Strategy Learner Project Report

Author: Yu Bai
Date: 12/03/2017

1: Strategy Learner construction & training

The same 5 technique indicators used in the Manual Strategy are also considered for the Strategy Learner: price/simple moving average ratio (P2SMA), percentage Bollinger® band (%BB), relative strength index (RSI), momentum, and relative standard deviation of the moving average (i.e. normalized to the moving average). Same as in the Manual Strategy the lookback window is 20 for SMA and 40 for momentum. The first 3 help judge if a stock is oversold or undersold, the latter 2 help filter out random fluctuation from true trading opportunities.

The Strategy Learner is Q-learning based. It models the stock trading problem as a Markov Decision Process as follows:

States: a state is determined by the values of the 5 indicators and the holding status. Each indicator is discretized according to a set of thresholds with right boundary exclusion.

$P2SMA\ thresholds = \{0.9, 1.1\}$:	$P2SMA < 0.9$, $0.9 \leq P2SMA < 1.1$, and $P2SMA \geq 1.1$ map to 0, 1, 2.
$\%BB\ thresholds = \{0, 1\}$:	$\%BB < 0$, $0 \leq \%BB < 1$, and $\%BB \geq 1$ map to 0, 1, 2.
$RSI\ thresholds = \{35, 60\}$:	$RSI < 35$, $35 \leq RSI < 60$, and $RSI \geq 60$ map to 0, 1, 2.
$momentum\ thresholds = \{0.2\}$:	$\|momentum\| < 0.2$, $\|momentum\| \geq 0.2$, map to 0, 1.
$rel.stdev\ of\ SMA\ = \{0.07\}$:	< 0.07, ≥ 0.07, map to 0, 1.
$holding\ status$:	$LONG, SHORT, CASH$ map to 0, 1, 2

The concatenation of the 6 discretized parameters (one digit per parameter) encodes a state. E.g. 010101 represents a state with $P2SMA < 0.9$, $0 \leq \%BB < 1$, $|momentum| \geq 0.2$, $rel.stdev\ of\ SMA < 0.07$ and holding LONG. In total, there are 3 x 3 x 3 x 2 x 2 x 3 = 324 states. Note that the indicators are not standardized because each is already after a certain normalization and unit-less. In addition, we do not perform any comparisons across indicators so there is no bias induced by the different scale of the indicators. Moreover, the thresholds based on the current indicator values are meaningful and can be applied to different trading periods. Rescaling/standardization may instead harm the model generalizability.

Actions: LONG, SHORT and CASH. It means the corresponding actions to achieve the 3 holding positions.

Rewards: for Q-learning, we only need explicitly consider the immediate reward r, which can be defined as the portfolio change from time $t-1$ to t with action in $t-1$ executed. It's calculated as

$$r = \frac{price[t]*holding[t-1]+cash[t-1]}{price[t-1]*holding[t-2]+cash[t-2]} - 1$$

In the above equation, the nominator is the portfolio at day t after executing the action in $t-1$, and the denominator is that at time $t-1$. The $holding$ and $cash$ have incorporated the consequence of the actions and the parameter $impact$.

Q-table: In Q-learning we don't explicitly compute Transition function. Instead, we optimize the policy, aka Q-table directly. The 324 States and 3 actions lead to a 324 x 3 Q-table. The Q-table is updated with each experience tuple(s, a, s', r) where s is the state encoded by the indicators and holding status at t, s' is the state at $t+1$, a is the action taken at t, and r is the immediate reward associated with the transition from s to s' via a.

The StrategyLearner calls the QLearner implemented in the previous project. Worthwhile to mention that we set the initial random rate $rar = 0.95$ and its decay rate $radr = 0.999$. The big initial random rate with a slow decay helps to better explore the optimal solution over multiple epochs, given we have a rather complex model of 342 states. We also choose $gamma = 0.5$, with which the influence of each day will not go beyond ~5-7 neighbor days. This $gamma$ is reasonable considering the rather transient behavior of stocks. The model doesn't converge efficiently for higher $gamma$ values. Learning rate alpha is set the typical value of 0.2.

