

Markov_technical

July 18, 2021

1 Hidden Markov Models on Chinese Commodity Futures

Nick Holden

2 Literature Review

- *High-order Hidden Markov Model for trend prediction in financial time series* (Physica A, 2019). Zhang et al use almost exactly the same methodology as the one we adopted. Though they used 3 hidden states and tested their method on CSI300 and SP500 data. They also considered higher order HMMs and tried some dimension reduction techniques which we did not experiment with in our analysis.
- *Hidden Markov Models Applied To Intraday Momentum Trading With Side Information* (arXiv:2006.08307, 2020). This paper makes two innovations on the common HMM method. First, rather than being pre-specified, its model is estimated using cross validation, penalized likelihood criteria and simulation-based model selection. Also, it considers using side information as part of the feature (observation) vector to improve trading signal performance.

3 Trading Environment

This model trades 23 of the most liquid commodity contracts traded in the two futures exchanges in Dalian and Zhengzhou, China. The contracts include a wide range of commodities including non-ferrous metals, precious metals, agricultural products and chemical products. Most contracts are traded weekdays 8:30-11:30 am and 9-11 pm or continuously 24 hours a day 5 days a week. Average annual contract volumes are in the hundreds of millions, and the market is quite sizable and liquid. An important attribute of the Chinese commodities market is its immaturity. On each ticker within the set of main expiry commodities contracts, there exists a roughly 2-3% wide spread. This allows for significant scalping and market making opportunities. We also note a small influx of “smart money” investors slowly entering this trading environment.

4 Model Specifications

A Hidden Markov Model (HMM) is a state-space model that can be used to model a wide range of processes. It consists of hidden states that evolve over time as a Markov process, and at each time the hidden state emits an observation according to a noisy distribution. With the observations, one can learn the transition and emission probabilities as well as infer the hidden state of the system based on observations. We model the process that generates the observed returns as an HMM. We

assume at each time t , the market is in a hidden state $0, \dots, 6$, and determine empirically on the training data the state with the strongest positive momentum (“bullish state”) and the state with strongest negative momentum (“bearish state”). At bullish states we are more likely to see positive returns in a future time period and at bear states the opposite holds. We learn the HMM using historical data and we deduce hidden states with the HMM by feeding it price-volume features extracted from a fixed historical window. We use a Gaussian HMM model from the *hmmlearn* Python package.

Compared to other state-space models such as LSTM, HMM has the advantage of being simple and more interpretable. The model has fewer parameters to estimate and is therefore more robust. Another advantage of HMM over comparable models is that we know the state of the system at any time, whereas models like LSTM are a complete black box. Existing momentum trading models suffer from time-lagging caused by the delayed frequency response of digital filters. Time-lagging results in a momentum signal of the wrong sign, when the market changes trend direction. A key feature of this state space formulation is that no such lagging occurs, allowing for accurate shifts in signal sign at market change points. Our HMM strategy essentially captures momentum of the market by labeling it into bull and bear states. Compared to simple mechanical momentum strategies such as using the MACD or fast-and-slow moving averages, HMM has the advantage of learning its parameters from the observations so it can make full use of our historical data. HMM can also take in much more complex features compared to simple momentum strategies that can only take in past prices in a straightforward manner.

Below, we have an example plot of the model’s predicted hidden states exhibited by Chinese Silver Futures in 2019.

```
[34]: plot_hidden_states(df19,hidden_states19,6,'Predicted Hidden States on Silver_
      ↪2019',states19[0],states19[1])
```



Visually we can see that the model detects bullish and bearish states reasonably well. We develop a trading strategy on this model and backtest it below.

