Nick Lee – nlee68

Strategy Learner Report

Describe the steps you took to frame the trading problem as a learning problem for your learner. What are your indicators?

In pursuing the Optimization Learner approach, I essentially built a knowledge-based system for stock trade strategy optimization. The optimizer is programmed to iterate through different thresholds and coefficients in a discretized hypothesis space to trigger technical trade decisions. The goal is to maximize portfolio return.

The trading strategy I implemented in Manual Strategy used three technical indicators to make decisions about trades and positions: Bollinger Bands, Price divided by 20-day Simple Moving Average, and a 3-day moving average of a 20-day Stochastic Oscillator.

In Manual Strategy Trader, I used these triggers to make buying and selling decisions. Each of the three technical indicators “voted” for a buy or sell action, and the average vote was used to decide the final action.

If current_stock_price > high Bollinger Band, BB issues a “Sell” vote.
If current_stock_price < low Bollinger Band, BB issues a “Buy” vote.
If 3-Day Moving Avg. of Stochastic Oscillator is < 20, Stochastic Oscillator issues a “Buy” vote.
If 3-Day Moving Avg. of Stochastic Oscillator is > 80, Stochastic Oscillator issues a “Sell” vote.
If Price divided by 20-day SMA is < 0.7, then P-over-SMA issues a “Sell” vote

Each vote had different weights attached to provide each indicator with more or less influence over the final decision. BB and SO both had a weight of 1 (equal weight), while I found that giving P-over-SMA a smaller weight of 0.5 yielded better results than giving it equal weight.

In Optimization Learner, I sought to find the optimal values for these variables:

- Best width of Bollinger Band to trigger buy/sell decisions
- Whether to use Stochastic Oscillator or not to make trade decisions
- Value of P-over-SMA to trigger a Sell vote
- Weight to give P-over-SMA in trade decisions

Our model performs well by optimizing the above four items (passes 4 of 4 ML4T tests provided by course instructor).

Did you adjust the data in any way (discretization, standardization)? Why or why not?

I did not adjust the stock data –I simply use the adjusted close to perform my calculations. My market simulator and indicator methods already perform necessary normalizations to calculate portfolio values and such, so my learner does not need to worry about normalization itself.

Nick Lee – nlee68

I discretized the hypothesis space because it would have been computationally impractical to find a global maximum with all variables within the 25-second time limit we had for training across an infinite space. For Bollinger Band width, I test widths of 75%, 100%, and 125% standard widths. For Stochastic Oscillator, I test 0 or 1 to determine the weight of the vote (0 means ignore, 1 means give normal vote). For P-over-SMA, I test values of 0.7, 0.85, and 1.0. For voting weight of P-over-SMA, I try 0 (ignore) and 0.5 (default value). I could have iterated through more values, but this would increase the training time significantly. For each addEvidence call, my program iterates through 36 permutations in about 15 seconds.

Experiment 1: *Using exactly the same indicators that you used in manual_strategy, compare your manual strategy with your learning strategy in sample. Plot the performance of both strategies in sample along with the benchmark. Trade only the symbol JPM for this evaluation. The code that implements this experiment and generates the relevant charts and data should be submitted as experiment1.py*

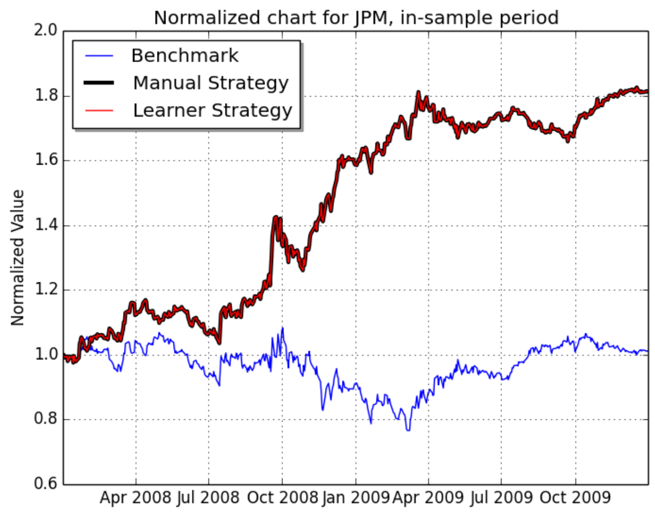
- **Describe your experiment in detail: Assumptions, parameter values and so on.**

