Pairs Trading Optimization with Time series prediction

Ruben Alejandro Ruiz Gomez (rr737), Majda Essadiqi (me449)

Abstract

We investigated statistical arbitrage and portfolio selection using pairs trading. In the first part of this project, cointegration modeling was used to choose valid pairs among several industries. Trading signals that can capture the mean-reversion of the spread between the prices of two stocks are constructed based on the fitted model. In the second part of this project, the idea of pairs trading was optimized incorporating different frameworks. For a more realistic approximation, we developed a Recurrent Neural Network algorithm to predict a path of 90 days, based on past data to train the strategy and then tested it in the actual path. Then we intend to optimize the trading signals with Bayesian Optimization and Spectral Fourier Analysis maximizing the profit and minimizing the risk at the same time. Results were discussed and possible future improvements are suggested to improve the performance of the algorithm.

1. Pairs Trading

Pairs trading is a strategy that uses convergence relationships to trade a pair of securities. The strategy was widely used in the 1980s. In theory, many pairs share similar patterns of movements, such as a pair of companies that make products that are alternatives to each other (Pepsi and Coke), or a pair of companies that make products that are supplements to each other (Chevron and Exxon). Therefore, when the two price series diverge, they are very likely to converge again as the Law of One Price indicates that the price of the same asset or commodity will have the same price globally. Traders use this relationship to open trades when the prices diverge (long the stock with lower price and short the stock with higher price). **Figure 1** below shows the stock price movement of Goldman Sachs and J.P. Morgan, and we could see that the log stock prices follow similar trends during the selected period.

Figure 1. Log stock prices of Goldman Sachs and J.P. Morgan: 1/2/1998 - 12/30/2018

The market-neutral property of pairs trading allows the coordination between a long position and a short position in a pair of cointegrated assets. This strategy generates profit from the difference of the price changes of the pair instead of the direction of each asset. Another advantage of this method is the possibility to utilize it during various market situations whether the market goes up or down.

1.1 Cointegration Based Pairs Trading

The pairs trading strategy is a market neutral approach because traders are making profits on the convergence relationship, instead of the trend of the stock prices. In this scenario, the shorted security and the longed security would always hedge each other. Therefore, even if the market crashes, the pair trading will profit as long as the convergence relationship still holds. To test the pair trading strategy, we first form the pairs out of potential stocks. While some researchers use the Euclidean distance approach to test comovement of pairs, we focus on the cointegration approach alone based on the following reasons we will discuss below.

In the Euclidean Distance approach, the spread variance of securities i and j with prices p can be denoted as:

$$s^2_{p_i - p_j} = \frac{1}{T}\sum_{t=1}^{T}(p_{i,t} - p_{j,t})^2 - (\frac{1}{T}\sum_{t=1}^{T}(p_{i,t} - p_{j,t})^2$$

T is the total period of time, $p_{i,\,t}$ is the price of the stock i at time t. And the average sum of squared distance could be calculated as:

$$\overline{ssd}_{p_i,\,p_j} = \frac{1}{T}\sum_{t=1}^{T}(p_{i,t} - p_{j,t})^2 = s^2_{p_i - p_j} + (\frac{1}{T}\sum_{t=1}^{T}(p_{i,t} - p_{j,t}))^2$$

Then, to minimize SSD, in the most ideal scenario, we need to find a zero spread variance $s^2_{p_i - p_j}$ with a zero average spread $\frac{1}{T}\sum_{t=1}^{T}(p_{i,t} - p_{j,t})$. However, from the standpoint of pair traders, we would like to select security pairs with a frequent divergence and strong convergence. Distance approach always supplies us with pairs with strict comovements and a minimized tendency of diverging, leaving less room for trading profits. Therefore, we will not focus on this approach.

Cointegration, on the other hand, provides us with high convergent pairs. Here, cointegration takes 2 time series $(x_1,\, x_2,\, ...,\, x_n)$ and $(y_1,\, y_2,\, ...,\, y_n)$. First, it tests if both the time series are of order d, which means that $(1 - L)^d x_t$ is a stationary process (L is the lag operator and (1 - L) is the first difference). Then, it tests if a linear combination of the two time-series (ax+by) is of order 0, which means that $\sum_{k=0}^{\infty}|b_k|^2 < \infty$ (b is the vector of the moving average weights). When these two series of stock prices pass both steps, there exists a cointegration relationship between them.