5 Trading Strategy

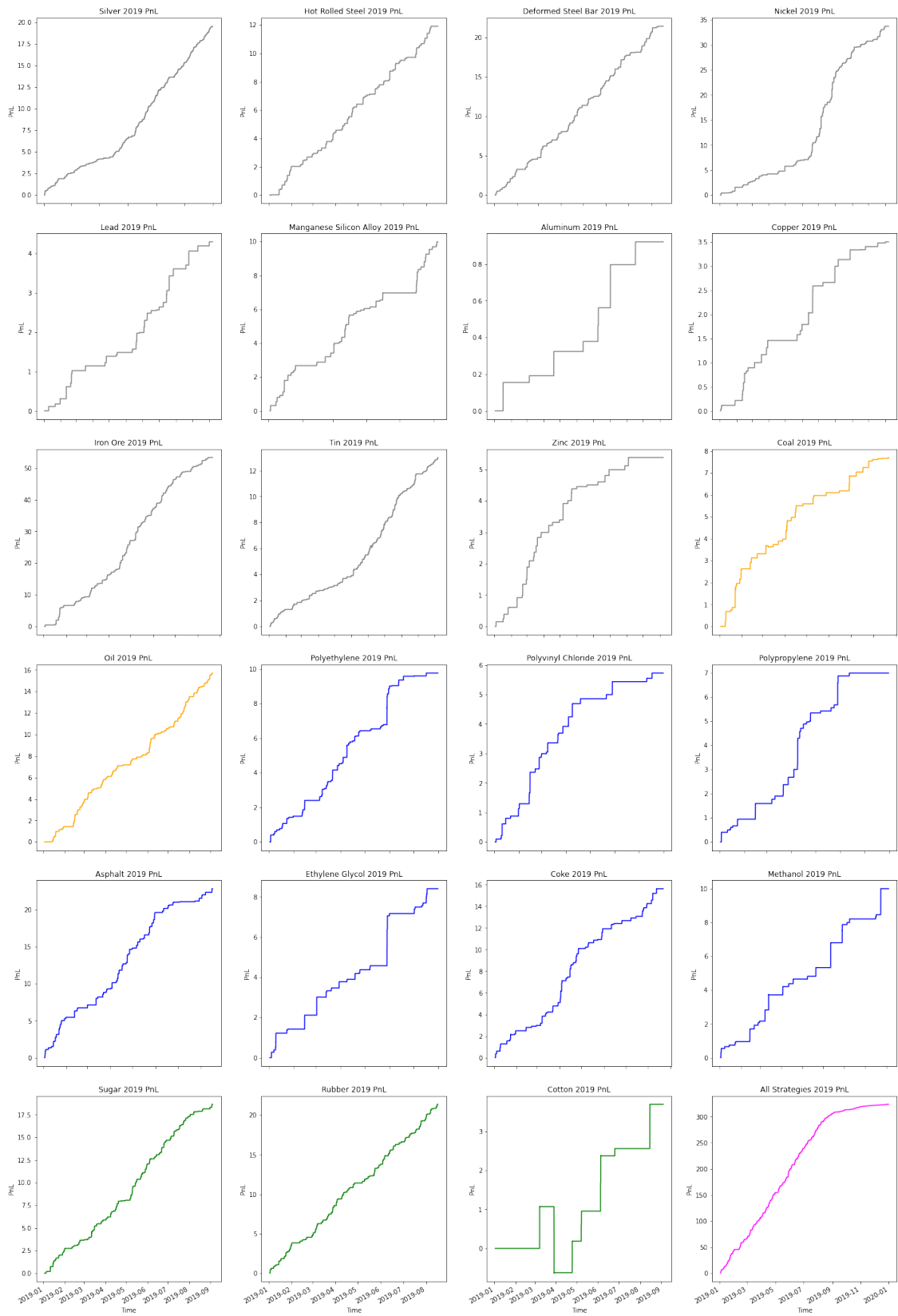
- We learn our HMM model using historical data from each commodity. Then in real time, our model will determine the current hidden state of a given commodity. On bullish states, we buy and hold the security before selling thirty minutes later. We run this model on our cross section of 23 main expiry contracts exhibiting their respective bullish states, buy at minute t and hold until time $t + 30$. We allocate cash in a $\frac{1}{n}$ manner equally across all bullish state exhibiting contracts. 30 minutes serves as our holding parameter because many human CTA traders use candle stick plots at 1/5/30min frequency. I believe 30 minutes is a good time frame to capture market psychology. Also, this time frame is long enough to be unaffected by liquidity constraints and microstructure effects.
- This model does not include a stop loss. This is to demonstrate the effectiveness of buying on the bull state; a stop loss does not help accomplish this. Additionally, a 30 minute is short enough such that dramatic moves are highly unlikely. The code for implementing the strategy can be found on my github.

6 Performance Analysis

First, consider the performance of the strategy on each future's contract denoted by the cumulative PnL over the time period. We include exchange transaction costs in our calculation, and assume we can purchase a security at it's listed price. We plot these below for 2019.

