Jim Holmes
Udactiy
Machine Learning
March 23, 2019

Capstone Report

Definition

Project Overview

Back in 2001 or 2002 I was very serious about becoming a day trader. I visited a few of the proprietary trading firms in Chicago. I even had Toby Crable's day trading book. Over the years I have worked at startups with ex CME and Citadel employees even though we are all just Network engineers.

At first I was looking to research a basket of equities that I could trade based on technical analysis. But was drawn to trading the S&P 500 E-Mini and the Nasdq 100 E-mini futures contracts because of the ability to trade the broader market with single instruments.

I remember when a member named QUAH, on a forum came up with a simple trading strategy called SVS - Something Very Simplistic and S/SVS - Son of Something Very Simplistic from 2002. 17 years ago the only way to test these trading ideas was to paper trade to see if worked. Some people could back test but not many people had the access we have today to test trading strategies.

The VIX Futures Basis: Evidence and Trading Strategies

> A study then demonstrates the profitability of shorting VIX futures contracts when the basis is in contango and buying VIX futures contracts when the basis is in backwardation with the market exposure of these positions hedged with mini-S&P 500 futures positions. The results indicate that these trading strategies are highly profitable and robust to transaction costs, out of sample hedge ratio forecasts and risk management rules. Overall, the analysis supports the view that the VIX futures basis does not accurately reflect the mean-reverting properties of the VIX spot index but rather reflects a risk premium that can be harvested.

Exploiting Term Structure of VIX Futures

Is a machine learning strategy that utilizes the VIX for it's trades. It does use the ES Futures contract as part it's algorithm.

With recent technological and data processing advances and new machine learning (ML) techniques it would be possible to test these two strategies using modern tools. Even with these technologies getting historical 1 minute futures data is expensive, so I will be using daily close information. Also, I don't want to "scalp the ES." I would like to find a longer term trend to hold a position for a specific point gain.

The following is the data source that was uploaded to github in .csv format.

Quandl SRF-Reference Futures E-mini Source

Quandl Barchart Continuous Futures E-Mini Source

I will be using data from the contract dates

- June ESH 2009 contract with the dates from 2009-06-18 to 2008-03-20
- June ESH 2010 contract with the dates from 2008-12-09 to 2010-02-19

This project intends to apply ML classification techniques to validate a buy or sell signal from these strategies and momentum indicators. I will be using futures data from quandl (barchart).

From a personal point of view, I would like to use this as strategy bases for future trading.

Problem Statement

This project's objective is to identify a buy / sell opportunities and position hold times for the S&P 500 E-Mini futures contract based on momentum indicators. Reinforcement learning techniques will be applied - Q Lerner - will be used.

Metrics

Since we are using the the futures market as our problem statement, our output metrics are relatively simple. We will be looking at a few different metrics to determine our rate of return over time with a learning algorithm for a specific futures contract. This will be visually represented by graphs.

To determine performance, we will use the Sharpe Ratio, along with different returns of the Portfolio vs the Benchmark. This is done quite simply with:
CumRet = (PresentValue - StartingValue) / Starting Value
This gives us a straightforward percentage. Calculated over time we can also look at our average daily reward which is just the mean of our daily rewards.

Since we are dealing with financial data, the clearest metric is our overall return irrespective of time. We are also using the Sharpe Ratio as an additional financial comparisons. This gives us an idea of performance other than just a portfolio dollar amount.