In this paper, we will perform cointegration tests on the indexed price processes in order to select the pairs with high convergence: Engle-Granger and Johansen tests are run on the two price series to check if there is a cointegration relationship. After checking for the cointegration relationship, we would also use the Augmented Dickey-Fuller test to check for the stationarity of the series. To expand on the details of the test process, the corresponding model of the Engle-Granger test can be shown as:

$$R1_t = \alpha + \beta R2_t + \varepsilon_t, \; t = 1,2,...,w$$

Where $R1_t$ and $R2_t$ are the time series returns at time t, α and β are the fitted coefficients, and ε_t is the fitted residual at time t. In the process of the Engle-Granger test, we first fit the OLS regression for $R1_t$ and $R2_t$. Next, by fitting the model, we find the values of α, β and ε_t. After finding the fitted values of the coefficients, we calculate the sum of the fitted residuals in the time interval we defined, and finally run the Augmented Dickey-Fuller test on the sum of the fitted residuals to test for stationarity.

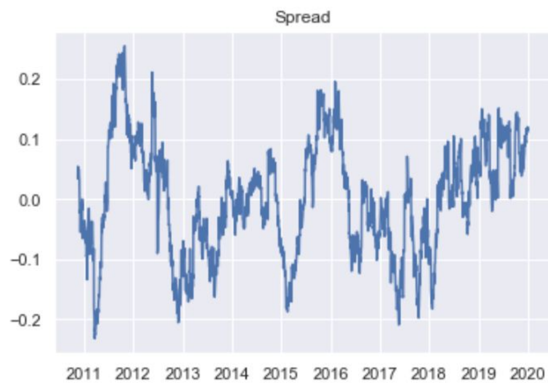


Figure 2: Spread for the pair NKE-LOW

1.2 Prescreening of Stock Pairs

Since we have a universe of thousands of stocks, choosing pairs through running cointegration tests on all possible pairs is not a viable option, e.g. S&P 500 comprises N = 505 common stocks and thus results in N×(N—1)/2 = 127,260 pairs of stocks. The most expedient way to reduce the number of pairs being tested is by using rules of thumb to choose those pairs that are likely to have high co-movement. [4]

Generally speaking, the candidate pairs are selected based on their industrial sector and market capitalization, because they have similar exposure to those common risk factors, and are thus more likely to have cointegrated returns. In this project, the top 10 holdings, by market capitalization, from the respective Sector SPDR ETF are selected, with a total of 50 stocks. We believe that any two of those 10 top holding stocks within the same sector have a much higher probability of being a valid cointegrated pair. Using this prescreening rule, only 10× 5×(5—1)/2 = 100 pairs need to be considered in the trading strategy. This greatly improves the efficiency of the pair-selection process

Moreover, stocks from the same industry generally tend to have stronger correlations than those in the unrelated industries, this way, to help narrow down pairs further we compute correlations between stocks.
Correlation is measured by the correlation coefficient ϱ, an indicator ranging between -1 and 1. When it is equal to 1, there exists a perfect positive correlation between the two variables, -1 means there is a perfect negative correlation and 0 means there is no correlation between the variables.

Two highly correlated variables will have the same movements when one goes up the same will apply for the second variable with the same magnitude, similar patterns can be seen when one of the variables goes down. This shows that both variables are connected and this property can be used for trading.

The correlation coefficient for the two variables can be calculated as:

$$Correlation(A,B) = = \frac{Cov(A,B)}{\sigma(A)*\sigma(B)}$$

where Cov(A,B) is the covariance between A & B, $\sigma(A)$ and $\sigma(B)$ are the standard deviation of the respective variables.

