Strategy Learner

Karel Klein Cardeña

Constructing Learner

In order to inform the trading strategy, three indicators are used to describe the current state of the stock. The first indicator is the simple moving average (SMA), which uses a lookback period to calculate the arithmetic mean of the stock over the last n periods. While the price of the stock fluctuates stochastically, the SMA erodes noise and allows the overall trend to be visualized. Prices that deviate sufficiently from the SMA will tend to revert to the mean, allowing it to be used as a signal to enter a profitable trade. To capture this deviation, we can use the ratio of price to SMA, where a ratio above one indicates a an excursion of the price above the SMA, and below one indicates price level below the SMA.

The second indicator used is the Bollinger Band %, a formalized indicator constructed by subtracting the lower BB band from the price, and dividing that quantity by the difference between the upper and lower bands. This BB% is thus less than zero when the price falls below the lower band and is greater than 1 when the price rises above the upper band.

Lastly, the normalized price ratio between SPY and the stock is used to indicate the relative difference between the stock in question and the market in general. The trends dictated by the market often act as a powerful attractor around which individual stocks fall subordinate and follow suit. Hence, a high ratio may indicate that the stock is lagging behind the market, signaling a potential rally.

To create a suitable training set for the learner, each indicator is calculated for each day in the sample period. The indicators are then used as features such that the training data is made up of 3 columns (1 per feature), and n rows, where each row represents one trading day. Each instance of the training set therefore represents a single day of trading with 3 features: SMA, BB%, and Ratio for that day. With the training data in order, we can bring focus to the target classes. The target class should embody the profitability of these indicator levels. To meet this criteria, the cumulative return is calculated by looking N periods into the future for each day. If the return is above the threshold YBUY, the instance is assigned a value of 1. Similarly, a return below the threshold YSELL is assigned a value of -1, and a value of 0 if it meets neither criteria. The set of classes {1,0,-1} therefore constitutes a discretization of the supervisory signal, and is suitable for training a classification learner.

Random forests are selected as the learner of choice. A key problem is that the random tree algorithm used previously was constructed as a regression learner. In order to use it to output trading decisions, however, modifications were made to transform it into a classification algorithm. To make this change, we modify how the tree is built by changing the leaf aggregation step. Instead of returning the average of the y values once the number of instances is below the target leaf size, we elect to return the mode. This change allows for a single class representing the most frequent class to be represented by the leaf of the tree. Since we are using random forests, the bag learner is also modified so that for each instance, a query returns the class that is the mode across all trees.

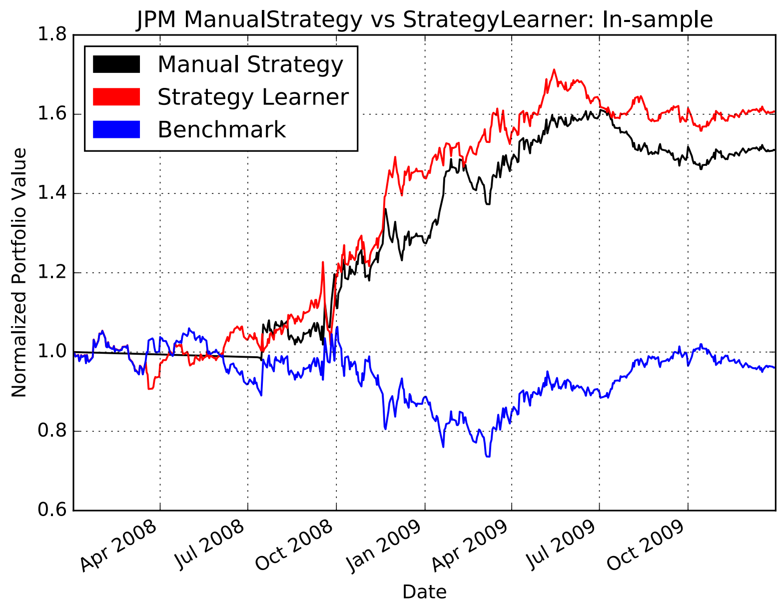
Using the aforementioned training set, we can now train the random forest classifier. Once the model is built, it can be queried with a 3-featured instances. The classifier returns a list of 1's, -1's, and 0's, indicating buy and sell signals according to the trading day's features. The strategy learner takes this list and formalizes a dataframe of trades by translating a 1 into a buy of 1000 shares (or 2000 shares if the prior position was -1000), a -1 into a sell of 1000 shares (similarly, -2000 shares if the strategy was long), and a 0 into a buy of 0 shares.

Experiment 1

In this experiment, we compare the in-sample performance of the newly made strategy learner with the manual strategy trained by a mere mortal. In the manual strategy, indicators are manually assigned a buy and sell threshold, which signal the strategy to take on a long or short position if all three indicators are above or below their respective thresholds. A lookback period of 50 days is used to try to describe long-term trends in the stock price.

For the strategy learner, we use a set of parameters made up of the lookback period, the YBUY and YSELL thresholds, a look-forward period, the leaf size, and the number of trees in the forest. To remain consistent, the lookback period is set to the 50 days used in the manual strategy, while a 30-day look-forward period is used to calculate cumulative return for each day. The YBUY and YSELL values discussed earlier are set to 0.001 and -0.001, respectively. This means that a 30-day return above 0.001 + impact will yield a signal to buy, denoted as a 1 in the training set. A leaf size of 10 is used in order to prune the tree and prevent overfitting on the training set.

