Strategy Learner (using QLearner)

1. **Indicators**
    a. Moving average convergence divergence (MACD)
    b. Relative Strength Index (RSI)
    c. Price/Simple Moving Average ratio

    To capture the values of the indicators from the beginning of the start period, I looked back number of days required so that the indicators are meaningful from the start date.
2. **Data steps (Discretization)**

    To specify a finite state space for the Q Learner, it is important to "discretize" the indicators such that we create buckets for each indicator. I created 5 bins for each indicator.

    Reason for choosing 5:
    - 5 bins capture sufficient variation and hence information of the indicator
    - A relatively smaller state space increases computational efficiency

    After creating these buckets, I combined discretized states of the 3 indicators into 1 value.
    i.e. $State = v1*5^0 + v2*5^1 + v3*5^2$, where v1, v2 and v3 are the values of the 3 indicators.
3. **Learner used – Q Learner**

    Parameters
    - Number of States = 5*5*5 = 125
    - Number of actions = 3 (Cash = 0, Buy = 1, Sell = 2)
    - Alpha = 0.2
    - Gamma = 0.9
    - Random action rate = 0.7
    - Random action decay rate = 0.999
4. **Steps for implementing Strategy Learner**

    For the policy learning part:

    Over the training period
    - Compute values of indicators
    - Discretization (explained above)
    - Instantiate a Q-learner
    - Get an initial action
    - Run a for loop over the training data
        - Action == 0 (cash), Action == 1 (buy or long position), Action ==2 (Sell or short position)
        - Update holdings, cash and portfolio value
        - Depending on the action taken, compute the reward (named step reward) for the last action
        - Query the learner with the current state and reward to get an action
        - Update the total reward by adding step reward

    Repeat the above loop multiple times (no. of epochs) until total reward stops increasing, or 200 (whichever happens sooner).

For the policy testing part:

Over the testing period

- Compute values of indicators
- Discretization
- Run a for loop over the testing data
    - Query the learner with the current state to get an action
    - Implement the action the learner returned and update portfolio value
- Return the resulting trades in a data frame

5. **Calculation of the Reward Value**

   I am calculating the reward whenever I take an action, i.e. a step reward. This is one of the approaches to calculate the reward. I chose this method over the cumulative reward method, as this method helps to converge the Q-Learner faster.

Experiment 1

Indicators used for Manual Strategy

- Moving average convergence divergence (MACD)
- Relative Strength Index (RSI)
- Bollinger bands

Indicators used for Strategy Learner

- Moving average convergence divergence (MACD)
- Relative Strength Index (RSI)
- Price/Simple Moving Average ratio

Parameters for all 3 cases (Manual, Strategy, Benchmark):

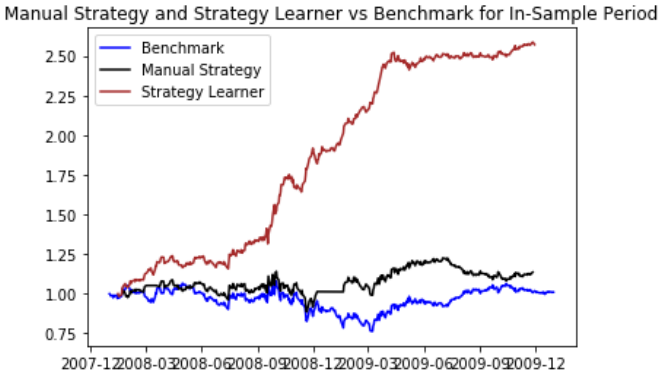
- Stock = JPM
- Impact = 0.005
- Commission = 0
- Start date = (2008,1,1)
- End date = (2009, 31, 12)

Assumptions

- We are not trading in real time and our decisions are based on historical prices. Hence, the real impact that would have occurred due to our buying or selling actions has not really been accounted for in the current scenario. Hence, the calculation of the market impact is an important assumption that we're making.
- We are assuming that there are no commission costs involved.
- We are assuming that our holdings can be either 0, 1000 or -1000.
- For calculating Sharpe ratio, the sampling frequency is 252 and risk-free rate is taken = 0.

Results of the experiment

The following chart shows the normalized portfolio values for the in-sample period for JPM for the Manual Strategy and Strategy Learner in comparison to the Benchmark. As we can see, the Strategy Learner outperforms the Manual Strategy and the Benchmark by a huge margin.



