Report: Project 8

CS7646 Machine Learning for Trading, 2019 Spring

Hui Xia (hxia40)
903459648
Georgia Institute of Technology

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.**

To convert a trading problem into a learning problem, one need to firstly build a proper model using exiting trading data, then predict trading movements by querying the model. Thus, in this project, I firstly trained a bag learner (20 bag random learner bundle, leaf size of the random learner equal to 5) using the training data, then queried the bag learner. The bag learner used values of four technical indicators and 14-day return of the target asset as the training data. Among the n columns of the data, the last column (14-day-return) was referred as the *Y* column, which represents the result that to be decided by the rest of the columns (i.e. the *X1, X2 … Xn-1* columns).

The technical indicators used in this project were directly adopted from **Project 6**. In short, they are Price/Simple Moving Average(SMA) ratio, Relative Strength Index (RSI), Money Flow Index (MFI), and the CBOE Volatility Index (VIX), were used. The details on how they were coded, the purpose of using each, and how they were implemented, could be found in my report for **Project 6.**

To standardize the *X* columns (i.e. the technical indicators), I firstly normalized the price of the target asset against its price on the first trading day of training data (in the method `addEvidence`), then calculated the technical indicators thereof (in the method `testPolicy`). The look-back of the indicators SMA, RSI, and MFI are 14 trading days. To endure that these indicators would be available on the first trading day of the data, all the indicators were calculated 45 calendar days ahead of the date of the first trading day. For example, if the first trading day of the data is Jan 2nd, 2008, all technical indicators would be calculated from Nov 19th, 2007. As bag learner using random learner is a classification model, to discretize the training results, I re-built the *Y* column in a classify manner based on 14-day return in price. If on a given date, the 14-day return is higher than 8%, Y value on that day will equal to 1; if the 14-day return is lower than -8%, Y value on that day will equal to -1; and the Y value on that day will equal to 0 on other cases.

The training data was then queried against the test data (which was built using the exact method) using the bag learner. In the query process, mode (rather than mean) of the data points was calculated, to ensure the resulted *Y* column will only contain categorized values (i.e. 1, 0, and -1). The resulted *Y* column was then used to generate a dataframe of trades (i.e. `df_trade`) to make trading decisions and hence provide a solution to the trading problem.

2. **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.**
    - **Describe your experiment in detail: Assumptions, parameter values and so on.**
    - **Describe the outcome of your experiment.**
    - **Would you expect this relative result every time with in-sample data? Explain why or why not.**

2.1. Experimental Methods

For strategy learner, the training data was feed to `StrategyLearner.py` by calling the `addEvidence` and `testPolicy` methods to generate `df_trade_sl`, as described in **Section 1**. In the strategy learner, if on a given date, the 14-day return is higher than 8%, the return on that day will be defined as 1; if the 14-day return is lower than -8%, the return on that day will be defined as -1; for the other cases, the return on that day will be defined as 0. For the bag learner, the bag number equal to 20. For the random learner in the bag learner, leaf size equal to 5.

For manual strategy, using the technical indicators described above in **Section 1**, a set of trading rules was implemented via calling `ManualStrategy.py`. The details on how these indicators were coded, the purpose of using each, and how they were implemented, could be found in my report for **Project 6**. The trading strategy is:

Long entry: when RSI < 30 & price/SMA ratio < 1 & VIX < 50

Short entry: when MFI > 70 & price/SMA ratio > 1 & VIX > 20

Long exit: when RSI > 55

Short exit: when MFI < 55

Then, similar to the strategy learner, a dataframe `df_trade_ms` that adjusts the holdings to long 1000 shares of the target stock if long entry appears, short 1000 shares if the short entry appears, and exit the respective position if long or short exit appears. The `df_trade_sl` and `df_trade_ms` dataframes were then respectively returned to `marketsim.py`.

2.2. Result

Figure 1 demonstrates the manual strategy, the strategy learner, and the benchmark portfolio. Overall speaking, the strategy learner overperforms the manual strategy and the benchmark with significantly higher Sharpe Ratio and cumulative return. The Sharpe Ratio of the manual strategy, the strategy learner, and the benchmark are 0.731, 1.487, and 0.157 respectively. The cumulative return of the manual strategy, the strategy learner, and the benchmark are 0.1766, 0.5691, and 0.0123 respectively. The standard deviation of the manual strategy, the strategy learner, and the benchmark are 0.0076, 0.0110, and 0.0170, respectively. The mean of daily returns of the manual strategy, the strategy learner, and the benchmark are 0.00352, 0.001029, and 0.00017, respectively.