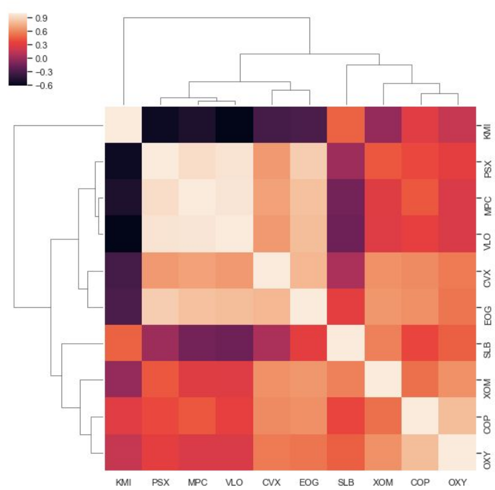


Figure 3. Energy Industry correlation matrix: 1/2/2008 - 12/30/2018

This property is interesting for pairs trading, for example, if the correlation between 2 stocks is high then if the price of the first asset goes up, so will the price for the second asset. With this characteristic, market neutral strategy is used when A is purchased and B is sold.
At that point, using the mean-reverting property of the pair while capturing a change in correlation creates a profit opportunity. However simply relying on the correlation is not enough as the stocks might continue increasing without ever mean-reverting. This is because the correlation approach is sensitive to small time deviations. Therefore, a deeper analysis is needed to select the appropriate pair with cointegration properties. **Figure 3** shows the correlation matrix among the stocks of the Energy industry. This is a heat map where lighter colors represent higher

correlation between those stocks. **Figure 4** shows a hexbin (A bivariate analogue of a histogram) of two highly cointegrated stocks, Dupont (DD) and Air Products and Chemicals (APD). The significant correlations shown in both figure 2 and figure 3 present that some of the preliminary conditions are fulfilled for our testing of the pairs trading strategy, yet we still need to conduct more practices to find out the returns this strategy could realize.

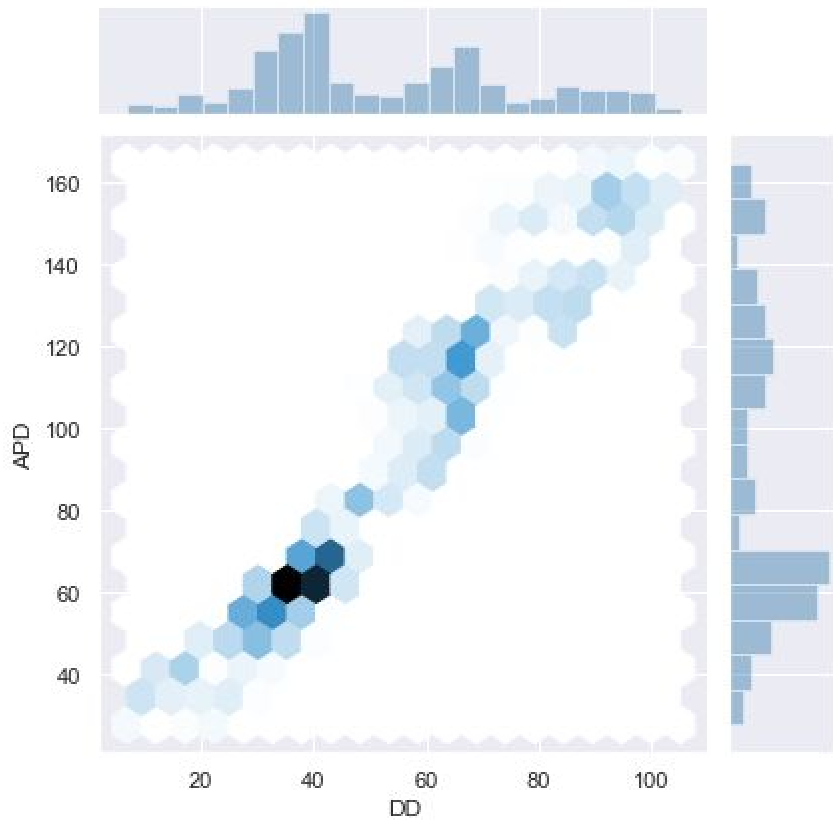


Figure 4. Hexbin of DD and APD : 1/2/2008 - 12/30/2018

1.3 Cointegration Testing and Identification of Stock Pairs

The most common test for Pairs Trading is the cointegration test. Cointegration is a statistical property of two or more time-series variables which indicates if a linear combination of the variables is stationary. In finance it can be used to find trading strategies based on mean-reversion. Two time series, $Y_{1,t}$ and $Y_{2,t}$, are cointegrated if each is I(1) but there exists a λ such that $Y_{1,t} - \lambda Y_{2,t}$ is stationary. For example, the common trends model is that