Analysis

Data Exploration

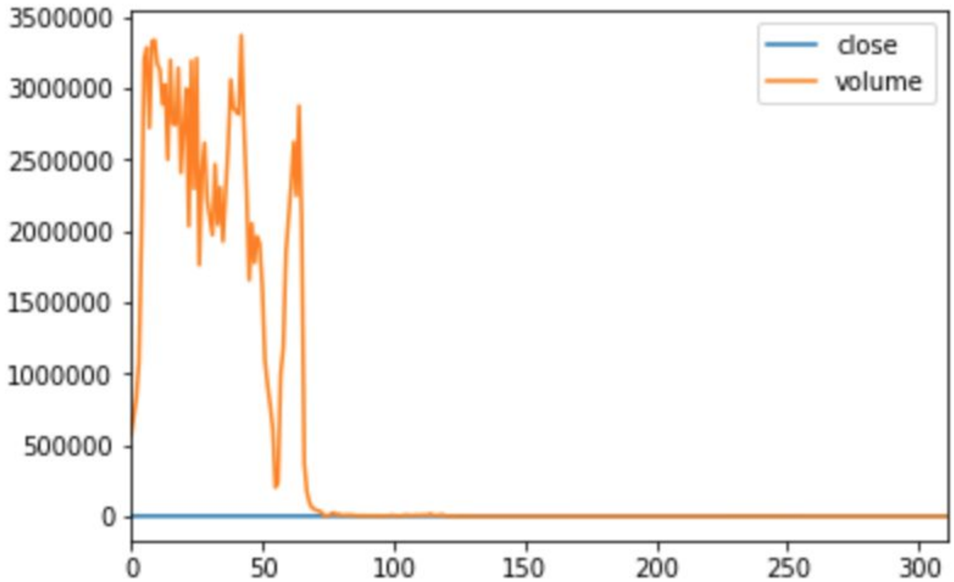
Equity futures market financial data is openly standardized. Historical data is provided as end-of-day OHLCV (Open-High-Low-Close-Volume) in CSV format. A sample is below after loading the data into pandas:

	date	open	high	low	settle	volume
None						
0	2019-04-05	7565.00	7612.75	7556.00	7604.25	356900
1	2019-04-04	7581.25	7598.00	7522.00	7561.50	429681
2	2019-04-03	7517.00	7616.25	7513.25	7582.25	479933
3	2019-04-02	7495.25	7532.50	7483.75	7519.25	358305
4	2019-04-01	7430.00	7506.50	7428.50	7498.50	410749

We will be using the settle column as the Adj Close even though futures contracts really do not have an Adjusted Close, it will help keep the data consistent if we wanted to compare futures to equities market data.

Exploratory Visualization

The graph below is a display of the settle price of the ES2009 Futures contract along with the volume. This is the primary data we will be working with.



With just the raw contract data as visualized above you can not derive a lot of information beyond the overall trend where the volume is much higher than the contract settle price. Our goal is to use this data to create additional features (preprocessing) to try to extract information that would be useful in creating a forecast of price movements.

Algorithms and Techniques

We will be using a Q-learning (reinforcement learning) on this problem. Q learning is well suited to financial time-series data since the reward state is easily defined - return on investment. This allows us to create a simple problem statement for the learner: " identify a buy / sell opportunities and position hold times for the S&P 500 E-Mini futures contract based on momentum indicators."

In our case, I have selected an alpha of 0.19 and a gamma of 0.93. Alpha will be the learning rate and gamma will be the future rewards value. This means that we have a relatively slow rate of learning - lower alpha - and we value our future returns pretty highly - higher gamma. In testing this proved the most appropriate action.

We will be feeding the learner with the OHLCV data (above) as well as three indicators generated from this base data. I have selected to use Bollinger Bands , momentum indicator and a simple moving average (SMA) using a rolling mean of -1.

Benchmark

Since we are using futures data, the benchmark we will be using is the VIX. Even though the VIX is not traded it will be used as a benchmark to trade against. We will be using 5 futures contracts of our target futures contract (CME S&P 500 E-Mini) and using the VIX as our benchmark and seeing if we can execute trades to beat that price or find areas of divergence where the two markets cross.

Methodology

Data Preprocessing

There are no issues with the source data since there are not many providers for historical futures data.

Since we used futures data based on the continuous contract. There are no instances when the upcoming futures contract becomes the front contract but the current futures contract is still traded. In this case the volume starts to dry up.

Implementation

The implementation of the Q-learner itself was a standard implementation, the challenging part of the implementation was all the processing and tracking of the market indicators and buy and sell signals - then simulating that to develop a real-world look at how the trades worked over time.

I defined the indicators I would be using. In my case, as mentioned above, I used Bollinger Bands, SMA and momentum along with the raw pricing data.

Bollinger bands was developed with a simple standard deviation for the top and bottom bands as well as a percentage change indicator. The SMA was a rolling mean And the momentum indicator used the following calculation:
momentum[t] = (price[t]/price[t-window]) - 1.