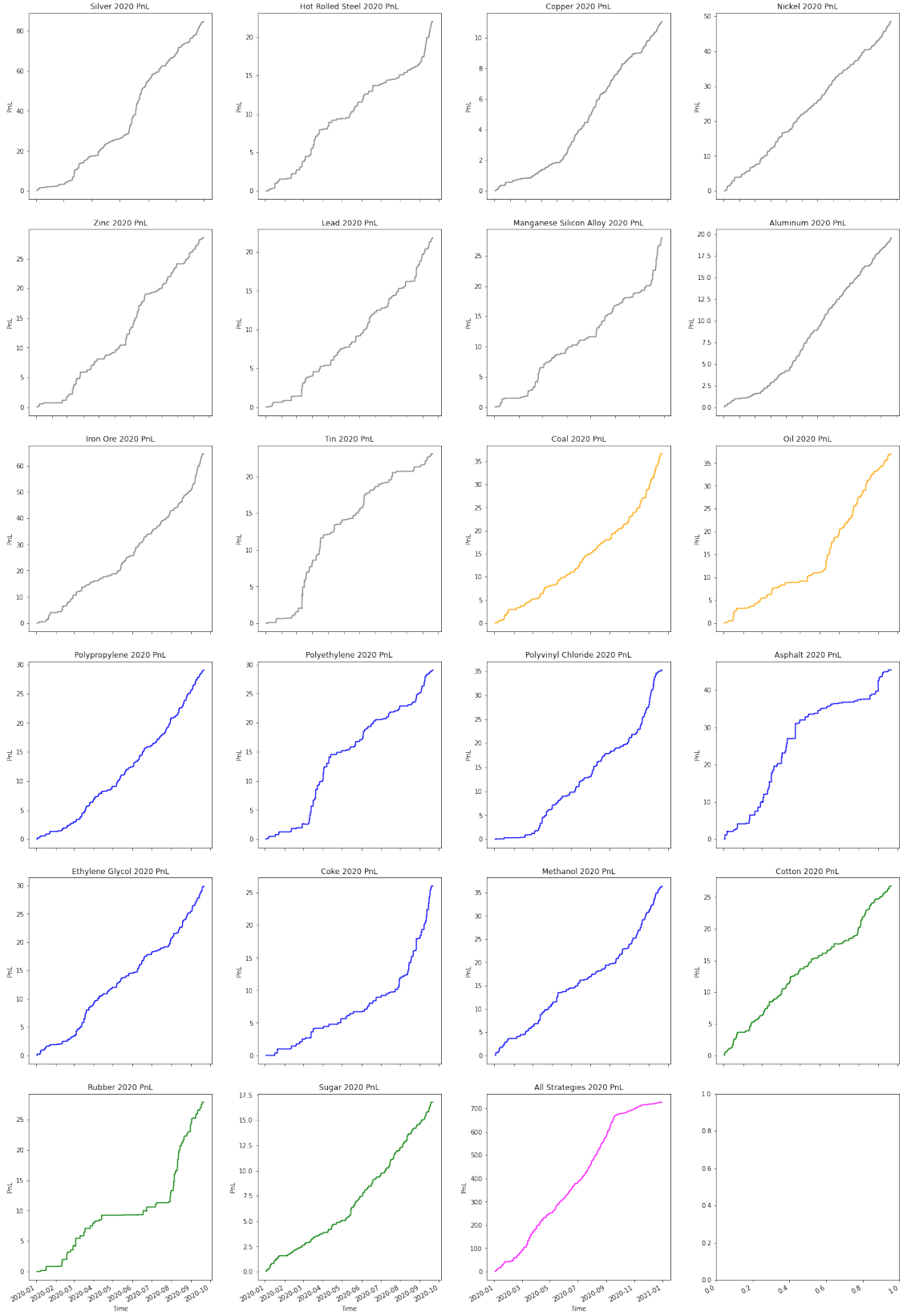
- *Note, the units of PnL on the y-axis are a multiple of a constant investment size on bullish states (ie if we invest k every time our model detects a bullish state, the y-axis represents the multiple of k we will profit by running our strategy).*
- The commodities are loosely color coded by industry so the interested reader might glean some industry-specific trends. The indicated industries are metals (grey), energy (orange), chemical manufacturing (blue), and agricultural (green). We also color the combined strategy of investing in all of these commodities in pink.