$$Y_{1,t} = \beta_1 W_t + \varepsilon_{1,t}, \; Y_{2,t} = \beta_2 W_t + \varepsilon_{2,t},$$

where β_1 and β_2 are non-zero, the trend W_t common to both series is I(1), and the noise processes $\varepsilon_{1,t}$ and $\varepsilon_{2,t}$ are I(0). Because of the common trend, $Y_{1,t}$ and $Y_{2,t}$ are non stationary but there is a linear combination of these two series that is free of the trend, so they are cointegrated. To see this, note that if $\lambda = \beta_1/\beta_2$, then

$$\beta_2(Y_{1,t} - \lambda Y_{2,t}) = \beta_2 Y_{1,t} - \beta_1 Y_{2,t} = \beta_2 \varepsilon_{1,t} - \beta_1 \varepsilon_{2,t}$$

is free of the trend W_t, and therefore is I(0).

1.4 A stationary process:

A stationary process is a stochastic process, where the joint probability distribution doesn't change when shifted in time.
As a result, its mean, variance, autocorrelation are all constant over time.
A first-order stationary process is defined as a process for which the first-order density function is independent of time. Consequently, the mean of this process is a constant,

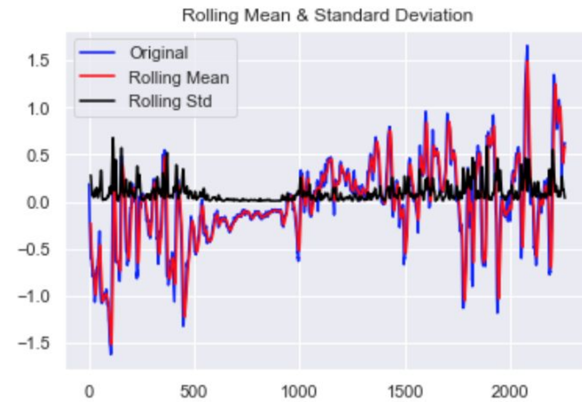
$$m_t = m$$

and the variance is also a constant,

$$\sigma_t^2 = \sigma^2$$

Stationary is very often an important feature in time-series analysis and applications of stock pairs trading. Pairs trading relies on the stationary property of the spread which allows for mean-reversion.

Additionally, having a stationarized series eases predictions since statistical properties remain constant over time, we applied this property later when with the RNN approach used to predict the spread path. **Figure 5** shows the stationarity of the linear regression of a stock pair.




```
Results of Dickey-Fuller Test:
Test Statistic                -8.065852e+00
p-value                        1.597112e-12
```

Figure 5. Stationarity test on the spread for the pair NKE-LOW

1.5 Training and Testing Data:

The stocks were selected based on the top holdings from 5 SPDR ETF (XLB, XLE, XLK, XLU, and XLY). The corresponding sectors are: Materials, Energy, Technology, Utilities, and Consumer Discretionary. Daily adjusted closing prices from 2010/1/2 - 2018/12/30 of the top 10 holding stocks from each sector are downloaded from the Yahoo Finance API. Since some of the companies haven't gone public at the start of the screening period, we decided to remove that particular stock for simplicity.

Typically before modelling we would split our feature and target data into training/test sets, but because if we'd like to train on as much data we have available at a specific timestep, we take the walk forward approach. This involves constantly training or refitting the model and predicting one step at a time. We first removed the last 90 days of the known dataset and stored it as the testing set. Then we implemented a RNN in order to predict a spread path to replace the removed data. We used this new whole path as our training dataset. Then, we test our optimized parameters with the actual path that we previously removed as if we were receiving real life trading data. We did this with all of our optimization methods.

