Machine Learning for Trading – Project 8

Tri Nguyen
tnguyen497@gatech.edu

Framing the Trading Problem for Q-learner

In this project, Q-learner is used to solve the trading problem. The integral elements required by the Q-learner are the states, actions and rewards. In other to determine those elements, first, it is necessary to frame the trading problem in terms of a learner trying to maximize the rewards for its actions.

Thinking of trading as a Q-learning problem is almost natural. In reality, trading involves the trader taking actions (buy, sell, hold etc.), and receiving monetary rewards. In fact, given the constraints of the project, we define:

Q-learning actions: LONG (to a maximum of 1000 shares), SHORT (to a maximum of 1000 shares shorted), and DO NOTHING (maintain the current holding).

Q-learning rewards: The daily return (next trading day vs today) as a result of an action taken today.

The final piece of the q-learning puzzle is the states. While for a real trader, intuitively, his or her state would be the current holding. However, we also want to encompass the states of the stock movement itself in order to predict the future. Therefore:

Q-learning states: given by the indicators (to be described) of the stock on any given day.

Note that we have left out Holding as a component of the states. The reason being we can cover Holding as part of the environment in which the learner operates (for calculating the rewards), which leaves the states simpler to maintain and understand.

The indicators used to describe the states of the stock on any given day are:

Momentum

Momentum of a stock over n days is the return of a certain day compared to n days before, specifically:

Momentum (n day) = (today price/price n days ago) – 1

Price/simple moving average ratio

Price/SMA shows the prices of the stock relative to its mean value (simple moving average).

Bollinger Band Percentage

Bollinger Band Percentage = (price–lower band)/(upper band–lower band)

Notice that all three indicators are ratios rather than absolute values. This is to make sure that typical values for any stock will be similar, thus generalizing the learning algorithm.

To turn the indicators into a discretized state, first, we calculate each indicator for a given day. Let's call them I1, I2, and I3. We then separate the values of each indicator across all trading days being used into 10 bins. Therefore, each indicator for a date will be given an index value from 0 – 10, corresponding to a bin it belongs to.

The discretized state of a trading day is then given by:

I1*100 + I2*10 + I3

In other words, for three indicators each having 10 possible values, we have a total of 1000 possible states for the Q-learner. We now have all the elements required for the Q-learner.

Experiment 1

In this experiment, we compare the performance of the Strategy Learner (Q-learner) described above and the manual strategy developed in project 6.

The experiment and assumptions

Trading details: JPM, 1st Jan 2009 to 31 Dec 2009.

Allowed position: 1000 shares long, 1000 shares short, 0 shares.

Impact: 0.005

Starting value: $100,000

Q-learner parameters: learning rate alpha 0.2, discount rate gamma 0.9, random action rate rar 0.5, random action decay rate 0.99, dyna 0.

The outcome

The experiment's result is shown in figure 1 and table 1.

The result shows that the Strategy Learner greatly out-performs the Manual Strategy in terms of cumulative return. This can be expected for the in-sample period. The Manual Strategy was manually tuned to maximize returns, and were limited to only two thresholds for each indicator. Whereas the Strategy Learner was trained with 10 discrete thresholds for each indicator, meaning the Strategy Learner can detect more nuanced changes in the stock movement.

Another obvious advantage of the Strategy Learner is that once built, it can be used to trade any stock, whereas the Manual Strategy requires the parameters to be manually tuned for each new stock.

The granularity of the states in the Strategy Learner should not be pushed too far though, as it can result in over-fitting and cause a drop in the out-sample performance.

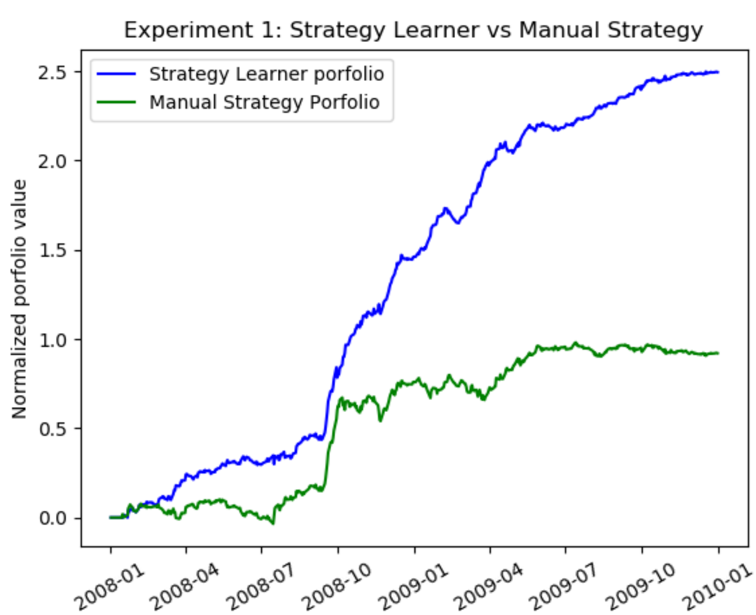

Figure 1: The performance of Strategy Learner (Q-learner) vs Manual Strategy

	Strategy Learner	Manual Strategy
Cumulative return	2.495	0.920
Std dev	0.008	0.010
Mean	0.0025	0.0014