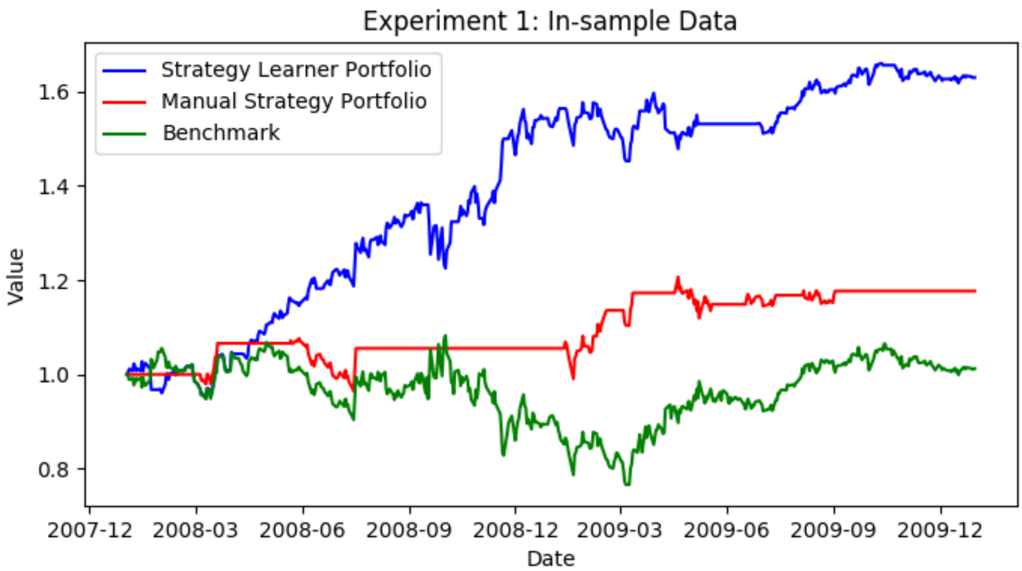

Figure 1: Representative performance of benchmark (normalized to 1.0 at the start, green line) , the manual strategy portfolio (normalized to 1.0 at the start, red line) , and the strategy learner portfolio (normalized to 1.0 at the start, blue line) over the in-sample period. (Transaction costs: Commission $0.00, Impact 0.000)

2.3. Discussion

The result of the strategy learner is dependent on randomness. To investigate how this randomness will affect the performance of the strategy learner, I performed in-sample trading simulation ten times using the strategy learner. While the Sharpe Ratio of the manual strategy kept constant at 0.731, the Sharpe Ratio from the strategy learner range from 0.746 to 2.047, with an average of 1.422. Thus, I would confidently expect the strategy will relatively overperforms the manual strategy.

3. 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). Conduct an experiment with JPM on the in-sample period to test that hypothesis. Provide charts, graphs or tables that illustrate the results of your experiment. The code that implements this experiment and generates the relevant charts and data should be submitted as experiment2.py. Be sure to add in an author method.

3.1. Experimental Methods

Hypothesis: increasing the value of impact will result in the decrease of the cumulative return.

In `experinment2.py`, I investigated how the impact will affect the cumulative return on the strategy learner and the manual strategy. The impact under various range were investigated. The resulted return from the strategy learner and the manual strategy under various impact are plotted in **Figure 2** and **Figure 3.**

3.2. Result

Figure 2 and **Figure 3** demonstrates the cumulative return of the strategy learner and the manual strategy portfolio under a range of impact. It is trivial to notice that the performance of both the strategy learner and the manual strategy monotonically decrease. Compared with the manual strategy, the strategy learner was more significantly affected by the impact, which is possibly due to that more trades were made by the strategy learner.