2. Experiment 1

We trained the StrategyLearner using JPM data from 2008-1-1 to 2009-12-31. We applied the same learner parameters as described in the previous section. The training stops when the percentage change of the cumulative return <=0.1%. We used impact=0 (and commission=0) for this experiment. Once trained, the learner was tested over the in-sample period. The results are summarized in Figure 1 and Table 1.

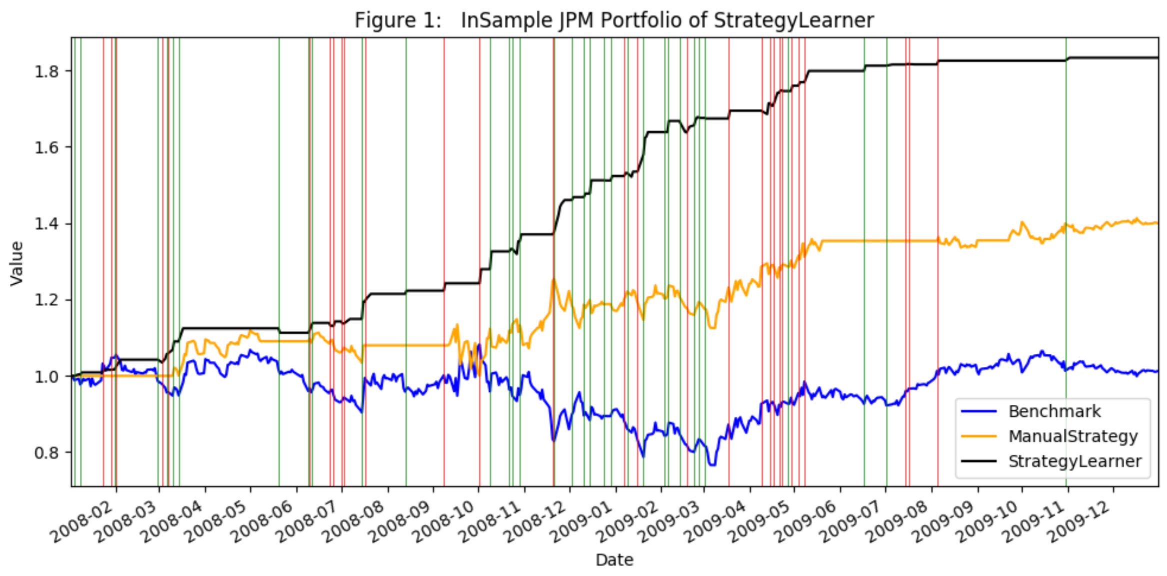
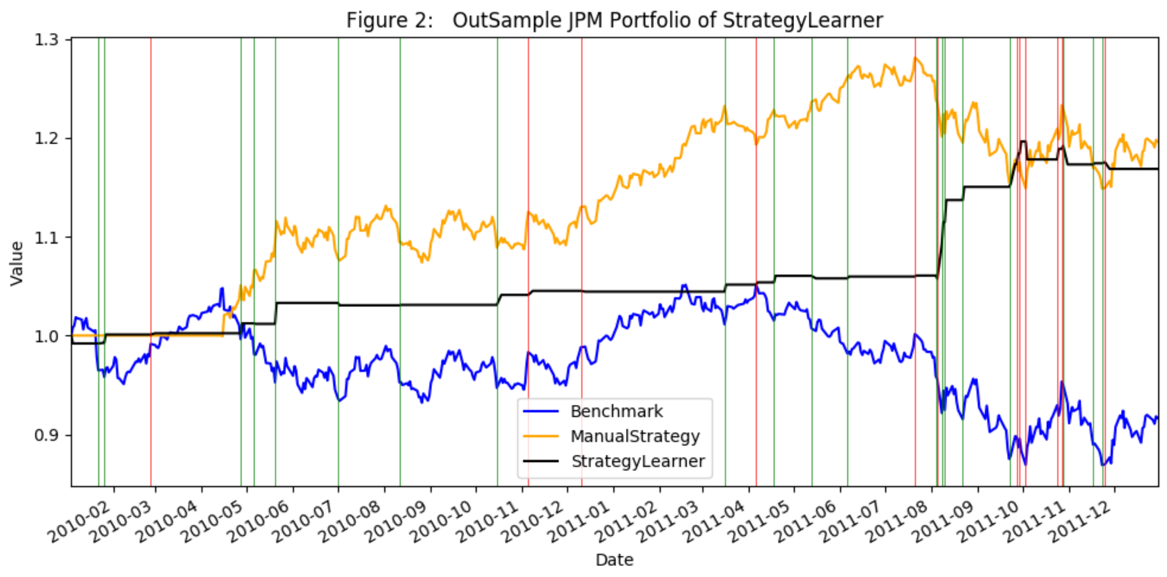


