Part I. Methodology

Steps, Indicators, Data Adjust

In this project, I develop a strategy learner based on the Qlearning approach, to help make informed decisions for trading stocks.

First, I process the data to be used for train the Q Learner. I used exactly same indicators as former Ma nual Strategy: **Bollinger Band Percentage, MFI, and Price/SMA ratio,** all of these 3 indicators are using 14 days window size.

Besides, I also **discretized** the indicator since the indicators data are continuous values, but qlearner need limited of discrete number as states to update Q tables/ tuple(state, reward). And also, by doing this, it will make the Q Learner converge more quickly. I discretize the 3 indicators each into 10 groups. I followed the procedures that professor mentioned in the lecture. Generate the 10 thresholds and divide the input indicators' table. After discretize the indicators import data frame. I calculate the states follow the equation:

psr*100+bbp*10+mfi

Thus the valid states are converted into 3 digit numbers.

In addition, I used **daily returns as rewards** to train the learner. As professor states, the daily return will make the learner converge more quickly than portval of the portfolio.

Second, I trained the learner based on Q Learning and develop a policy based on available market data of the stock via the indicators states and daily return rewards. I set the number of states to maximum 1000, to allow the qlearner to go over the training data, and keep record the total rewards as the loop going. To update the Q table, I set the strategy to be take 3 actions: hold--0, long--1, short--2. If the action says long, and current hold is 1000, then we do nothing. If it goes from short to long or vice versa, we double the holding shares action if current holdings are not positive or negative.

Once the total reward does not change any more and the iteration number is large than 30, which guarantee that there is no random successful learning, it means that the learner has converged, and it will stop to start another new process of looping over the train data. By the end of training, the addEvidence step will store a Qtable/Policy table to allow query in testPolicy steps.

Third, I tested the learner based on future data of the JPM stock. The test process is test the Policy that generated during the train steps. So, I only using the querysetstate() method in Q Learner to avoid updating Q tables. If the action we query from policy is 0, we hold the position. If it's equals 1, we buy 1000 shares, and sell 1000 shares if it's equals 2.

Part II. Experiment 1

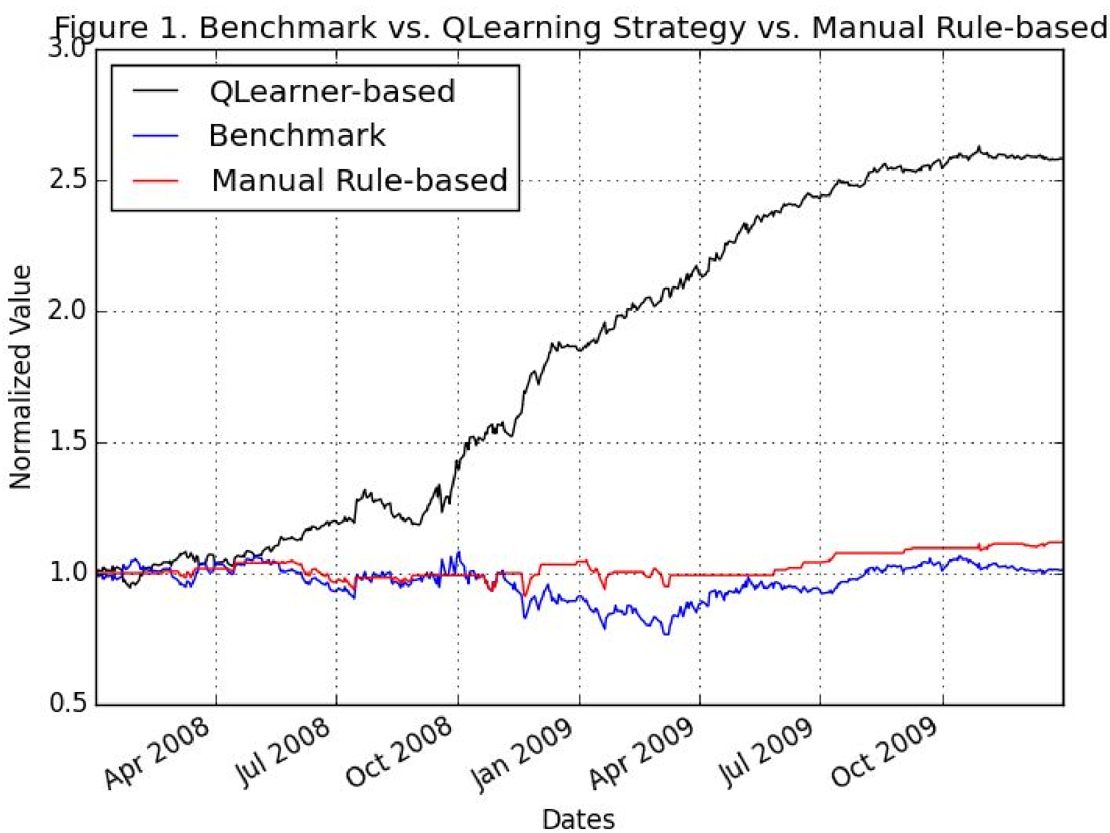
Experiment (Assumptions, parameter values and so on)

The experiment1 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.
- Strategy Learner: this is strategy is to use Q Learner model to learn from in-sample data.