[44]: `plot_pnl(coms2019)`



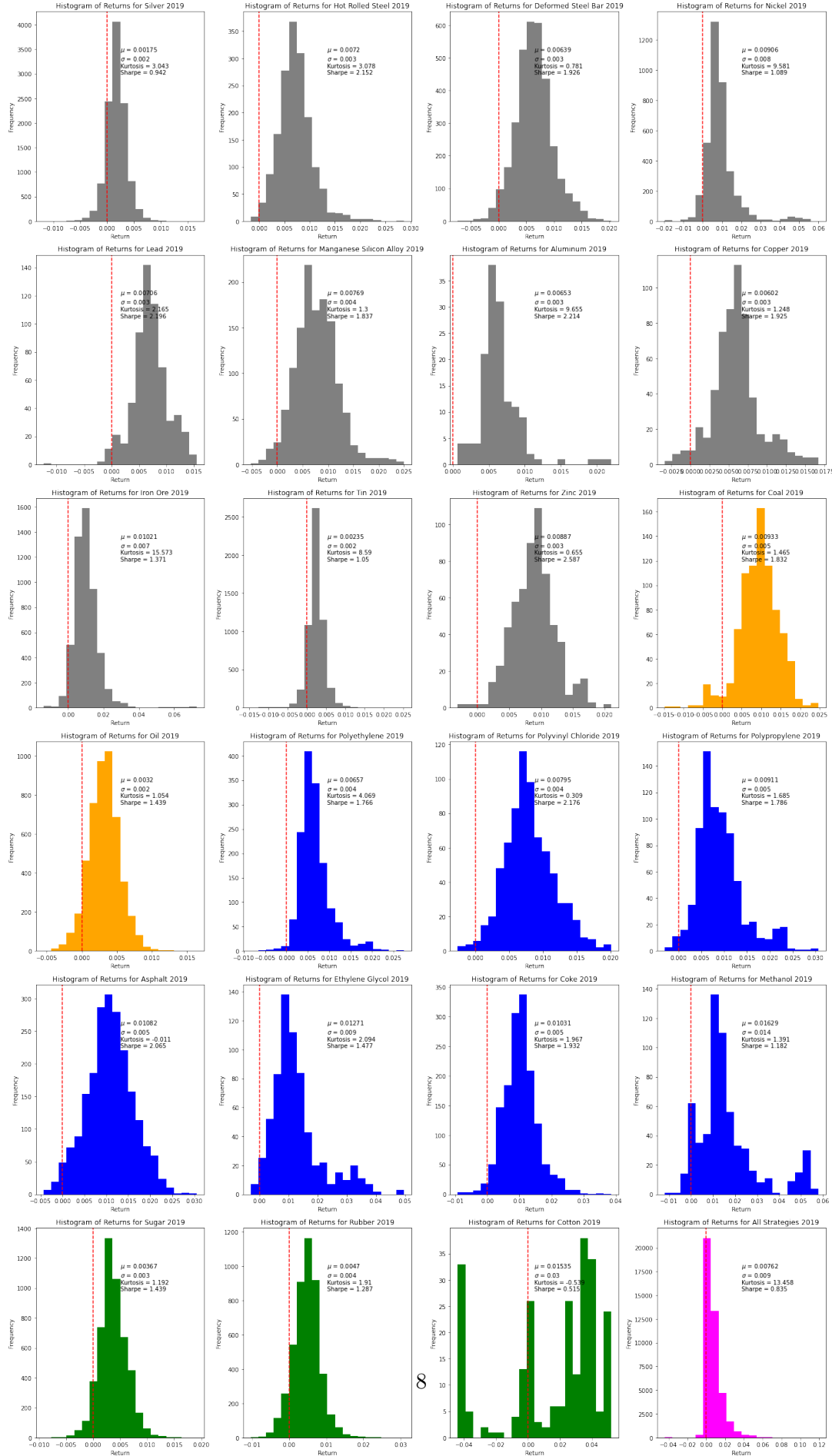
Now, we plot PnL for 2020.

```
[8]: plot_pnl(coms2020)
```