Description of Experiment: Run Manual Strategy and Strategy Learner and compare the cumulative return (as well as other technical measures like Sharpe Ratio and Average Daily Returns) at the end of trading. I expect that Strategy Learner will match or exceed the performance of Manual Strategy every time because we initialize Strategy Learner with the same default values used in Manual Strategy. But Strategy Learner's advantage is that it can iterate through other weights and coefficients for technical indicators, potentially allowing it to find a superior trading strategy to the static Manual Strategy. We will never underperform Manual Strategy because we always use Manual Strategy's default values as an "initial guess." If it turns out Manual Strategy is the best strategy, we will default to the values that emulate that strategy.

- **Describe the outcome of your experiment.**

For Experiment 1, our Strategy Learner discovered that the Manual Strategy is the ideal strategy in its discretized hypothesis space. The results are identical to Manual Strategy because the parameters are the same.

Nick Lee – nlee68



The Manual Strategy and Learner Strategy curves are identical. One line is thickened to make it easier to see.

- **Would you expect this relative result every time with in-sample data? Explain why or why not.**

Yes, I would expect this relative result every time with in-sample data unless we adjusted other parameters (e.g. commission and impact) to make the environment different. Strategy Learner can adapt to its environment by tuning parameters, while Manual Strategy is fixed in place. Since Strategy Learner iterates through multiple strategies (including the same strategy used by Manual Strategy), it will always match or outperform Manual Strategy. And since Manual Strategy is tuned to perform well for the in-sample period, I expect Manual Strategy and Strategy Learner to perform at the same level.

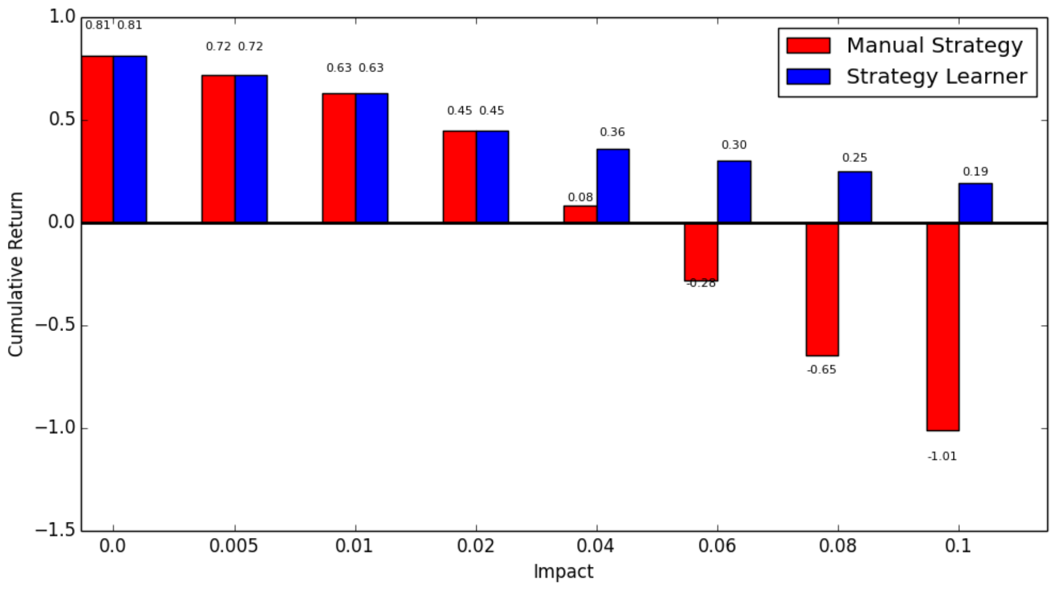
Experiment 2: *Provide an hypothesis regarding how changing the value of impact should affect in sample trading behavior and results (provide at least two metrics). Conduct an experiment with JPM on the in sample period to test that hypothesis. Provide charts, graphs or tables that illustrate the results of your experiment. The code that implements this experiment and generates the relevant charts and data should be submitted as experiment2.py*

Since our Strategy Learner is using a brute force optimization to find the best balance of technical indicators to make buy and sell decisions, I do not expect the buy and sell decisions to be affected by impact –but the outcome of those decisions will be affected, so the optimal strategy will differ from Manual Strategy.