To make the experiment more stable, I sets impacts=0.0 and commissions=0.0. 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 $100,000, and the benchmark started with 1000 in a long position until the last day with a cash position.

Outcome of Experiment



Cumulative Return of Manual Rule-based Strategy: 0.1178

Standard Deviation of daily return of Manual Rule-based Strategy: 0.00825365081753

Mean Daily Return of Manual Rule-based Strategy: 0.00025484099733

Portval of Manual Rule-based Strategy: 111780.0

Cumulative Return of Benchmark: 0.0123

Standard Deviation of daily return of Benchmark: 0.0170043662712

Mean Daily Return of Benchmark: 0.000168086978191

Portval of Benchmark: 101230.0

Cumulative Return of QLearner-based Strategy: 1.5833

Standard Deviation of daily return of QLearner-based Strategy: 0.0107394080155

Mean Daily Return of QLearner-based Strategy: 0.00194226390549

Portval of QLearner-based Strategy: 258330.0

From above chart and figure, it's clear that Q-learner based Strategy generate much higher return than manual strategy and the benchmark.

Expect this relative result every time with in-sample data

Yes, I think the Qlearner will do better than Manual Strategy and Benchmark, but less than Best Possible Strategy. Since Qlearner will model the actions to get most rewards by learning the train data and update the policy table.

Part III. Experiment 2

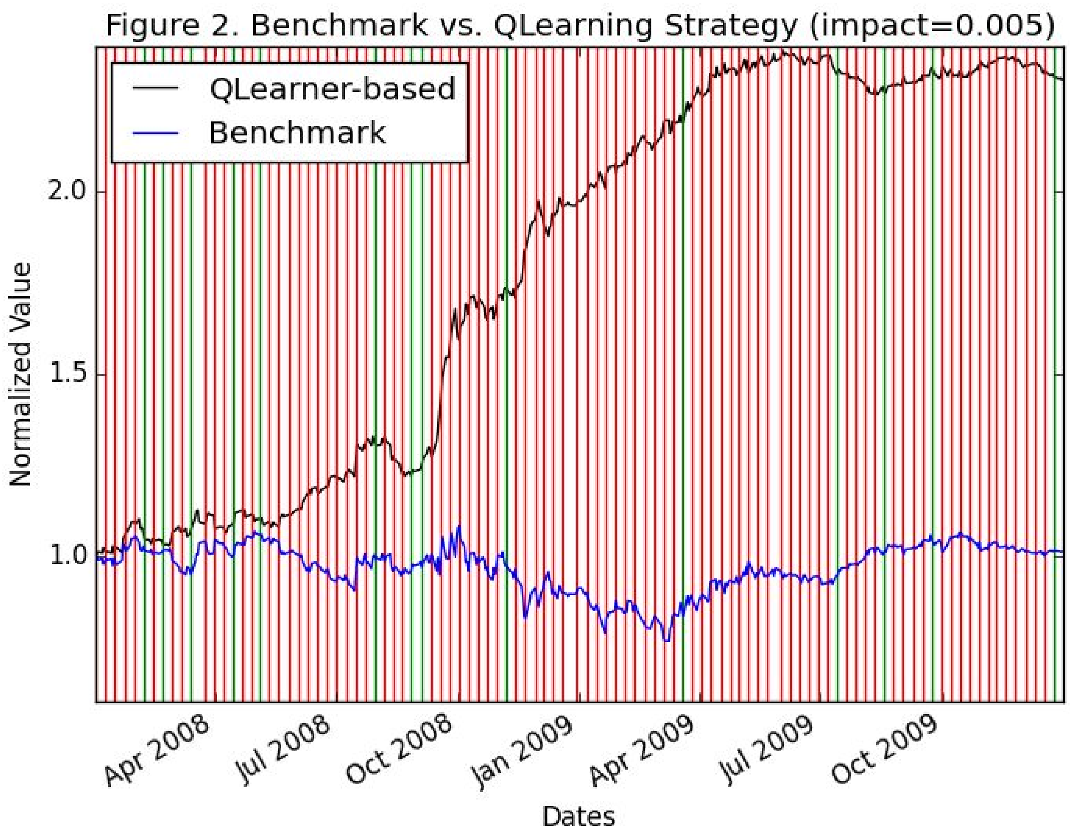
The experiment2 is to test the strategy learner and see if changing of parameters will influence the training result of Q learner strategy.