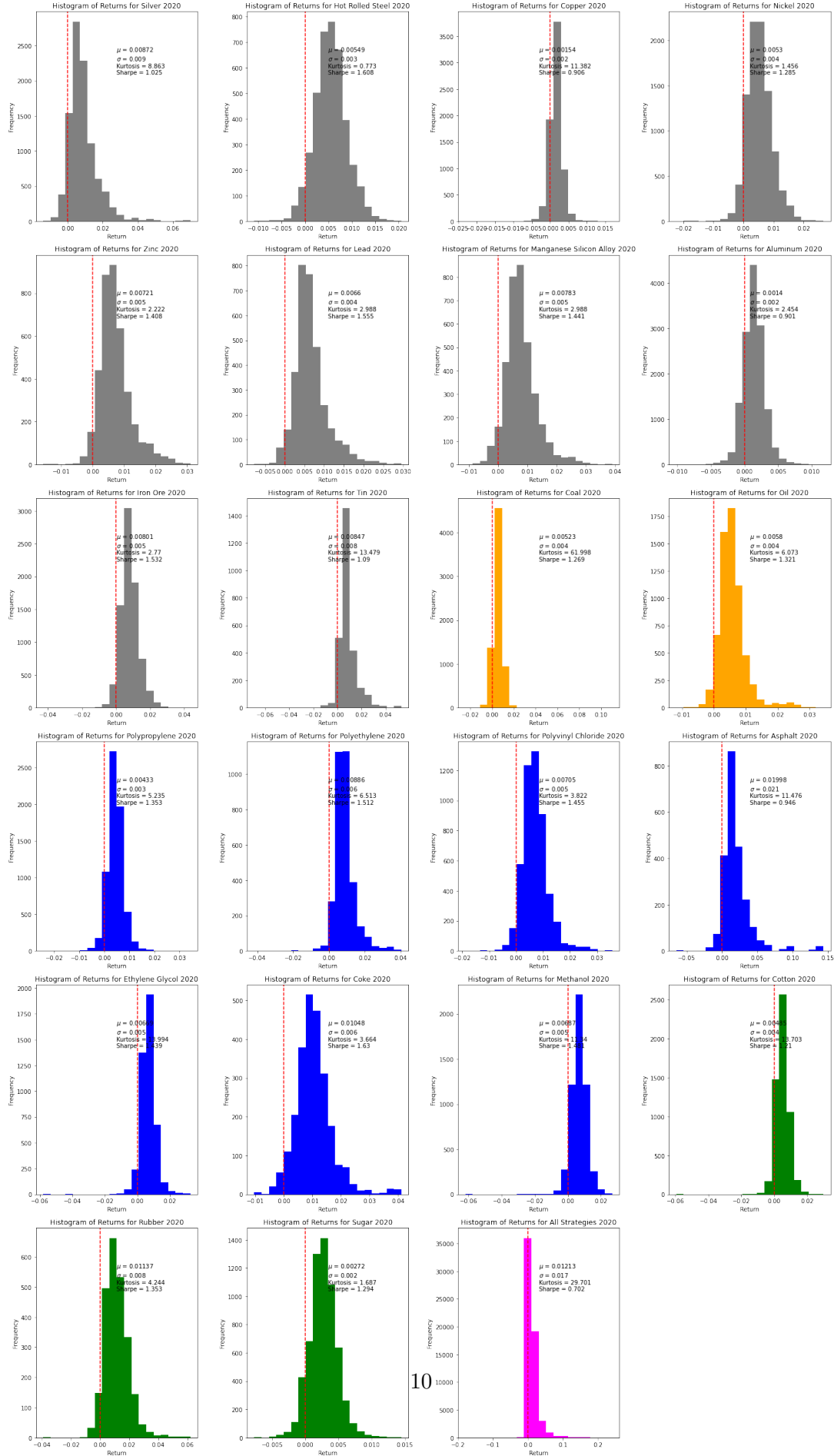
Next, consider the distributions of minutely returns by commodity for 2019. We plot them.

```
[65]: plot_histograms(coms2019)
```