Additionally, simulating the market was challenging. Having to track all the variables and execute all the trades on a day-by-day (if needed) and then calculate portfolio values over time compared to the benchmark took a little bit of trial and error. In the end, just cycling through it day by day was the easiest means of completing it.

Refinement

Using a Q-Learner gives you the option to adjust your alpha and gamma values. I started with a 0.25 alpha and 0.85 gamma to have a moderate learning and a moderate value on future earnings.

The results were ok, a 0.97% gain.

```
Performances during training period for ESFULL
Date Range: 2016-12-23 00:00:00 to 2019-03-14 00:00:00
Sharpe Ratio of Portfolio: 0.33120380319821296
Sharpe Ratio of Benchmark : 0.74540894918218

Cumulative Return of Portfolio: 0.009737499999999955
Cumulative Return of Benchmark : 0.02761250000000004

Standard Deviation of Portfolio: 0.0009847982984126941
Standard Deviation of Benchmark : 0.0010807235328127679

Average Daily Return of Portfolio: 2.054671203433263e-05
Average Daily Return of Benchmark : 5.074683257779249e-05

Final Portfolio Value: 100973.75
Final Benchmark Value: 102761.25
```

For a midpoint run I used .22 alpha and .91 gamma
The results were still not good, a 0.22% gain.

```
Performances during training period for ESFULL
Date Range: 2016-12-23 00:00:00 to 2019-03-14 00:00:00
Sharpe Ratio of Portfolio: 0.08666556911645469
Sharpe Ratio of Benchmark : 0.74540894918218

Cumulative Return of Portfolio: 0.0022374999999998924
Cumulative Return of Benchmark : 0.02761250000000004

Standard Deviation of Portfolio: 0.0009453152064889126
Standard Deviation of Benchmark : 0.0010807235328127679

Average Daily Return of Portfolio: 5.160870563948003e-06
Average Daily Return of Benchmark : 5.074683257779249e-05

Final Portfolio Value: 100223.75
Final Benchmark Value: 102761.25
```

For a high-point run I used .19 alpha and .95 gamma
The results were better, but still a 1.10% gain.

```
Performances during training period for ESFULL
Date Range: 2016-12-23 00:00:00 to 2019-03-14 00:00:00
Sharpe Ratio of Portfolio: 0.35857851270412144
Sharpe Ratio of Benchmark : 0.74540894918218

Cumulative Return of Portfolio: 0.010974999999999957
Cumulative Return of Benchmark : 0.027612500000000004

Standard Deviation of Portfolio: 0.0009529217726750479
Standard Deviation of Benchmark : 0.0010807235328127679

Average Daily Return of Portfolio: 2.1524904888087806e-05
Average Daily Return of Benchmark : 5.074683257779249e-05

Final Portfolio Value: 101097.5
Final Benchmark Value: 102761.25
```

For a final run I used .19 alpha and .96 gamma.

The resulted in a loss as well, but still a 0.89% gain.

```
Performances during training period for ESFULL
Date Range: 2016-12-23 00:00:00 to 2019-03-14 00:00:00
Sharpe Ratio of Portfolio: 0.31408292248618347
Sharpe Ratio of Benchmark : 0.74540894918218

Cumulative Return of Portfolio: 0.008899999999999908
Cumulative Return of Benchmark : 0.02761250000000004

Standard Deviation of Portfolio: 0.0010261815675709266
Standard Deviation of Benchmark : 0.0010807235328127679

Average Daily Return of Portfolio: 2.030337623420603e-05
Average Daily Return of Benchmark : 5.074683257779249e-05

Final Portfolio Value: 100890.0
Final Benchmark Value: 102761.25
```

I decided to stick with .19 alpha and .95 gamma as the final alpha / gamma values.

Results

Model Evaluation and Validation

Overall, the reinforcement learner performed as expected. My in-sample performance was successful with a modest gain. However the out-of-sample performance was not successful. The results are not entirely unexpected, since beating the market based on simple machine learning is highly unlikely or there would be a lot more stock traders walking around.

Below, are graphs of the results of both in-sample and out-of-sample performance.
The vertical lines indicate when the model made a trading decision (green lines indicate a BUY and red lines a SELL).

