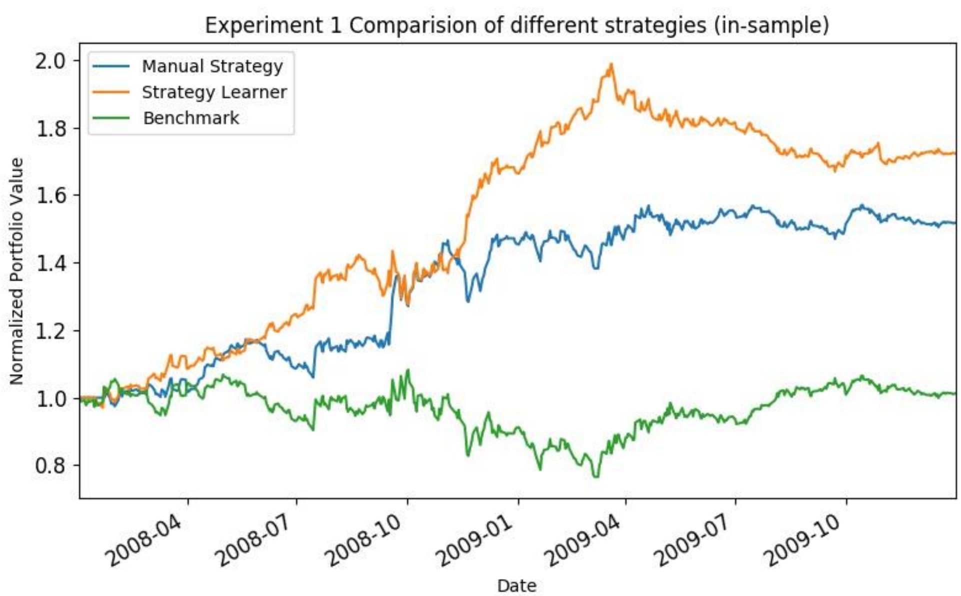

Figure 1. In-sample comparison of different strategies with JPM from January 1, 2008 to December 31 2009.

It could be seen that strategy learner outperforms manual strategy in most cases. In the end of year 2008, the two strategies perform similarly. The advantage of strategy learner is not obvious in 2008, but it becomes clearer in 2009. Moreover, these two strategies both perform significantly better than the benchmark.

Besides, several parameters are calculated as well and shown in Table 1. Judging from the sharpe ratio, both strategies could be considered good by investors. But strategy learner is even better than manual strategy.

Table 1. Cumulative return; standard deviation & mean of daily returns and sharpe ratio of the benchmark, strategy learner and Manual Strategy from January 1, 2008 to December 31 2009.

Strategy	**Cumulative Return**	**Stdev of daily returns**	**Mean of daily returns**	**Sharpe Ratio**
Manual Strategy	0.518	0.0120	0.0009	1.1941
Strategy Learner	0.723	0.0111	0.0011	1.6374
Benchmark	0.012	0.0170	0.0002	0.1572

I ran experiment 1 several times and the exact value of these parameters in strategy learner changes every time. This is understandable since it uses the random tree approach with varying splits, and this can lead to different results. In some cases, the manual strategy beats strategy learner by a tiny margin in the end of 2018, which is shown in Figure 2. In extreme cases, the strategy learner performs even worse and gains little return. Considering the cumulative return, the relative advantage of strategy learner remains in most cases, with certain probability of worse performance than manual strategy.

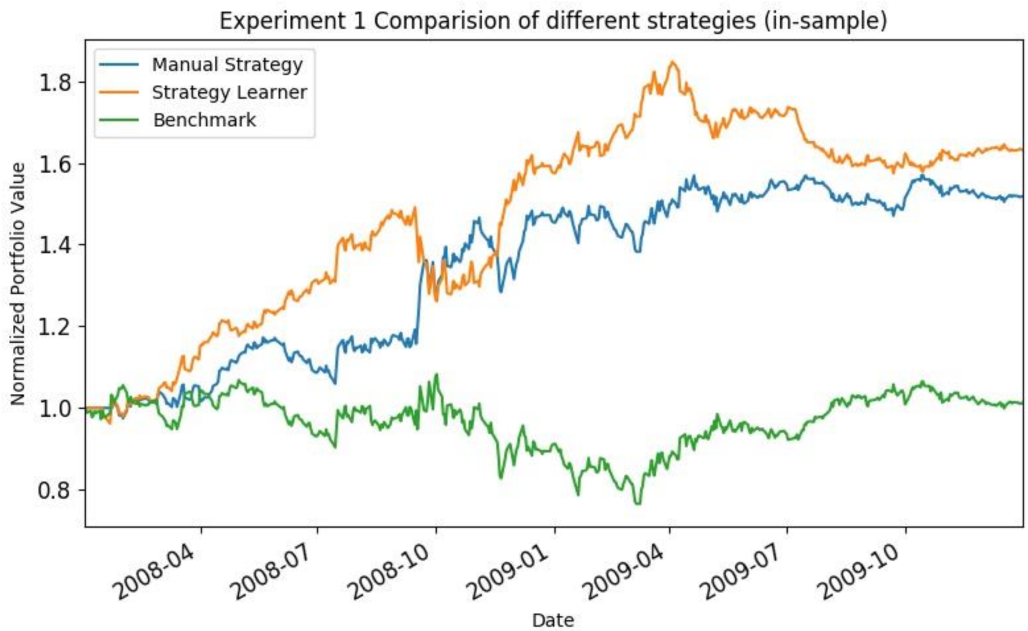

Figure 2. Another In-sample comparison of different strategies with JPM from January 1, 2008 to December 31 2009.