2.1 Pairs trading strategy:

Pairs selection was done within 2 steps as explained above, first we studied the correlation and then the cointegration between potential pairs.

First, for dimension reduction, we selected pairs from the same ETF (to ensure they belong to the same industrial sector and face the same economic and structural factors) where the correlation coefficient was higher than 0.85 by using the correlation test explained above. Then, we conducted a cointegration test on the remaining pairs using the Engle-Granger approach that utilizes the ADF test statistics. To expand on the details of the test process, the corresponding model of the Engle-Granger test can be shown as:

$$R1_t = \alpha + \beta R2_t + \varepsilon_t,\ t = 1,2,\ldots,w$$

Where $R1_t$ and $R2_t$ are the time series returns at time t, α and β are the fitted coefficients, and ε_t is the fitted residual at time t. In the process of the Engle-Granger test, we first fit the OLS regression for $R1_t$ and $R2_t$. Next, by fitting the model, we find the values of α, β and ε_t. After finding the fitted values of the coefficients, we calculate the sum of the fitted residuals in the time interval we defined, and finally run the Augmented Dickey-Fuller test on the sum of the fitted residuals to test for stationarity. These tests were completed on the preselected pairs. The final selection of stock pairs had to pass the cointegration test. Finally we ranked the stocks based on the cointegration test values. These tests were conducted on the in-sample data.

The smaller the cointegration test value was, the higher the rank the stock pair was assigned to. Final selection of the stock pairs from the top rank was used for the out-of-sample testing periods. Given the coefficients we can create a spread that has a mean-reverting property with a mean equal to 0.

The final step of the pairs trading strategy was to define a couple of trading rules. To do so, simultaneously take both long and short positions for pairs trading, we determine when we should open and when we should close the pairs trading based on quantitative definitions. We start the trading strategy with an initial wealth of $100 and invest the integrity of the wealth when opening a position.

Opening a position:
To open a position, the regression residual must exceed the opening threshold or be below the negative opening threshold.
For example when A and B are two cointegrated assets, if the residual $\varepsilon_t = A_t - \beta * B_t - \mu$ is above the opening threshold, we open a position by longing asset B and shorting asset $A * \beta$ at time t'.

Closing an opened position:
When the absolute value of the regression residual is below the closing threshold, the position was closed.
For example if previously we longed asset B, then

$$wealth_t = P_{Bt} * n_B - P_{At} * n_A$$

Where:
n_A and n_A are our current position on that asset.

Stop loss:

Furthermore, in order to prevent a significant loss if the spread starts diverging, a stop-loss was used to close the position when the regression residual hit stop loss threshold or negative stop loss threshold.

4. Recurrent Neural Network

For this project we computed a Long short-term memory (LSTM) algorithm. An LSTM is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition, speech recognition and anomaly detection in network traffic or IDS's (intrusion detection systems).

A common LSTM unit is composed of a **cell**, an **input gate**, an **output gate** and a **forget gate**. The cell remembers values over arbitrary time intervals and the three *gates* regulate the flow of information into and out of the cell. LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. This Relative insensitivity to gap length is an advantage of LSTM over RNNs, hidden Markov models and other sequence learning methods in numerous applications.

In a Traditional Neural Network, inputs and outputs are assumed to be independent of each other. However for tasks like text prediction, it would be more meaningful if the network remembered the few sentences before the word so it better understands the context. The same can be said for time series/sequential research, or predicting something cyclical in nature. RNNs overcome this problem as they have loops inside them, allowing them to have a memory of their previous computations.

Backpropagation in feedforward networks moves backward from the final error through the outputs, weights and inputs of each hidden layer, assigning those weights responsibility for a portion of the error by calculating their partial derivatives – $\partial E/\partial w$, or the relationship between their rates of change. Those derivatives are then used by our learning rule, gradient descent, to adjust the weights up or down, whichever direction decreases error.

Our aim was to train the LSTM to predict the intercept of a linear regression equation, given the ß value between the dependent and explanatory variables and these variables themselves. **Figure XX** shows a 90 days path prediction for a stock pair (PSX-VLO).

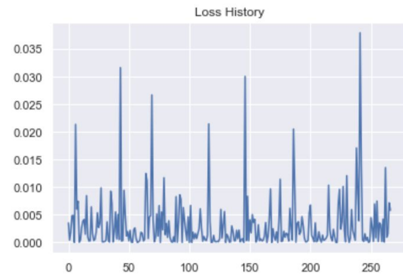


Figure 6. Loss history

Figure 7. 90 days stock prices prediction

Figure 8. Testing loss

5. Optimal Threshold Selection

5.1 Fixed Threshold

The first approach uses the standard deviation. This method relies on a constant threshold strategy. We set a constant opening threshold defined as 2 standard deviations from the in-sample spread, a closing threshold as 0.5 standard deviations from the in-sample spread, and a stop loss threshold equal to 3 times the standard deviations from the in-sample spread.

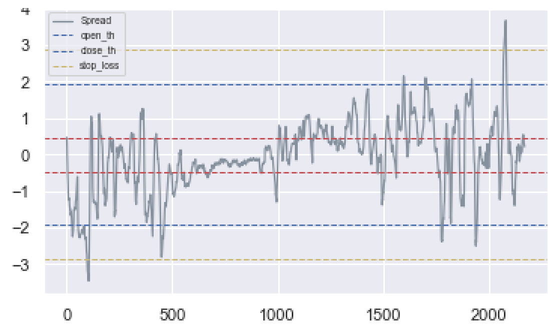


Figure 9. Fixed thresholds for the pair NKE-LOW on the train set

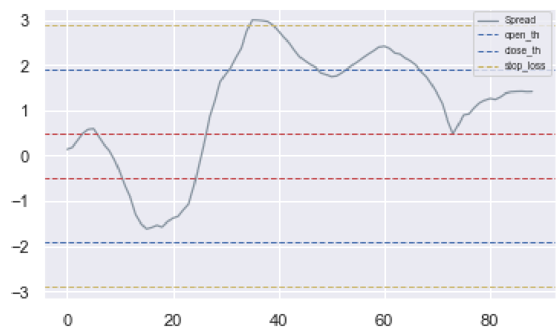


Figure 10. Fixed thresholds for the pair NKE-LOW on the test set

5.2 Spectral Analysis Threshold

In order to better capture the movements of the spread, we implemented another approach that is based on the cyclical component of the spread series, the main cycles can be identified by spectral analysis.

Assuming that the spread has a periodic behavior,we can filter the relevant frequencies that allow to smooth the spread and extract the cyclical behavior. This way we can identify the peaks and optimize the opening threshold yielding higher profits and more reliable results.
To extract the fitting, we first apply a Fourier transform and then via spectral analysis we get the relevant frequencies.

With the set of frequencies, we apply an OLS regression to find the fitting as follows:

$$spread_t = \mu + \sum_{i=1}^{N} \{\alpha_i sin(2\pi\omega_i t/252) + \delta_i cos(2\pi\omega_i t/252)\} + \varepsilon_t$$

Where μ is a constant, ε_t corresponds to the residuals and α and δ are the amplitudes for the frequency i — th.

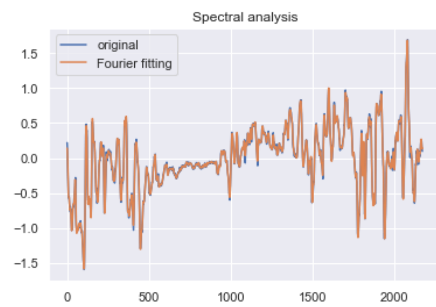


Figure 11. Fourier fitting on the spread of the pair NKE-LOW on the train set

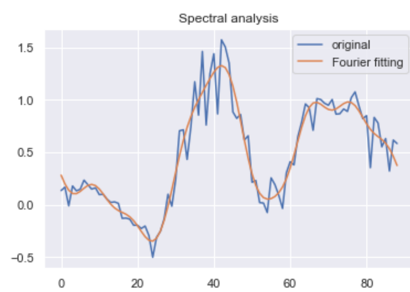


Figure 12. Fourier fitting on the spread of the pair NKE-LOW on the test set

Using the frequencies and the estimated amplitudes, α i and δ i , we can fit a fourier series on the spread and identify peaks to set opening thresholds.

For a period $t < h$, where t is the present time and h is the next maximum over the time period on the fitted series, we set the next maximum over the time range to be the upper threshold for opening positions and the next minimum over the time range to be the lower threshold for opening positions. When the series reaches this point, we open a position.

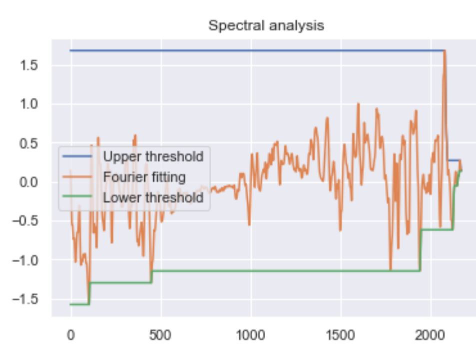


Figure 13. Dynamic thresholds for the pair NKE-LOW using spectral analysis on the train set

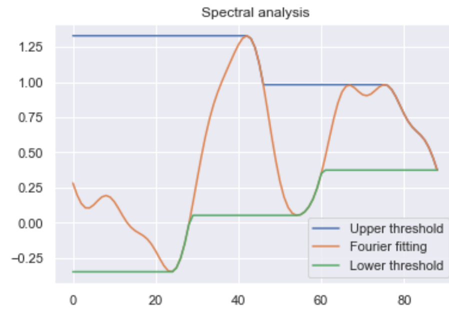


Figure 14. Dynamic thresholds for the pair NKE-LOW using spectral analysis on the test set

For the selected pair, this method has been able to adapt the threshold to match the fitted series and better extrapolate the movements of the spread.
As in the previously mentioned strategies, the position is closed when the absolute value of spread is under the closing threshold.

5.3 Bayesian optimization:

Maximizing the profit over pairs trading is a black box optimization problem where the objective function f(x) is a black box function. We do not have an analytical expression for f nor do we know its derivatives. The only information can be extracted by sampling as a point x. When f is easy to evaluate a solution is conducting a greedy search by testing multiple points via a grid search, random search or numeric gradient estimation.
Bayesian optimization solves this problem by attempting to find the global optimum in a minimum number of steps. Bayesian optimization treats the function as random and places a prior belief about f and updates the prior with samples drawn from f to get a posterior that better approximates f.

Surrogate model:
The surrogate model is used for approximating the objective function, here we use the Gaussian processes.

Acquisition functions:
Use an acquisition function (also called utility function) that directs sampling to areas where an improvement over the current best observation is likely. These functions trade off exploration and exploitation. Exploitation is conducted to sample points where the surrogate model yields higher objective values whereas exploration means sampling at points where the prediction is high. The point with the highest acquisition value is selected as the next sample.

Algorithm:
For t=1... number of steps:

1. Find the next sampling point x_t by optimizing over the acquisition function
2. Extract a noisy sample $y_t = f(x_t) + \varepsilon_t$ from the objective function f.

3. Add the sample to previous samples 1:t={1:t—1,$((x_t, y_t))$} and update the gaussian process

The bayesian algorithm picks the optimal values for the opening, closing and stop loss thresholds that are constant over the time range.

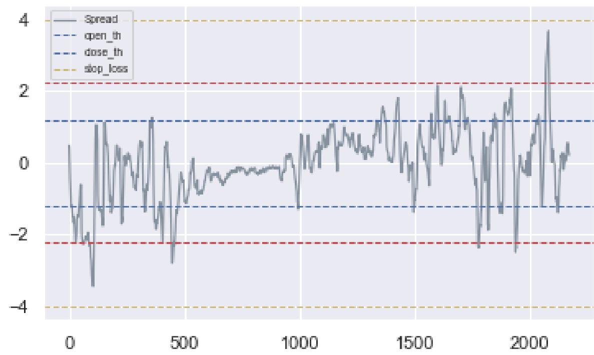


Figure 15. Optimal thresholds for the pair NKE-LOW on the train set using bayesian optimization

