

Zongran Luo

Zluo76

This project is to design a learning trading agent to make decisions of trading. My strategy learner is to use random decision tree to train a classification decision tree using indicators generated by prices of a symbol and decisions made by best policies. My strategy learner is 100% more benefit than the manual strategy and 2.5 times more than the benchmark.

Raw Data

I am using prices of a symbol between a range of time. Generally, I will use prices of a symbol for two years for training.

Indicators and Outcome

With the raw data, I can generate the input and output data. The inputs in this project are indicators generated by prices of a symbol. In the experiment 1, I will use three indicators, simple moving average, Bollinger Band and Stochastic Oscillator. These three indicators are normally used for predictions and making decisions of stock. In my project, these three indicators are calculated based on the 14 previous days and are used as variables for predictions of trading actions.

The outcome, which is holding position, we are using here is generated in the best situation. Under this situation, we can guarantee each position we made is 100% beneficial and no loss. I have three possible positions here, 1000, -1000 and 0. 1000 represents a long position, meaning I am currently holding 1000 shares in hand. -1000 represents a short position, meaning I am currently shorting 1000 shares. In other words, I owed 1000 shares by selling them out in advance. 0 is the position of cash, meaning I am currently out of the market and have no shares on hand. The transitions between positions result in trading. For example, if I make a short position, -1000, from a long position, 1000, I will sell 2000 shares, which are marked as -2000. In this case, any trade we made is positive and this is a good way to train a model. The market impact is considered since a high market impact may discourage the number of trading and an increasing stock is not necessarily profitable unless the stock can cover the cost of market impact. Since the market impact influences the best strategy, which decide the Y of training set, the strategy learner will be trained once the market impact is changed.

Strategy learner

So now, we have a set of data, three independent variables, simple moving average, Bollinger Band and Stochastic Oscillator, and a dependent variable, the holding positions. The strategy learner is based on a classification random decision tree. The model of random decision tree will randomly pick one of three features and the median will be picked to split the data set. The data set will be split into leaves until either the leaf has five rows of data or all dependent value, which is holding position here, in the leaf are the same. Since this is the classification decision tree, all the dependent values have three classifications, -1000 (SHORT), 0 (CASH) and 1000 (LONG), the values of leaves are determined by the

mode of these three classifications. In other words, the leaf will be marked by one classification which has the most in this leaf. Here is the pseudo code to train the model of decision tree:

```
def buildtree(X, Y):
    if X.size <= leaf_size:
        return [['leaf', mode(Y), ""]] # find the Y with the most as the leaf

    if Y has the same values:
        return [['leaf', mode(Y), ""]]
    else:
        split_feature = randomly pick one feature from X
        split_value = median (split_feature)
        lefttree = buildtree(X[X[split_feature]<= split_value], Y[X[split_feature]<= split_value])
        righttree = buildtree(X[X[split_feature]> split_value], Y[X[split_feature]> split_value])
        root = [split_feature, split_value, 1, len(lefttree)+1]
        return ([root]+lefttree+righttree)
```