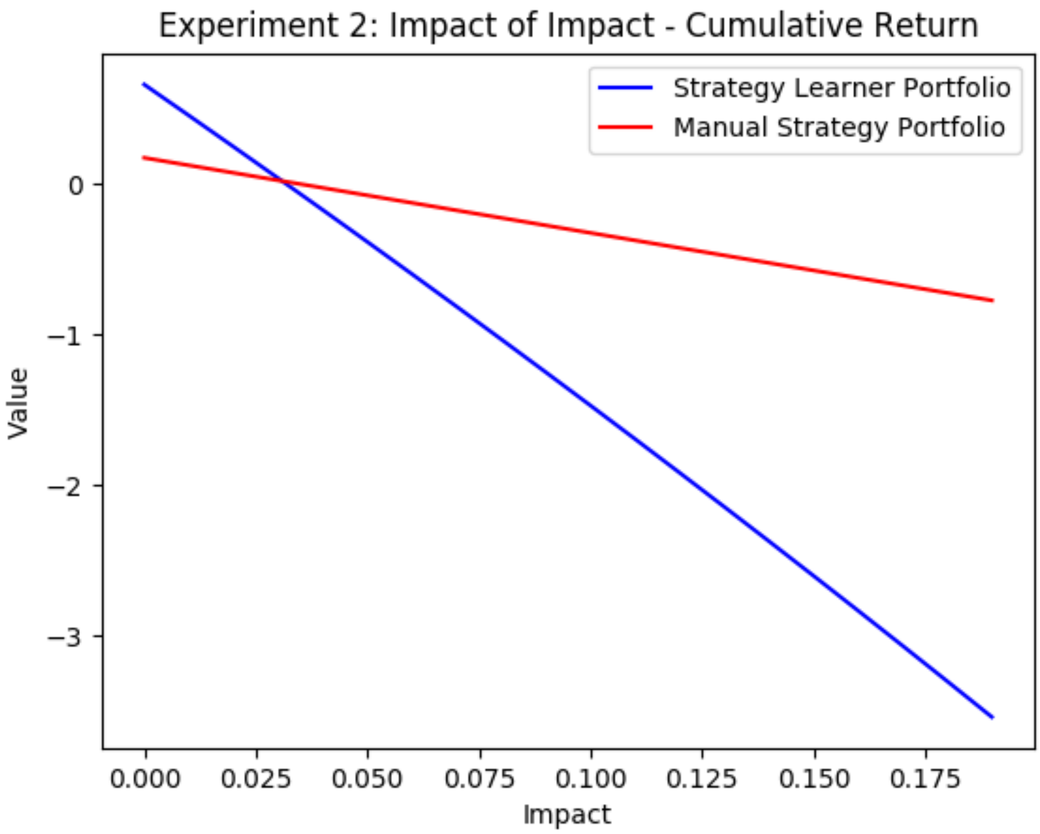

Figure 2: Trading impact vs. the cumulative return of the manual strategy portfolio and the strategy learner portfolio. Range of impact is 0.0 to 0.2.

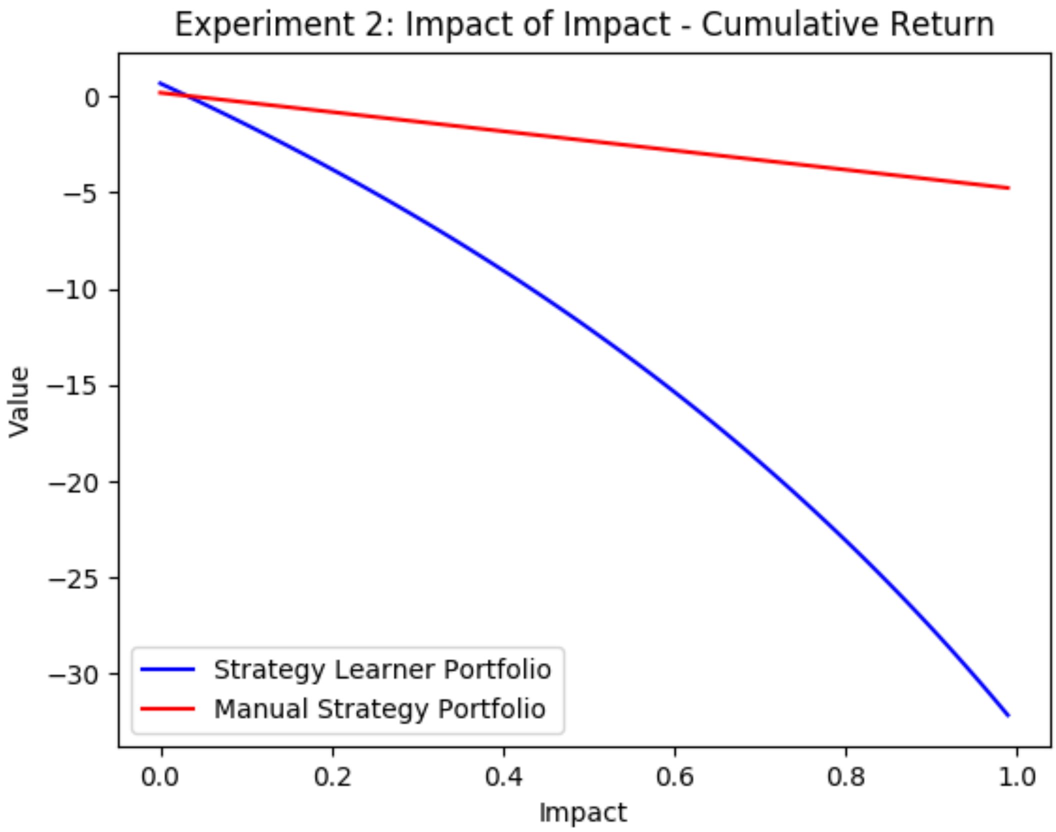

Figure 3: Trading impact vs. the cumulative return of the manual strategy portfolio and the strategy learner portfolio. Range of impact is 0.0 to 1.0.