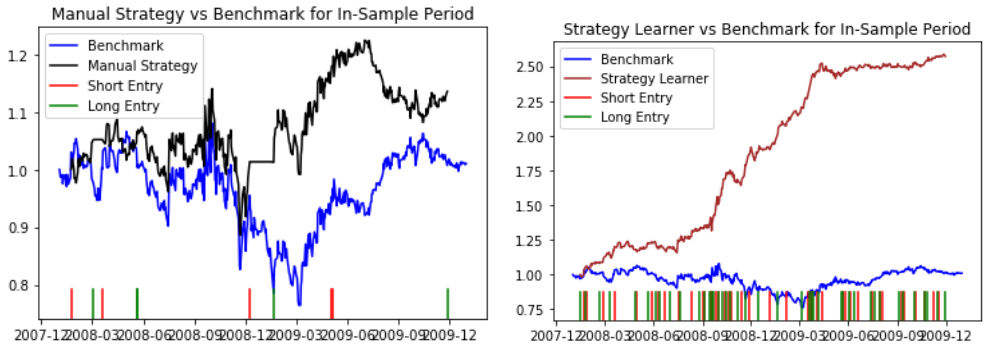
Portfolio Statistics

	Benchmark	Manual Strategy	Strategy Learner
Cumulative Return (Ratio)	0.010376	0.135805	1.571050
Average Daily Return	0.000165	0.000381	0.002053
Standard Deviation of Daily Returns	0.017040	0.014790	0.010116
Sharpe Ratio	0.153637	0.409668	3.222294

Note – The portfolio statistics for the Strategy Learner may vary every time the code is run, because of the random actions the Q Learner takes. Hence, the trades, and the portfolio values would be different every time (not a lot, but slight variations are expected.

The Strategy Learner outperforms the manual strategy by a huge margin, and we would expect this relative result every time with in-sample data.

Trades Frequency



As we can see, the Strategy Learner is making more number of trades than the Manual Strategy.

Experiment 2

Hypothesis

Buying or Selling shares can affect stock prices in the market, such that the change in price will act against us.

To incorporate the impact into the Q-Learner, I calculated the step reward in the following manner.

Step Reward = (% Change in Price - % Impact) *Holdings

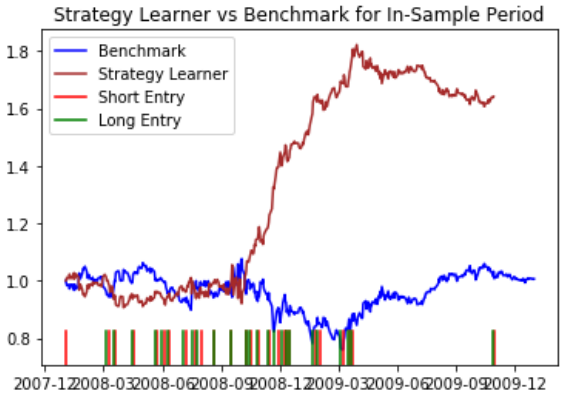
Hence, the reward reduces due to the impact. Higher the impact, lesser is the step reward.

My hypothesis is that if the market impact is higher, leading to lower reward values, the Q-Learner would make lesser number of trades, and also lead to a lower cumulative return.

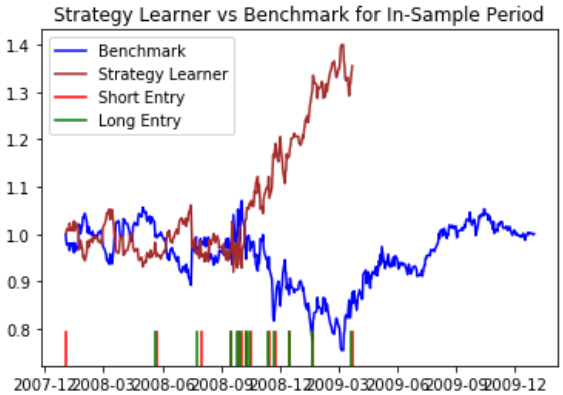
Experiment

The following 2 graphs show the Strategy learner performance for impact = 0.015 and impact = 0.03.

Impact = 0.015



Impact = 0.03



Results

Following are the metrics of the 2 scenarios.

<u>Market Impact = 0.015</u>

- Cumulative Return = 0.64160
- Sharpe Ratio = 1.32969
- Number of Trades = 51

<u>Market Impact = 0.030</u>

- Cumulative Return = 0.35518
- Sharpe Ratio = 1.05747
- Number of Trades = 25

Hence, the experiment results support the hypothesis.