Experiment 2

Provide a hypothesis regarding how changing the value of impact should affect in sample trading behavior and results (provide at least two metrics)

My hypothesis is that if the impact increases, the portfolio value decreases. Moreover, the average daily return decreases while its sharpe ratio also decreases. When there is completely no impact and when the impact is too huge such as 0.5, the trading behavior and results would be very different. So here I only conducted the experiment with in-sample JPM data using impact from 0.0005 to 0.05. Other parameters are unchanged. The trading rules also remain the same. As shown in Figure 3, the strategy learner with smaller impact generates higher portfolio value in most days. But the difference between impact 0.005 and 0.005 is not very remarkable. For impact 0.05, the difference is more significant as in some days the portfolio value is a negative number.

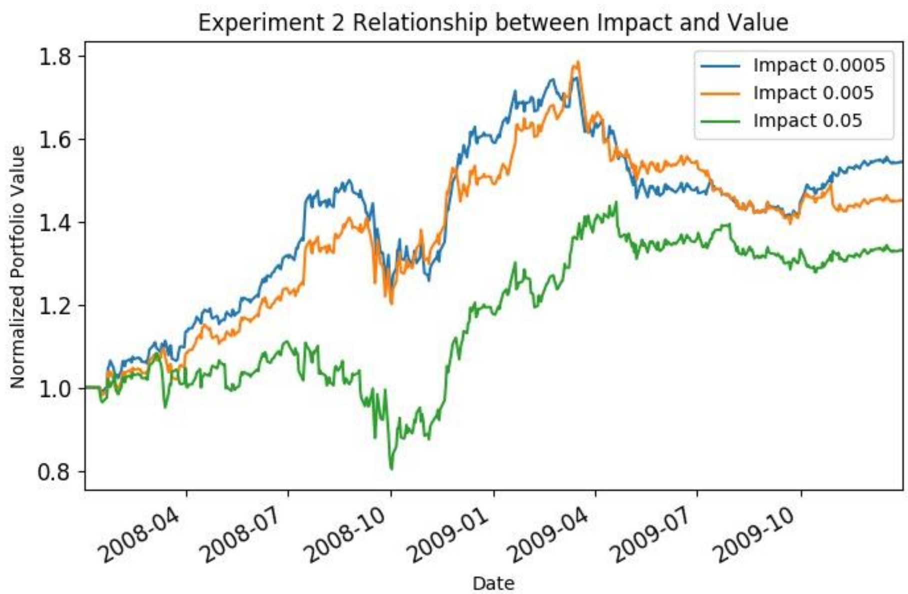

Figure 3. In-sample comparison of different impact with JPM from January 1, 2008 to December 31 2009.

Several parameters are calculated as well and shown in Table 2.

Table 2. Cumulative return; standard deviation & mean of daily returns and sharpe ratio of strategy learners with different impact from January 1, 2008 to December 31 2009

Strategy Learner's Impact	Cumulative Return	Stdev of daily returns	Mean of daily returns	Sharpe Ratio
0.0005	0.542	0.0115	0.0009	1.2748
0.005	0.449	0.0119	0.0008	1.0791
0.05	0.329	0.0171	0.0007	0.6615

We could see that when the impact increases, the cumulative return and average daily return both decrease. Meanwhile, the standard deviation of daily returns increases, while the sharpe ratio decreases. This means the return becomes more unstable when impact increases. Judging from the sharpe ratio, strategies with impact 0.0005 and 0.005 could be considered good, since the ratio exceeds 1, which is a sign of considerable return despite the risk taken. Besides, the performance of strategy with 0.0005 is better than impact 0.005, since sharpe ratio of impact 0.0005 is higher. For impact 0.05, the cumulative return is still a positive number, but the sharpe ratio is below 1, which is not optimal.

Certainly, since the strategy uses random tree, the value of these learners varies every time. But the relative tendency of results remains.

Reference

Machine Learning for Trading | Udacity. (2019). Retrieved from https://www.udacity.com/course/machine-learning-for-trading--ud501

Summer 2019 Project 8: Strategy Learner (2019). Retrieved from http://quantsoftware.gatech.edu/Summer_2019_Project_8: _Strategy_Learner

Classification Trader Hints (2019). Retrieved from http://quantsoftware.gatech.edu/Classification_Trader_Hints