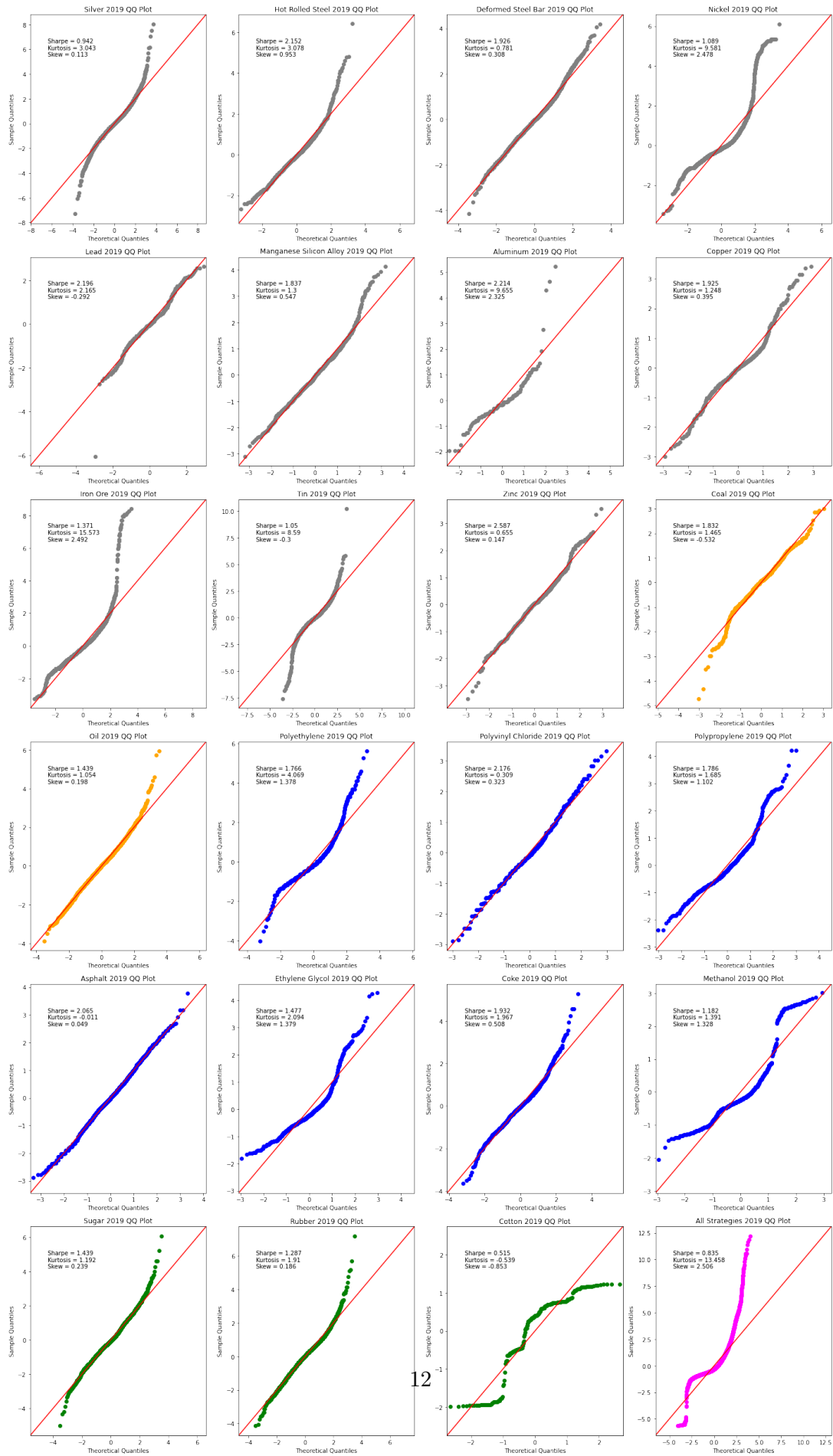
And now we plot them for 2020.

```
[7]: plot_histograms(coms2020)
```



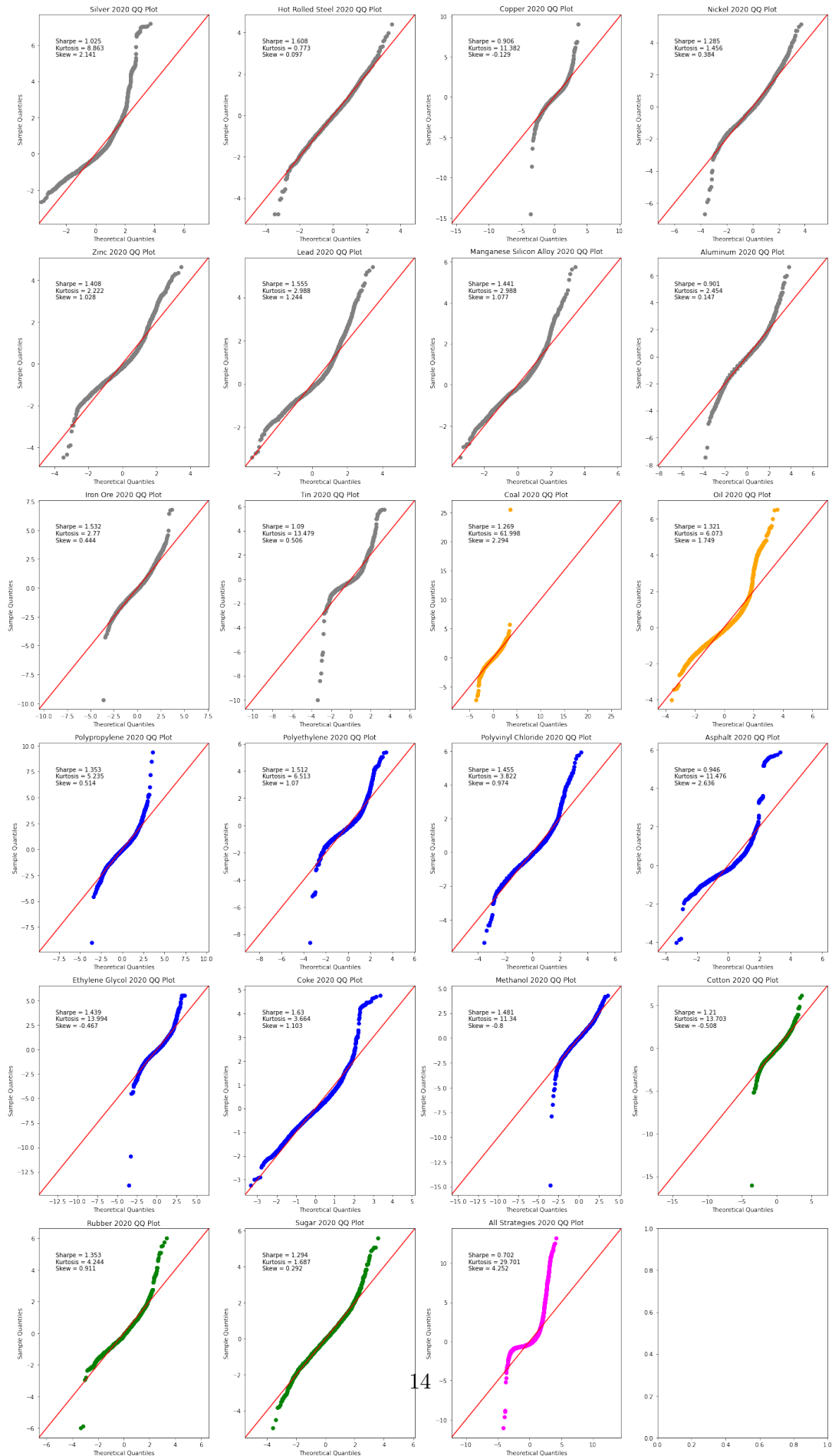
First, note the difference between these commodities in terms of investibility. Some commodities exhibit much higher Sharpe ratios, higher kurtoses, and higher mean returns than others. Contrary to our expectations, the trading environment appears to be improving, with Sharpe ratios and returns more attractive in 2020 than in 2019. This is a counterintuitive result considering the influx of traders to this immature market. Continuing, we now plot the QQ plots of each strategy. QQ plots are a useful tool for determining if and how a distribution deviates from a normal distribution. We first plot for 2019:

```
[21]: plot_qq_plots(coms2019)
```