Figure 1: In-Sample ES Contract (2016-12-23 to 2019-3-14)

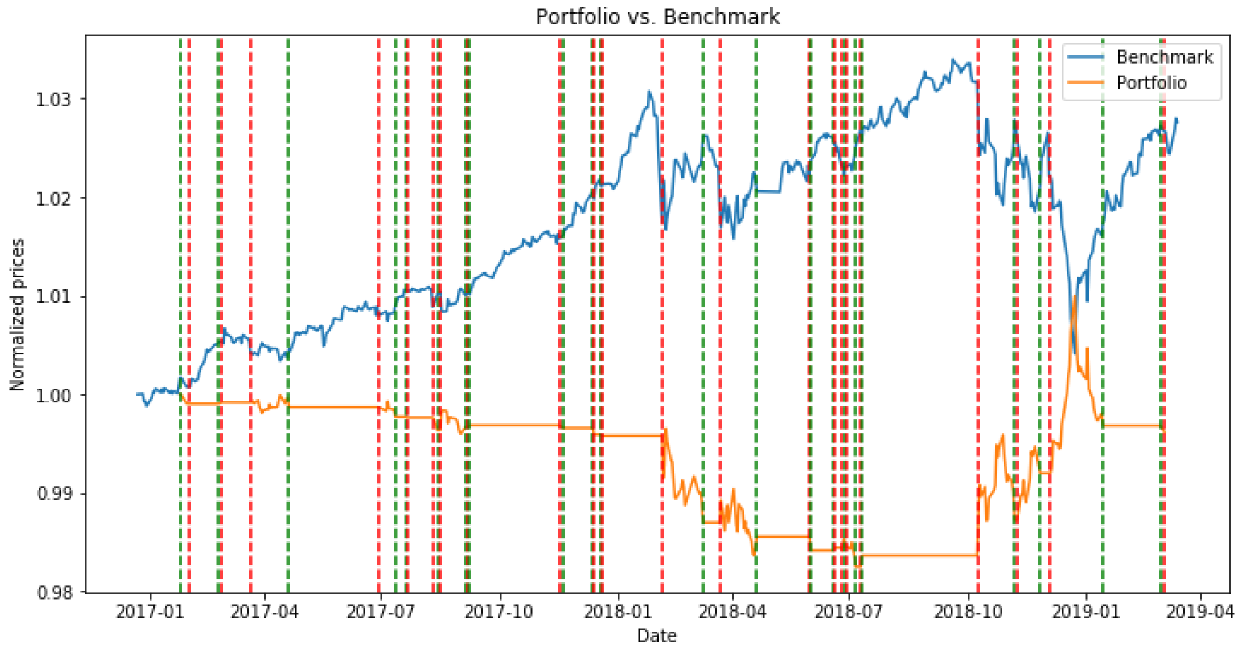


Figure 2: Out-of-Sample ES Contract (2014-9-11 to 2016-12-22)