Optimization Learner is more resilient to impact changes because it can iterate and find the best strategy for a new environment. I expect that Manual Strategy performance will drop faster than Optimization Learner. As impact increases, the profit from wise decisions will narrow and the losses from unwise decisions will widen. Cumulative Return and Sharpe Ratio will decrease. If we keep increasing Impact to unrealistic levels (e.g. 5%+ impact, which is never seen in real life), we should eventually see that Manual Strategy underperform Strategy Learner, and if we keep pushing impact out to ridiculous levels we may see both strategies underperform the benchmark.

Experiment

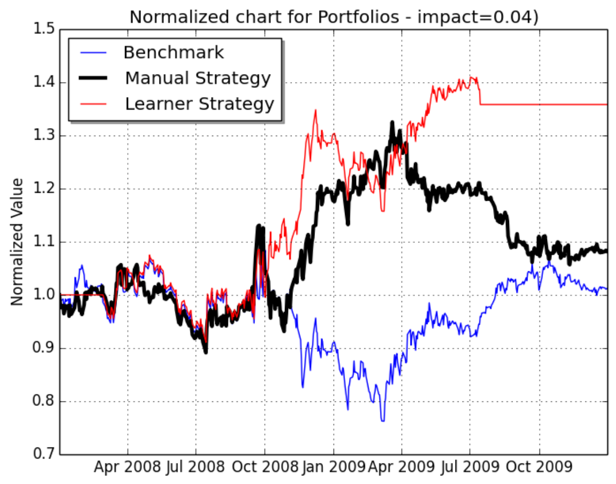
We can see that Manual Strategy and Strategy Learner performances decline as we increase impact. But after a while, the inflexible Manual Strategy collapses while Strategy Learner has the luxury of using other parameter values to minimize damage. You can see that Strategy Learner can still return a profit at unrealistically high impact levels.



We see that Strategy Learner outperforms Manual Strategy somewhere between 0.02 and 0.04 impact. I did not push the experiment beyond 0.1 impact because we would most likely never see impact that high in real-life trading.

Nick Lee – nlee68

Here is the portfolio value plot at 0.04 impact (Symbol JPM):



The chart below shows Cumulative Return (CR) and Sharpe Ratio (SR) for each model at different impacts. Highlighted squares show experiments where Manual and Learner strategies were identical. You can see that Sharpe and CR decrease as we increase impact, but Manual Strategy dips into negative returns and negative Sharpe Ratios while Strategy Learner remains positive.

	<- CR / SR	**CR / SR ->**	
Manual Strategy – Impact 0	0.8119 / 1.6848	0.8119 / 1.6848	Learner Strategy – Impact 0
Manual Strategy – Impact 0.005	0.7207 / 1.5290	0.7207 / 1.5290	Learner Strategy – Impact 0.005
Manual Strategy – Impact 0.01	0.6295 / 1.3653	0.6295 / 1.3653	Learner Strategy – Impact 0.01
Manual Strategy – Impact 0.02	0.447 / 1.0186	0.447 / 1.0186	Learner Strategy – Impact 0.02
Manual Strategy – Impact 0.04	0.0821 / 0.2858	0.3581 / 0.8306	Learner Strategy – Impact 0.04
Manual Strategy – Impact 0.06	-0.2827 / -0.4215	0.3029 / 0.7234	Learner Strategy – Impact 0.06
Manual Strategy – Impact 0.08	-0.6475 / -1.0056	0.2477 / 0.6128	Learner Strategy – Impact 0.08
Manual Strategy – Impact 0.10	-1.0124 / -0.7593	0.1925 / 0.5011	Learner Strategy – Impact 0.10

Wordcount: 1388