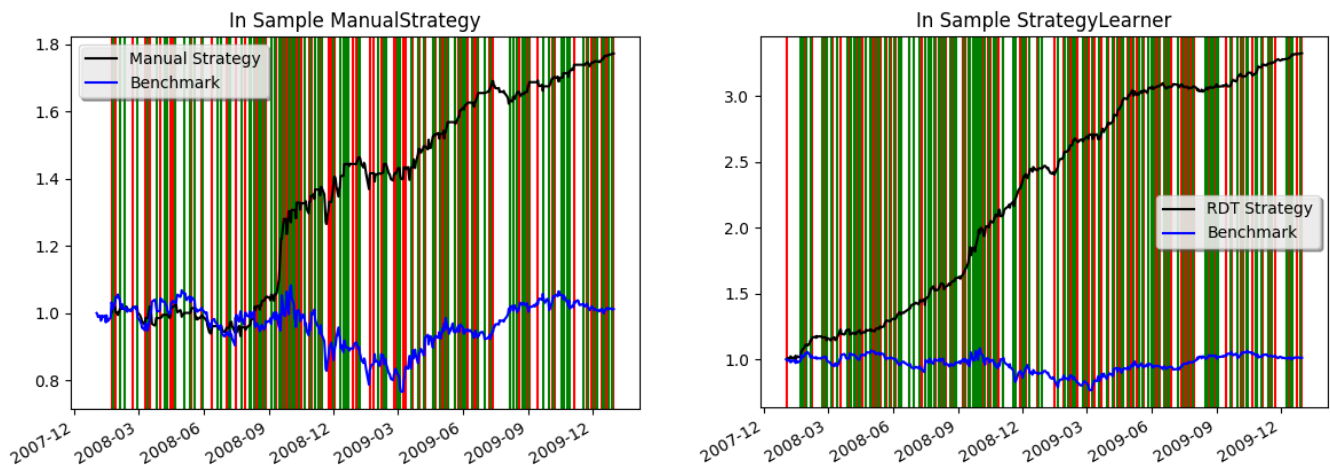
Experiment 1

In the first experiment, my goal is to test the strategy learner and see if it is better than the manual strategy. I implemented three strategies to see the differences.

- Best Possible Strategy: give out the best decisions at all time
- Manual Strategy: this strategy is to determine positions by taking the most ones made by indicators. For example, if two out of three indicators refer to a LONG position, the manual strategy will go for long.
- Strategy Learner: this is strategy is to use random decision tree model to learn from in-sample data created by Best Possible Strategy.

To make the experiment more stable, I did not add impacts and commissions. Also, I assume only three positions are available in this experiment, which are long, cash and short positions, and only one symbol JPM in this evaluation. The start value for each strategy is \$10000, and the bench mark started with 1000 in a long position until the last day with a cash position.

Here are the plots for two strategies. Y axis is the value of portfolio after being normalized. Green lines are the time to sell while the red lines are the time to buy. The best strategy has the most optimal decisions of trading and these decisions are the dependent variables of the training set for the strategy learner. The graphs below show the trending of the values of portfolio made by manual strategy and strategy learner.



The blue line here is the bench mark, and we can see both of manual strategy and strategy learner are better than the benchmark. However, we can see the strategy learner has a much better performance than the manual strategy. Strategy learner tribbles the started value while the manual strategy did not catch the double.