And now 2020:

```
[20]: plot_qq_plots(coms2020)
```



From the QQ plots, we see our distributions on average have fatter tails than a normal distribution. This is consistent with the observed kurtosis values. We also observe a right skew for most futures, particularly the combined strategy. This indicates that if we were to implement this strategy it likely will act similar to an inverse carry trade.

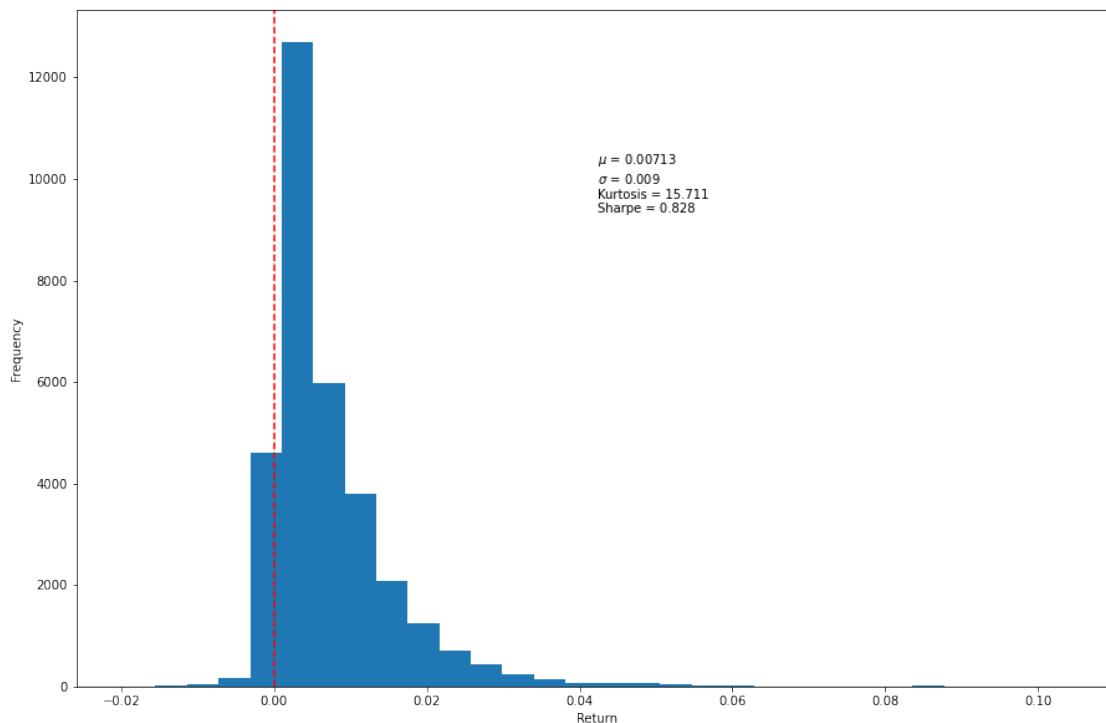
Before we interpret these *PnL* plots, let us consider the following distribution of returns for the cumulative strategy.

```
[21]: plt.figure(figsize=(15,10))

series = metals2019['netret'].loc[ (metals2019['netret'] != 0) &
    ~ (metals2019['netret'].isna())

plt.hist(series, bins=30)
plt.axvline(x=0, color='r', linestyle='--')
plt.annotate('$\mu$ = ' + str(round(np.mean(series),5)) + '\n$\sigma$ = ' +
    str(round(np.std(series),3)) + \
    '\nKurtosis = ' + str(round(series.kurtosis(),3)) + \
    '\nSharpe = ' + str(round(np.mean(series) / np.
    ~std(series), 3)),
    xy=(.5, .7), xycoords='axes fraction')
plt.xlabel('Return')
plt.ylabel('Frequency')
```

```
[21]: Text(0, 0.5, 'Frequency')
```