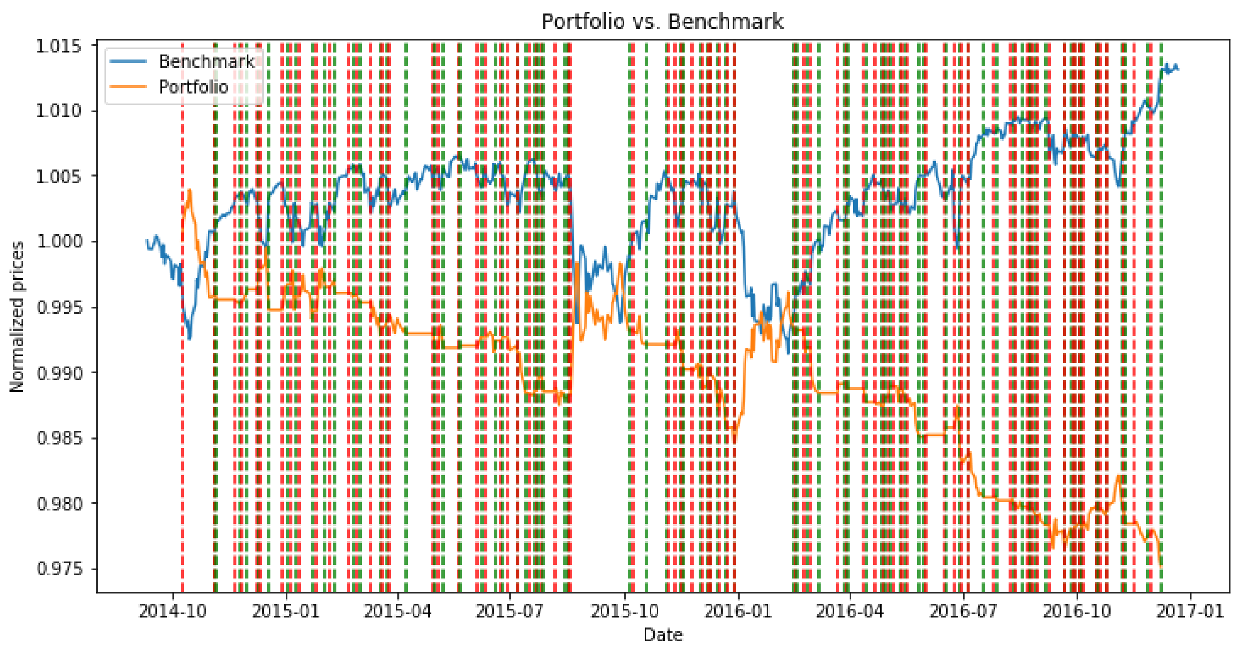
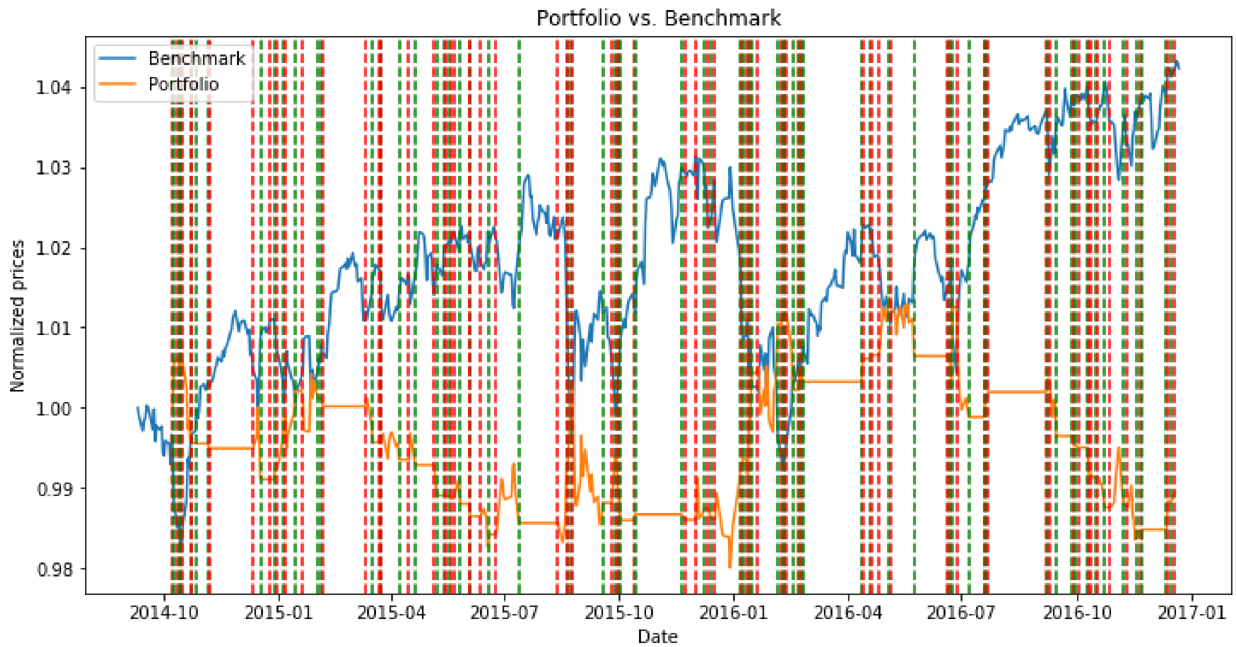


Figure 3: Out-of-Sample NQ Contract (2014-9-11 to 2016-12-22)