Table 1:

Performance	In-Sample			Out-Sample		
	Benchmark	Manual Strategy	**StrategyLearner**	Benchmark	Manual Strategy	**StrategyLearner**
Final portfolio	101230	140010	**183350**	91660	119630	**116860**
Cumulative return	0.0123	0.4001	**0.8335**	-0.0834	0.1963	**0.1686**
Mean of daily returns	0.000168087	0.000746991	**0.001216477**	-0.000137203	0.000378262	**0.000313724**
Stdev of daily returns	0.017004366	0.012611059	**0.005127862**	0.008481007	0.006623197	**0.002817872**
Sharpe ratio	0.156918406	0.940295401	**3.765890669**	-0.256812961	0.906620586	**1.767366085**
num. of trades	1	11	**114**	1	4	**62**

Figure 1 compares the portfolios from the benchmark, the Manual Strategy and the StrategyLearner over the in-sample period. One can see that the StrategyLearner outcompetes the benchmark and the Manual Strategy significantly during the in-sample period. It reaches a cumulative return of 0.83 whereas the benchmark cumulative return is 0.0123 and the Manual Strategy is 0.4 (see also Table 1). Besides having the highest return, the StrategyLearner portfolio has the lowest volatility, ~half of those in the Benchmark and Manual Strategy (Table 1, stdev of daily returns). As a result, it achieves the best Sharpe ratio of 3.76 among the three. The trading executions performed by the StrategyLearner are illustrated in Figure 1 (green vertical lines for LONG, red vertical lines for SHORT). Most LONG trades occur when the price dips and the SHORT ones take place when the price spikes, confirming that after training the StrategyLearner grasps the trading opportunities correctly. Interestingly, StrategyLearners trades much more frequently than the Manual Strategy (114 vs 11, Table 1). One reason is that zero market impact (and zero commission by default) is applied in experiment 1, so the optimal strategy should be to trade as frequent as possible once an opportunity is detected. Second, given the large degree of freedom in the model (324 states), the learner can come up with more complex decision rules to capture more trading occasions.

We expect that the better performance of StrategyLearner over the Manual Strategy over the in-sample data should hold for different stock and different training period, because it is designed to optimize the return and it explores the solution space much more efficiently than a manual try-and-error. The StrategyLearner is also more complex than the Manual Strategy (324 states vs. essentially 3 states), which typically renders better in-sample results. The tradeoff may be the loss of generalizability in the StrategyLearner, which leads to our additional test using the out-sample data.