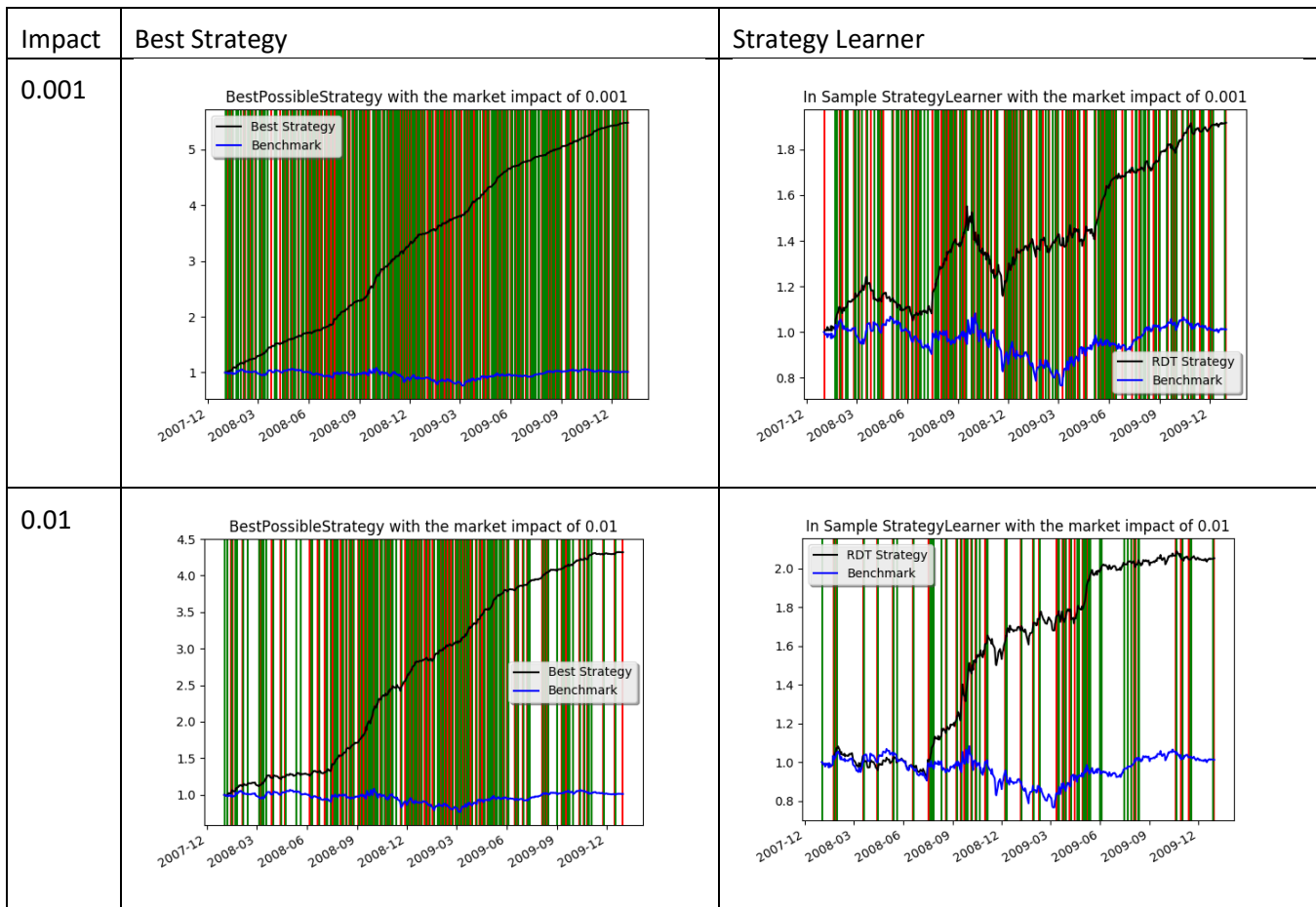
This experiment could be expected in any datasets, since the random decision tree model is to learn from the sample which follows the best strategy while the manual strategy is simply fowling the prices. If we simply look at the in-sample experiment, the strategy learner can always overshadow the manual strategy.

Experiment 2

This experiment will consider the market impact, which is the proportion that will be charged while doing the trades. The price will increase a percentage of impact while the stock is purchased while the price will decrease by a percentage of impact while the stock is sold.

To consider the influences of the market impacts, I will change the impact to see what happened. A high impact will increase the cost of each trade and so my hypothesis is that a higher impact will result in a lower number of trading.

I set the impacts as 0.001 and 0.01. The stock will increase by 0.1% and 1% separately while buying and decrease by the same percentages while selling.



For here, blue lines are the bench mark and the black lines are the profit. We can see with the impact increases, the lines of buying and selling, colored by red and green, become less dense. However, we can see the normalized values of portfolio becomes less than before. In the Best strategy, the value of the portfolio decreases from over 5 to less than 4.5, but we can see the value of portfolio of the strategy learner is relatively stable. The frequency of trading becomes less often but the normalized value of portfolio is about 2.0.

Conclusion

In these two experiments, we can find the value of portfolio is increasing by adding learner based on the indicators by prices, and a higher impacted result in a lower number of trading, but did not decrease the value of the portfolio dramatically.