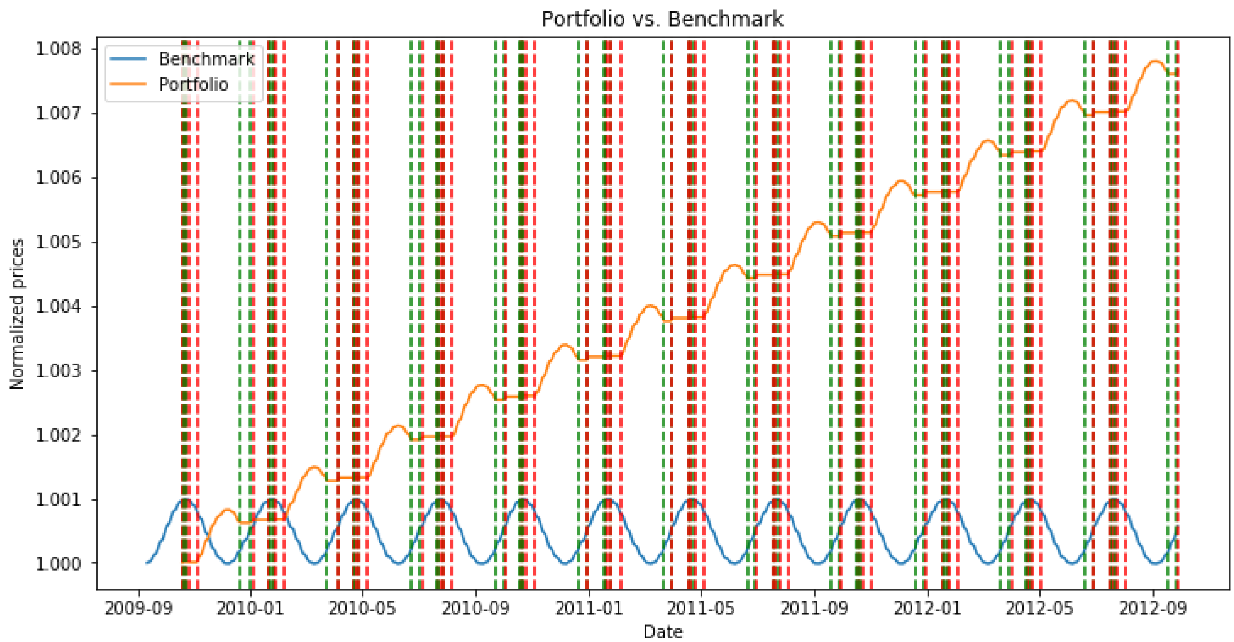
I included a third out-of-sample run using the NQ Emini NASDAQ 100 futures contract source data with the same continuous contract date range to check the sensitivity of the learner. This also allowed it to model a different type of futures contract compared to the primary subject E-Mini S&P 500 (ES).

This model is reasonable considering what it is -- modeling the futures market is difficult given the Efficient Market Hypothesis. Working with end of day data also has its challenges due to missing out on small fluctuations on intra-day trading.

I feel this model meets my expectations. I would not trust the output of the model for live trading given the 5 contract size for a 1% gain. But a combination of learners in agreement would be more likely to produce reliable results.

A reinforcement learner should be robust for the datatypes it handles (time-series financial data). As a comparison, below is the output using the model with a sine wave dataset.

Figure 4: In Sample Model Robustness sine-wave data (2009-9-11 to 2012-9-30)



As can be shown in this sine-wave input file the model performed as well as the other data-sets with a cumulative return of 0.76%. This test shows that the model is robust and can handle variations in input data and is well generalized on the data input into the model.

Justification

Final results - from starting value of $100000:

Portfolio In-Sample Performance Cumulative Return	**Portfolio** Out-Of-Sample Performance Cumulative Return	**Portfolio** In-Sample Performance CME-NQ Cumulative Return		**Portfolio** In-Sample vs Out-of-sample Cumulative Return
$100,713.75 (+0.71%)	$96,908.75(-3.93%)	$100648.75(+0.64%)		$3,091.25 (-2.48%)

Benchmark In-Sample Performance Cumulative Return	**Benchmark** Out-Of-Sample Performance Cumulative Return	**Benchmark** In-Sample Performance CME-NQ Cumulative Return		**Benchmark** In-Sample vs Out-of-sample Cumulative Return
$102,761.25 (+2.76%)	$101,308.75 (+1.38%)	$104,227.50 (+4.23%)		$1,452.50 (+1.41%)

You can see that the Benchmark performance return on the primary S&P 500 futures contract (ES) beat the portfolio by 2% for in-sample, and a wide margin for the out-of-sample performance.