We see excellent PnL results. Further, we see a reasonably high Sharpe ratio for the cumulative strategy and excellent Sharpe results for individual contracts. Our mean cumulative return is positive and the return distribution is right skewed and highly kurtotic. Note that we include the exchange listed trading costs of 0.0345% in our backtest. This strategy seems to be very profitable.

To properly interpret the PnL plots we define recurring investment amount K as the nominal amount invested into each security equally. We track cumulative PnL as a function of our constant, small, investment size. This investment size would need to be less than or equal to $\frac{K}{30 \cdot n}$ where n is the number of commodities we are trading and K is our total capital. This is because for each security, we can have up to 30 positions in that security (say we buy on $t, t + 1 \dots t + 29$). In our back test we ran this strategy for $n = 23$ commodities futures, so our high frequency trading size k would be $\frac{K}{690}$.

- There is certainly room for more detailed analysis of position sizing using a Markowitz model or modern portfolio theory, but that is not the goal of this exercise. There is also an argument to be made for $\frac{1}{n}$ sizing (see *Optimal Versus Naive Diversification: How Inefficient is the 1/N Portfolio Strategy?* DeMiguel, Garlappi, and Uppal)

7 Risks

Across these commodities, we see a large bid-ask spread of roughly 2-3%. Our backtest results have a mean return of roughly 0.7% and thus incorporating the spread into backtesting would hurt performance heavily, most likely moving our mean past zero. Then, we would see somewhat of an inverse carry trade, with a slightly negative mean return, right skew, and high kurtosis.

While bid-ask spreads are large, we note that there is significant trading volume, with annual Chinese commodity trading totaling 48 trillion dollars. Thus, we see minimal liquidity risks.

An additional risk comes from the nature of the HMM itself. Because HMM states are inherently hidden, there can be no economic rationale or logical basis for their existence. Thus, should a sizable shock affect the commodities market, the strategy's reaction is not predictable, as the states are not known and cannot be manually corrected even if the effect is known to the traders. This represents risk during regime changes, geopolitical events, and other statistically rare market shocks. Further, risks for these events are hard to control since the underlying states cannot be altered.

It is also worth considering currency risk. By trading on Chinese commodities, we expose ourselves to currency risk to convert earnings back to US dollars. Additional risk is imposed when converting our initial investment to Yuan. Should the strategy hold sufficient outside investor capital, the lack of liquidity for a client can be a large downside to investing.

8 Conclusions

Overall, we see that this model is quite promising. However, without tick-level data to examine the effects of the wide bid-ask spread, we cannot make a definitive statement on the tradeability of the HMM strategy. We can say that using an HMM as a predictor of equity returns appears to be a promising avenue for traders, particularly in the developing Chinese commodities market. It is also worth noting that our HMM uses a single layer for prediction. Current literature suggests that deepening the model with more layers or vectorizing the inputs to the HMM can yield better prediction results and subsequently more profitable and consistent trading. Additionally, we have chosen parameters that are able to be scaled sufficiently and allows for implementation, such as our K capital parameter and 30 minute window.

[]: