Sparse Mean Reverting Portfolios using Semi Definite Programming

Introduction

In this iteration of the SMRP strategy, I have used semi definite relaxations on non-convex optimization problems. The core idea is to minimize three mean reverting criteria- namely the VAR(1) predictability, the portmanteau statistic at lag 10 and the crossing statistic whilst ensuring that the generated synthetic portfolio is sparse and has a minimum threshold for variance in order to be profitable. The addition of these two constraints makes the problem non-convex and thus Semi Definite relaxations are needed. Weights for the two most optimal assets are extracted using a sparse PCA procedure and then the portfolio is expanded to 5 assets using the fast PCA procedure. This step is repeated multiple times with randomisation in order to generate ~90 - 100 slightly different mean reverting portfolios. The portfolios are then truncated to eliminate noisy allocations and then validated by trading them using an EWMA approach on the training set and a validation set. The top 2 portfolios that yield positive returns during these two periods as well as exhibit ADF stationarity at a p-value of 0.25 are then traded on the test period.

The Stock Universe

In an ideal world, one would use similarity and mean reverting metrics (viz. hurst exponent) to group stocks and then use this optimised basket of stocks to find mean-reverting portfolios from. I had a universe of ~ 50 tickers and it was unfeasible to club these together without the inherent stringency of the metrics reducing the basket to a paltry 3-5 stocks. A very quick way to circumvent this was to choose all stocks in the following GICS level 1 sector classifications:

1. Materials
2. Consumer Discretionary
3. Consumer Staples
4. Health Care
5. Financials

This leads to a final universe of :

```
custom_tickers = ["AXISBANK.NS", "BAJAJFINSV.NS", "BAJFINANCE.NS", "HDFCBANK.NS", "ICICIBANK.NS",
                  "KOTAKBANK.NS", "SBIN.NS", "BANKBARODA.NS", "PNB.NS", "ITC.NS", "BRITANNIA.NS",
                  "HINDUNILVR.NS", "DABUR.NS", "APOLLOHOSP.NS", "CIPLA.NS", "DRREDDY.NS", "SUNPHARMA.NS",
                  "BAJAJ-AUTO.NS", "HEROMOTOCO.NS", "TVSMOTOR.NS", "TITAN.NS", "TRENT.NS",
                  "AMBUJACEM.NS", "ULTRACEMCO.NS", "VEDL.NS", "TATASTEEL.NS", "GRASIM.NS", "UPL.NS"]
```

The Optimisation Problem

We will separately minimise three criterias subject to sparsity and minimum variance constraints using the SDP formulations below :

1) VAR(1) predictability: ***How close the time series is to noise in terms of variance***

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(MY) + \rho\|Y\|_1 \\ \text{subject to} & \mathbf{Tr}(A_0 Y) \geq \nu \\ & \mathbf{Tr}(Y) = 1,\ Y \succeq 0 \end{array} \tag{SDP1}$$

2) Portmanteau Criterion at lag 10: ***How close the time series is to noise in terms of autocorrelation***

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{p} \mathbf{Tr}(A_i Y)^2 + \rho\|Y\|_1 \\ \text{subject to} & \mathbf{Tr}(BY) \geq \nu \\ & \mathbf{Tr}(Y) = 1,\ Y \succeq 0, \end{array} \tag{SDP2}$$

3) The crossing statistic with mean 0.1 and lag 10: ***How frequently the time series crosses its mean***

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(A_1 Y) + \mu \sum_{i=2}^{p} \mathbf{Tr}(A_i Y)^2 + \rho\|Y\|_1 \\ \text{subject to} & \mathbf{Tr}(BY) \geq \nu \\ & \mathbf{Tr}(Y) = 1,\ Y \succeq 0 \end{array} \tag{SDP3}$$

The decision to choose a lag of 10 and a mean of 0.1 is a hyperparameter that has been chosen after testing out different lag periods and mean values.

For minimum variance, I have used a tenth of the median variance of all assets in the baskets over 1/9th the total period chosen to run,train, validate and test the strategy. Training takes place for 4/9ths of the period,with validation and testing each for 2/9ths of the period. This is an attempt to first and foremost avoid any look ahead bias and ensure training takes place for twice the period as the test and validation. The regularisation parameter(which induces sparsity), rho is set to 0.00001. This was found to be optimal with sparse PCA weight selection, going any higher makes the overall optimisation too stringent . This causes the solver to generate portfolios of fewer than 2 assets, which is flawed by design.

The autocovariance matrices have been computed using returns and not prices of the assets. This is since returns have been empirically and statistically known to be more robust and modelable than simple price levels.

Extracting weights from the SDP

The above listed SDP's to generate SMRPs do not directly generate weights, rather an optimal matrix Y* from which optimal weights y* need to be extracted. This extraction procedure can either be achieved using sparse PCA, enforcing it to generate the top 2 sparse assets. This method is preferred, however, this enforced sparseness on top of the optimization problem (that already has sparsity constraints) fails to generate two asset portfolios, in this case we can use a regular PCA procedure, extracting the eigenvectors corresponding to the least eigenvalue. The slight disadvantage of this is that we need to iterate over all possible two asset portfolios, solve the SDP, extract their regular PCA weights and thus choose the two assets that minimize each statistic. This adds to computational overhead but still produces relatively accurate results that are close to the sPCA procedure.

Greedy fast-PCA Method to generate portfolios

Once we obtain the two optimal assets, we use a greedy fastPCA approach to greedily choose 3 more assets that minimize the minimum eigenvalue of the covariance matrix thus formed. We also introduce randomness in the stock universe to generate portfolios that slightly differ from one another. We greedily generate these 5 stock portfolios 30 times for all the three different criterias to effectively obtain 90 different mean reverting portfolios.

Truncation Method for Sparsity

Once we have obtained optimal weights for 90 different, 5 asset portfolios we use the truncation method to remove noisy allocations. The truncation method is a heuristic approach that removes assets by comparing their normalised weights. We divide absolute weights by the price of the asset and rank these in descending order. The top 4 assets by this metric are chosen. This metric effectively eliminates small, noisy allocations whilst ensuring that the mean-reverting characteristics of the SMRP are retained. We set the weights of the eliminated assets to 0 and rescale the top 4 assets' weights so that the portfolio remains fully invested.

Trading the spread

Once we have portfolio weights, we construct the spread and start trading it. The trading rules are simple, I use a 15-day rolling Exponential Weighted Moving Average(EWMA) of the spread with upper and lower bands at +/- 1.5 standard deviations above and below the 15 day rolling EWMA. When the spread goes below the lower band, we go long the portfolio and exit at the mean. Similarly, when the spread goes above the upper band, we short the portfolio and exit at the mean. Additionally, there are **-1% / +3% PnL thresholds** to avoid drawdowns

Validating the SMRPs

In order to gauge the effectiveness of our aforementioned algorithm, we use the training and validation period to check all the 90 portfolios. The check can be for mean-reverting nature or more directly we can use the modified bollinger bands strategy to check whether these portfolios netted positive returns during both of these periods. This check is essential to see if the portfolios behave radically different in the training and validation period before we actually set out to trade them. This ensures that we are more confident that the portfolios will show similar mean reverting characteristics in the actual testing phase. Furthermore, we also check for stationarity of these portfolios in the validation and training period at a p-value of 0.25. This high value is used since we do not want to penalise too much for non-stationarity but don't want portfolios that completely drift away either. From these two checks we choose the top 2 portfolios that pass and have the least average train and validation p-value from the ADF test.

The Results

```
Period breakdown:
- Reference period: 2015-03-24 to 2015-05-23
- Training period: 2015-05-23 to 2016-01-21
- Validation period: 2016-01-21 to 2016-05-21
- Test period: 2016-05-21 to 2016-09-21
Median variance: 0.000418
Volatility threshold: 0.000042
```

Fig 1: Random Period chosen to run the strategy

```
Selected 2 out of 4 portfolios that passed checks:
  Portfolio 15: ADF p-value = 0.040940 (train), 0.094377 (val)
    Tickers: ['HDFCBANK.NS', 'KOTAKBANK.NS', 'BRITANNIA.NS', 'BAJAJ-AUTO.NS', 'GRASIM.NS']
    Weights: [-0.5537, 0.2011, 0.0356, 0.0573, 0.1522]
  Portfolio 77: ADF p-value = 0.129865 (train), 0.036234 (val)
    Tickers: ['HDFCBANK.NS', 'KOTAKBANK.NS', 'CIPLA.NS', 'BAJAJ-AUTO.NS', 'TATASTEEL.NS']
    Weights: [-0.5949, 0.2931, 0.0063, 0.0983, -0.0075]
```