When you add in the NASDAQ 100 futures contract (NQ) using the in-sample model, both the Portfolio and Benchmark still out performed by 3.5%.

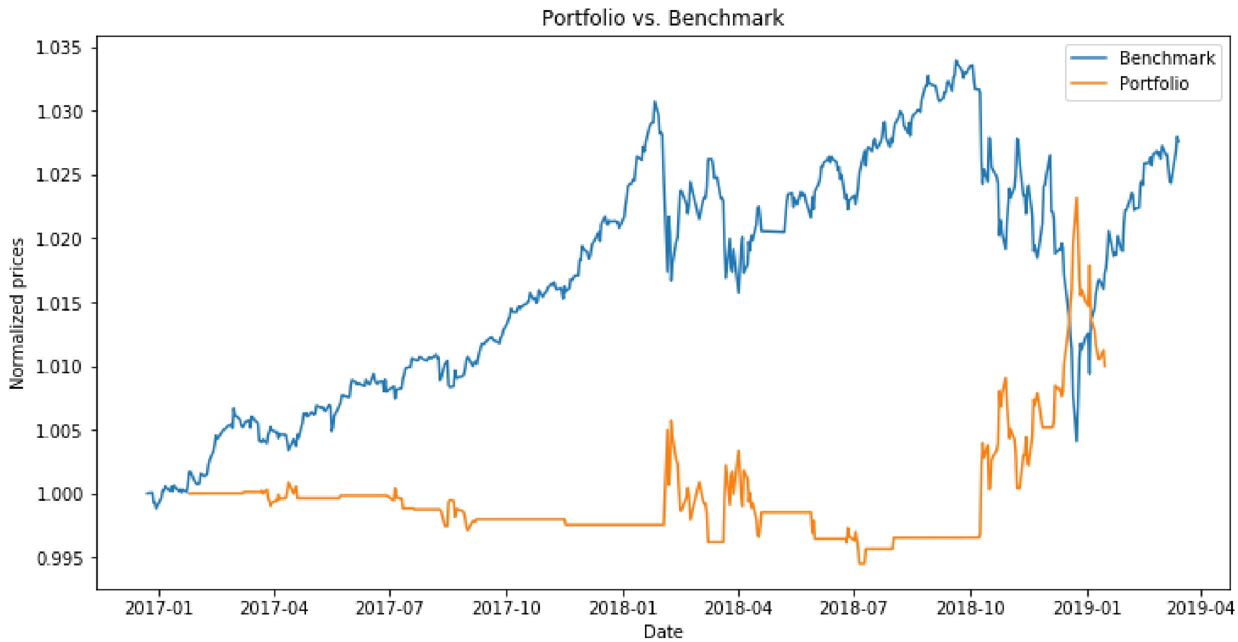
Overall the Benchmark performed better in all instances.

I do not think there are other indicators things that could be tried; like EMA instead of SMA. It would just end up a case of trial-and-error. The more indicators you use, the larger the state table becomes and you introduce more noise. Stock market analysis, like many things, benefits from a few specialized indicators over time.