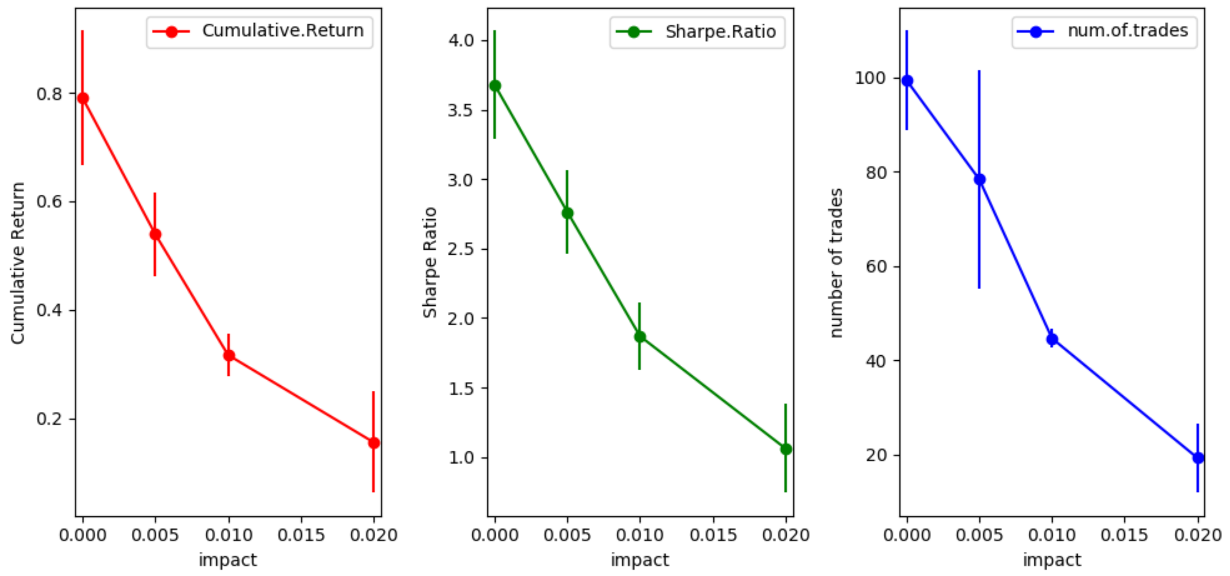


The out-sample analysis is not required per the instruction but is performed out of curiosity. During the out-sample period (Figure 2), the StrategyLearner still beats the benchmark by all performance measures (Table 1). The LONG (green vertical lines) and SHORT (red vertical lines) operations occur at the appropriate time in general. Considering the risk normalized return, the StrategyLearner wins over the Manual Strategy with a better Sharpe Ratio (1.76 vs. 0.907), mostly because of the low volatility (Table 1). However, its return (0.1686) is not as high as what the Manual Strategy offers (0.1963). Overall the performance of the StrategyLearner in the out-sample is OK but not as impressive as in the in-sample. It is possible that the StrategyLearner indeed suffers a certain degree of overfitting because of its high complexity.

3. Experiment 2

In this experiment, we focus on the in-sample JPM data from 2008-1-1 to 2009-12-31. The goal is to alter the $impact$ factor to see how it affects the learner performance. We applied the same learner parameters as described in the model building section. The convergence is defined as the percentage change of the cumulative return <=0.1%. We varied impact from 0, 0.005, 0.01, 0.02 (commission=0 by default) for this experiment. The learner was trained & tested over the in-sample period. For a better quantification, we repeated the experiment 3 times and summarized the results in Figure 3.

Figure 3: Performance of StrategyLearner vs. impact



One hypothesis is that with increasing the market *impact* factor, the return from the learner should decrease. It is not only because each trade will cost more cash against the market impact, but also a large impact may overwhelm the signal of the reward (typically ~0.001, see Table 1 In-Sample daily return of StrategyLearner). That is, the learner receives a reward predominated by the market impact instead of the daily return. In this case, learning if an action is a correct or wrong becomes more difficult. Another hypothesis is that there should be fewer trades when the impact goes up because the trade will be punished by the impact and hence discouraged.

In Figure 3, the dots represent the mean cumulative return (red), Sharpe ratio (green) and the number of trades (blue) over 3 repeats, with their standard deviations shown as the error bars. The cumulative return (from ~0.8 to ~0.15) and the Sharpe ratio (from ~3.7 to ~1.0) indeed decrease with respect to the increase of the *impact* value. In addition, the number of trades drops from ~100 to ~20 when the impact increases from 0 to 0.02. These observations confirm the above hypotheses.