Hypothesis

My hypothesis is the influences from the market impacts will lower the influence of Q learner, which means the Q learner strategy will not generate impressive result or policy from learning the train data. since according to the free-market theory, it's little chance to beat the market with technical indicators. I use impact of 0.005 and 0.02 to test the influences.

Outcome of experiment

impact=0.005



Cumulative Return of Benchmark: 0.0123237046459

Standard Deviation of daily return of Benchmark: 0.0170394293305

Mean Daily Return of Benchmark: 0.000168726001316

Portval of Benchmark: 101037.65

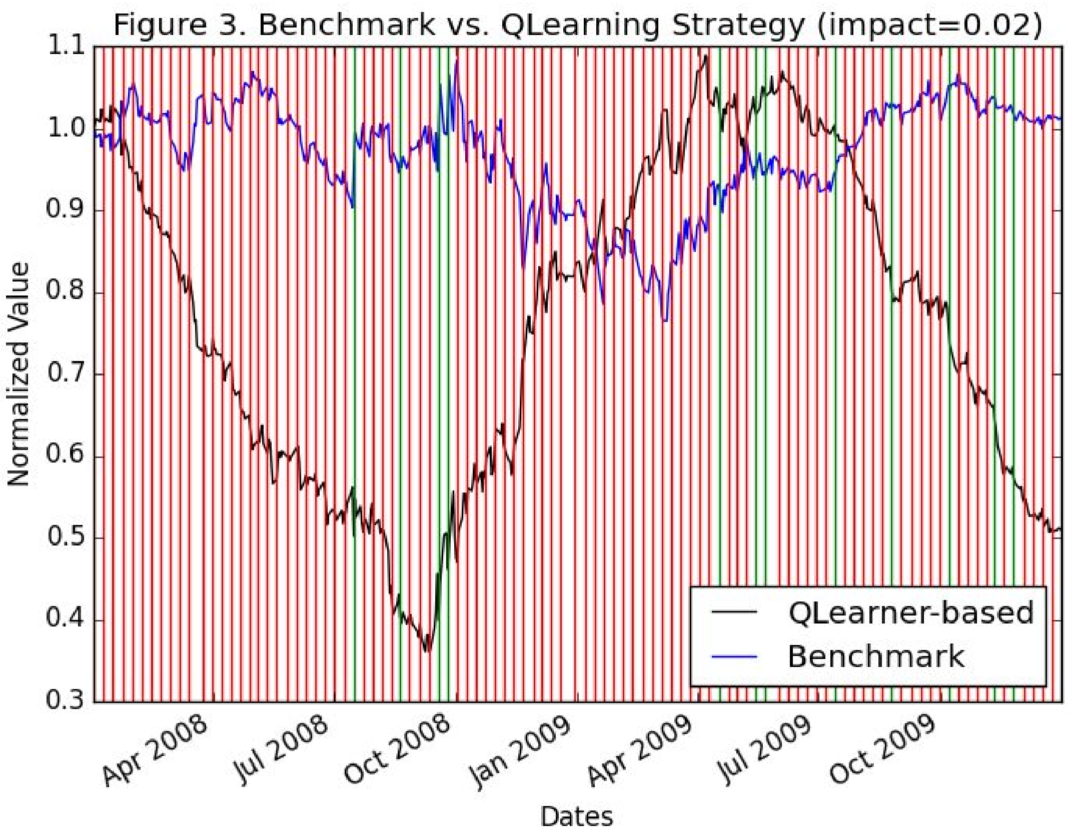
Cumulative Return of QLearner-based Strategy: 1.01489314697

Standard Deviation of daily return of QLearner-based Strategy: 0.0111699956894

Mean Daily Return of QLearner-based Strategy: 0.00145302937019

Portval of QLearner-based Strategy: 201101.75

impact=0.02



Cumulative Return of Benchmark: 0.0123953699766

Standard Deviation of daily return of Benchmark: 0.0171455005272

Mean Daily Return of Benchmark: 0.000170666429535

Portval of Benchmark: 100460.6

Cumulative Return of QLearner-based Strategy: -0.490518045845

Standard Deviation of daily return of QLearner-based Strategy: 0.0266598725604

Mean Daily Return of QLearner-based Strategy: -0.000983899845559

Portval of QLearner-based Strategy: 50556.2

Conclusion: From the above tests, it support my hypothesis that large market impact will lower the benefit from qlearner based strategy.