Fig 2: Top 2 most optimal portfolios from the initial 90 SMRPs

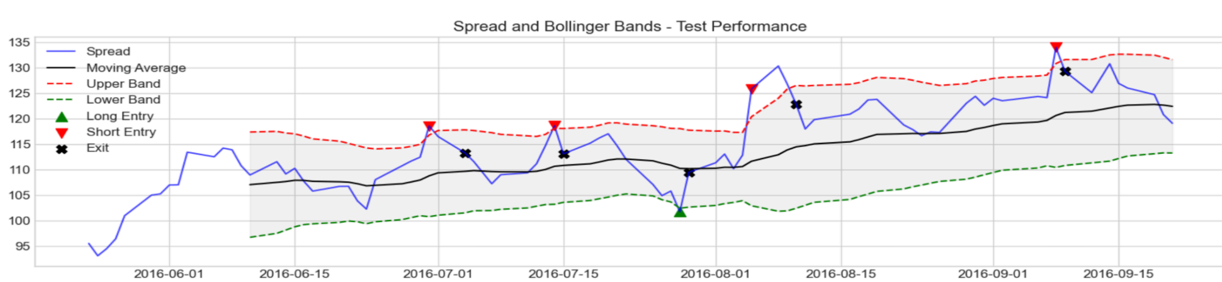

Fig 3: Performance of chosen portfolio 1 over test period

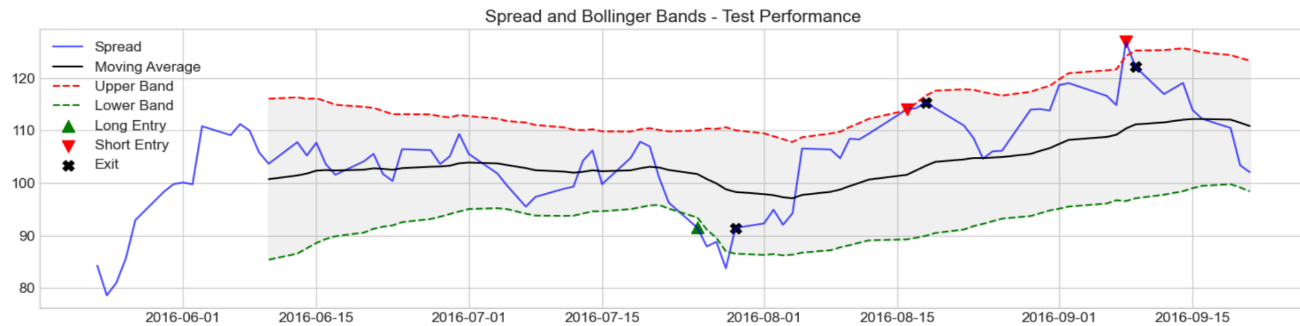

Fig 4: Performance of chosen portfolio 2 over test period

```
Analysis completed in 8.00 seconds (0.13 minutes)

Final Combined Test Performance:
Total Return: 0.1405
Annualized Return: 0.4834
Annualized Volatility: 0.1293
Sharpe Ratio: 3.3910
Maximum Drawdown: -0.0327
Hit Rate: 0.7692
```

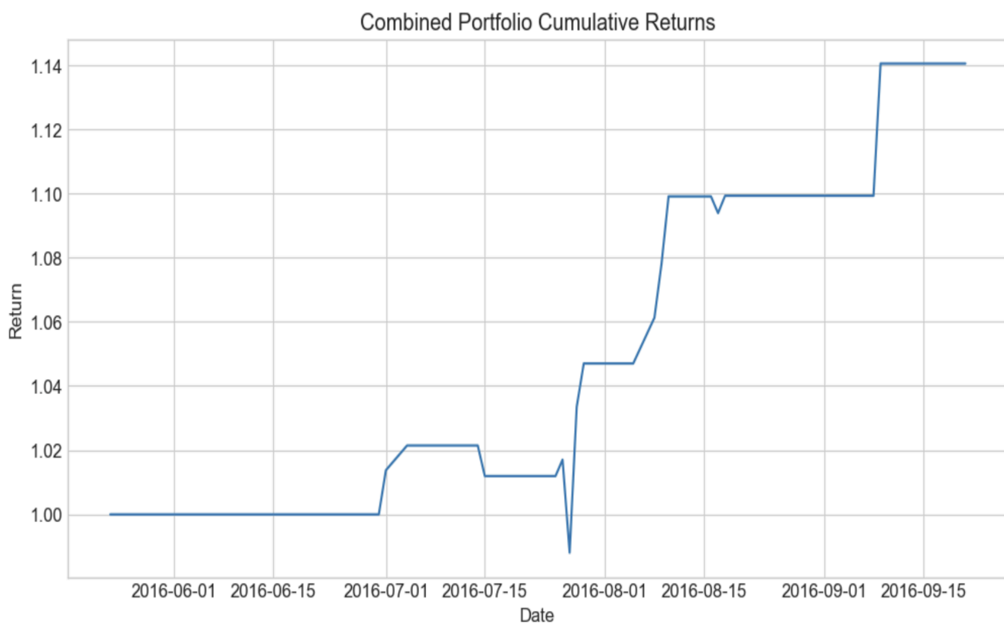

Fig 5: Performance of the combined portfolio

The figures above show how the strategy performed for a given randomly chosen period. It has a Hit Rate of ***0.7692***, an Annualised Sharpe of ***3.39*** and a relatively modest Maximum Drawdown of ***3.27%.***

It is worth noting that due to the conservative approach adopted in the strategy of robust return checks and added stationarity checks over two periods, the strategy simply does not trade when none of the portfolios pass these checks. This makes it less susceptible to incurring large losses.

Future Improvements

1. Using intraday data and higher frequencies like per- minute close price data to gauge the viability and profitability of the strategy.
2. Using a larger stock universe and choosing clusters using the hurst exponent and DBSCAN
3. Adding an additional factor based risk model that makes the portfolio immune to factor risk i.e. making it factor neutral so that it is driven purely by idiosyncratic returns

Resources

1. *https://arxiv.org/pdf/0708.3048*
2. *https://arxiv.org/abs/1509.05954*
3. *https://www.scirp.org/journal/paperinformation?paperid=100361*
4. *https://www.algos.org/*