Table 1: The performance of Strategy Learner (Q-learner) vs Manual Strategy

Experiment 2

Hypothesis

If the impact is increased from zero to a small amount, there will be little to no impact on the trading behavior.

If the impact is increased further, the trading behavior will be more notably impacted. The algorithm will trade a lot less frequently.

With non-zero impact, the performance will decrease as it is now costlier to trade and make a profit, and the learner now trades less often and therefore is less likely to take advantage of the changing in prices.

Concretely, as impact increases:

1. Number of trades decreases.
2. Accumulative return decreases.

The experiment

Trading details: JPM, 1st Jan 2009 to 31 Dec 2009.

Allowed position: 1000 shares long, 1000 shares short, 0 shares.

Impact: Three different impact values are tested **0.0, 0.0005, and 0.01**

Starting value: $100,000

Q-learner parameters: learning rate alpha 0.2, discount rate gamma 0.9, random action rate rar 0.5, random action decay rate 0.99, dyna 0.

The result

Figure 2 shows the learner performance with respect to three different impact values. The higher the impact, the poorer the performance as expected. Table 2 summarizes the performance in more details.

Interestingly, as impact increases, the learner trades slightly less often. This is due to the fact that there will be occasions where trading is less profitable due to the cost associated with trading, and therefore the learner would opt to do nothing instead.

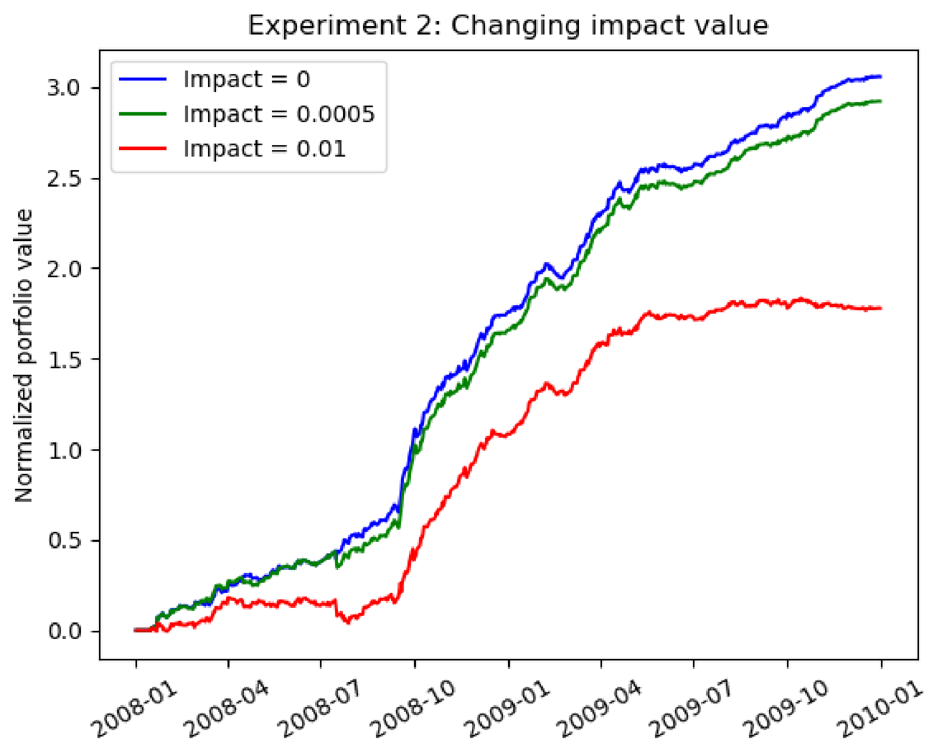

Figure 2: Strategy Learner performance with different impact values

Impact	0	0.0005	0.01
Number of trades	191	187	85
Cumulative return	3.056	2.92	1.78
Std dev	0.0078	0.008	0.01
Mean	0.0028	0.0027	0.0021

Table 2: Strategy Learner performance with different impact values