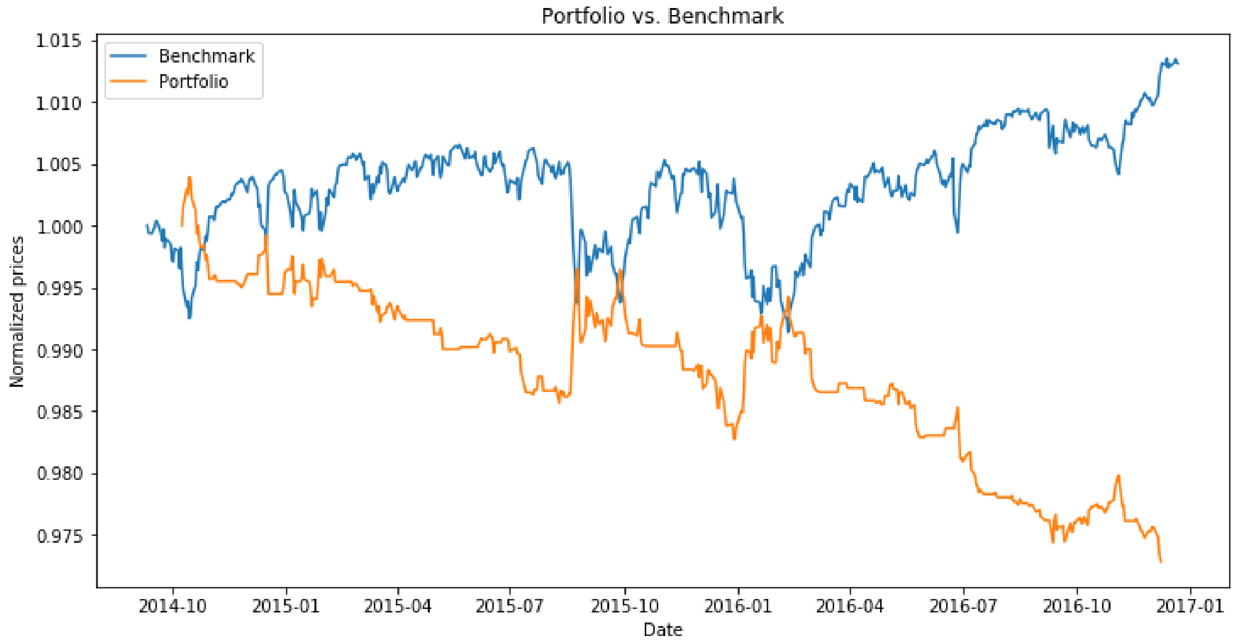
Conclusion

Free-Form Visualization

In-sample



Out-of-Sample



Above you can see a graph of the ES Futures contract from the In-sample and out-of-sample test with the trades removed. Without the trade bars you can see some indications of a reversal with the in-sample chart, where the benchmark (VIX) and the Portfolio (ES) seem to be pointing at each other. Starting in late 2018 you can start to see a trend and by early 2019, you can see the two briefly converge.

For the second chart with the out-of-sample tests, you can see more times of convergence. This is with different trade dates, but much more convergence opportunities.
While busy you can see one item which is of interest and worth further investigation if I were pursuing this.

Reflection

This was a challenging project from start to finish. While it was easy to think of using the Futures market instead of the Equities market, coming up with a problem statement proved more difficult. I didn't go about it as "can you beat the market" since I think there is more to a successful strategy. The actual implementation contained multiple moving parts to both model and analyze the results. Reinforcement learning appears to be a solid choice as a learner for stock-market time-series data.

A challenging aspect of this project was turned out to be easier than expected. Using a Q-learner you are constantly concerned about state: how big, how much, etc. If you use stock data as-is and the indicators as calculated you end up with an infinite number of states.

Discretization is how you solve this. By binning your results over a range you reduce the number of states to a finite value while still retaining a lot of the data's value.

By definition:
"In applied mathematics, ***discretization*** *is the process of transferring continuous functions, models, variables, and equations into discrete counterparts. This process is usually carried out as a first step toward making them suitable for numerical evaluation and implementation on digital computers."*

What I also learned from this project, for stock market data, a learner is only one part of the problem. You need to validate your output, you need a market simulator that would go through and execute the trades as they were generated and allow me to see the overall returns.

While I would not trade with this model in any way, I could see continuing development on it to become another datapoint when dealing with futures trading. Investment firms spend many millions of dollars on this specific problem to gain an edge, that's really long odds to gain an edge.

Improvement

The main improvement that I would look into is the use of the VIX Benchmark in a different way. Since this was a start for a trading model I chose to use price data, but I think there are different ways to use the VIX. The VIX may also help identify the convergence you see between the two models.

If I were really trying to do this successfully I would also use a quorum of learners rather than a single q-learner. This kind of problem is likely helped by using multiple different types of learning and then generating quorum for your buy/sell/hold decisions. For example a neural network and a random forest along with the q-learner would probably be much more effective than a q-learner alone.