With these parameters, the manual strategy and strategy learner are trained with JPM stock data over the period from January 1, 2008 to December 31, 2009. A benchmark portfolio is also added, which simply buys and holds 1000 shares of JPM bought at the start of the period and sells them on the last day. Each portfolio's results are summarized in the below chart.



In-sample Performance

	Manual Strategy	Strategy Learner	Benchmark
Cumulative Return	0.508390	0.605309	-0.037925
Stdev of Daily Returns	0.011840	0.012760	0.017468
Mean of Daily Returns	0.000886	0.001021	0.000075

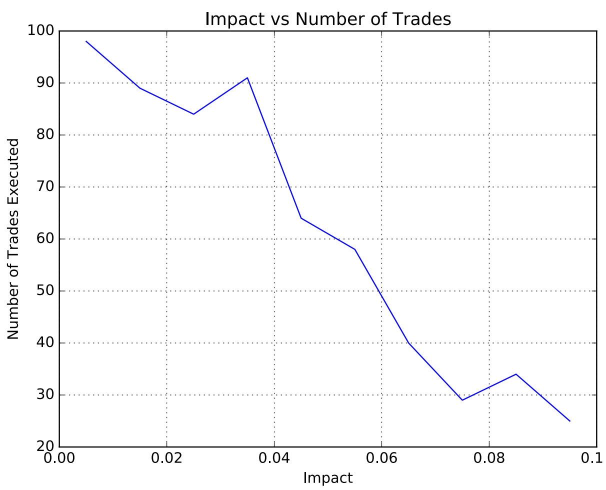
Despite the outstanding performance of the manual strategy over the benchmark, the machine learner still manages to outperform the manual strategy by nearly 10% in cumulative return, representing a 20% increase in performance. It follows that the strategy learner earns a 0.00115 increase in mean daily return over the manual strategy, albeit at the cost of a 0.00920 increase in standard deviation. With this kind of performance, the strategy learner is difficult to rival with any manually trained strategy.

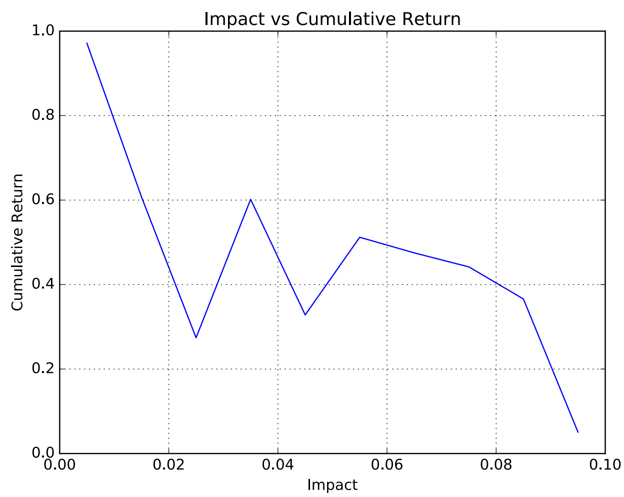
However, we can expect the relative performance of the learner to fluctuate over multiple runs. This can be explained by the randomness introduced by the random tree. When selecting features to split upon, the random tree algorithm may select the feature that is least indicative of future performance, and may repeat this selection over multiple splits. This will result in a tree that fails to predict future performance as well, and could underperform compared to its manual counterpart. While the bagging aspect of the random forest should help alleviate this deficiency, the phenomenon is still liable to occur as long as randomness is used to select features to split upon.

Experiment 2

In this experiment, the value of market impact is varied in order to observe its effect on in-sample trading behavior. To measure behavior, we use two metrics: number of trades executed, and cumulative return. The larger the impact each trade makes, the more the price of the stock will move against the position the strategy is taking on. We can therefore expect that trade to lose profitability as its impact increases. The strategy learner would thus make less trades in order to counter the effect of more costly trade executions. Correspondingly, we can expect the cumulative return of the strategy to decrease with increasing market impact.

To run the experiment, we test impacts from 0.005 to 0.095 with a step size of 0.01, running one strategy learner per impact and collecting the number of trades executed as well as the cumulative return of each strategy. The results of this experiment are summarized in the two charts below.





The results validate the working hypothesis. When impact is at its lowest point of 0.005, the strategy can afford to make many trades without impacting the profitability of its decisions. As impact increases, the stock price moves against the position regardless of the outcome of the trade, making the strategy's profitability increasingly dependent on the number of trades executed. A higher impact means that fewer trades will have 30-day cumulative returns meeting the YBUY and YSELL thresholds, thus yielding less buy and sell signals over the in-sample period. With higher constraints on the trading signals, the strategy learner will be prompted to make less trades.

In the chart describing cumulative return, a similar trend is observed, albeit with more volatility. Higher impact on the market reduces the return of each trade. If the strategy continued making the same number of trades regardless of impact, we would see the cumulative return marginally decreasing. However, we have noted that the number of trades executed decreases with an increasing impact. Therefore it is logical to see the cumulative return make more abrupt drops; as a function of increasing impact, the strategy is both making less trades and losing profitability on each of its trades.