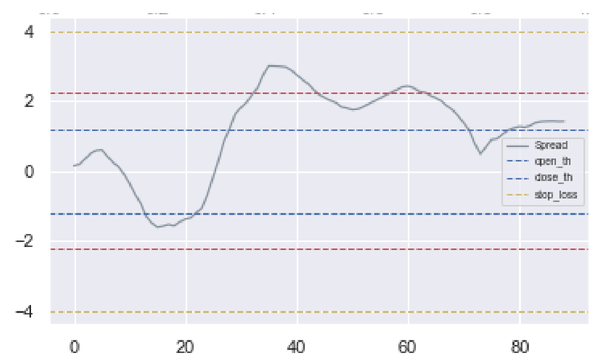


Figure 16. Optimal thresholds for the pair NKE-LOW on the test set using bayesian optimization

6. Results

6.1 Metrics:

To assess the pairs trading strategies performances, various metrics are performed.

Total return:

A Straightforward method that calculates the rate of return of a trading strategy over a given evaluation period. Total return reflects percentage of the amount invested and be calculated as:

$$total\ return = \frac{final\ wealth - intial\ wealth}{initial\ wealth}$$

$$return\ R(t) = \frac{wealth(t) - intial\ wealth}{initial\ wealth}$$

Average Return:

The average return is the average of returns over the time period.

$$Average\ return = E[R]\ where\ R\ is\ the\ return\ of\ the\ trading\ strategy$$

Sharpe Ratio:

The Sharpe ratio is used to help investors understand the return of an investment by adjusting it to its risk. The ratio is defined as the average return earned in excess of the risk-free rate divided by the volatility.

$$Sharpe\ ratio = \frac{E[R-R_f]}{\sqrt{Var(R)}}\ where\ R\ is\ the\ return\ of\ the\ trading\ strategy,\ Rf\ the\ risk\ free\ rate$$

Volatility:

Volatility is a statistical measure of the riks, it measures the dispersion of returns for a given security or market index. In most cases, the higher the volatility, the riskier the security.

$$Volatility = \sqrt{Var(R)},\ R\ is\ the\ return\ of\ the\ trading\ strategy$$

Max Drawdown:

A maximum drawdown (MDD) is the maximum observed loss from a peak to a trough of a portfolio, before a new peak is attained. Maximum drawdown is an indicator of downside risk over a specified time period.

$$MDD = \frac{peak - trough}{peak}$$

6.2 Pairs trading and S&P 500

Performance of different optimizations:

	Average return	Sharpe ratio	Volatility
Fixed threshold	3.6%	0.4	1.5%
Spectral analysis	18.4%	1.8	8.5%
Bayesian optimization	7.8%	0.9	5.3%
S&P 500	9.8%	1.0	14.7%

7. Conclusion

The first part of this project focuses more on the theories behind the pairs trading strategy using cointegration modeling. We showed that if we could properly select stock pairs that are exposed to the same risk factors, there exists a correlation between their prices (log prices). Even further, cointegration relationships can be shown, and are advantageous for us to create a trading strategy that outputs excess return. Also, demonstrate the advantage we can gain from implementing a machine learning algorithm. Moreover, combining the fact that cointegration test says if the linear combination of a pair of stocks is stationary, we can be confident that this property will be true also in the future and then predict the time series based on this fact.

Regarding the optimization algorithms, we get very good results from both. Spectral Analysis outputs better results because it has a dynamic threshold and, given the accurate predictions that the RNN computed, it adapted way better to the market movements. Bayesian Optimization is closer to a “smart” brute force algorithm for black box functions, but given that we assume stationarity of the path, we can argue that future behavior will be similar to past behavior.

We can also conclude that this strategy is profitable due to some market’s inefficiencies. This means that despite what the theory indicates, the price does not reflect all the information at every moment. We cannot conclude that this strategy will work for every stock. It needs high volatility in order to capture the mispricing of the stocks and the potential profit. But at the same time, higher volatility comes with higher risk, so there is a trade off that has to be done for all the pair traders.

For future work we would like to expand the universe, because right now we are only analyzing large cap stocks, and maybe more volatility can be captured in stocks with smaller capitalization. Also, it would be very beneficial for a project with this scope to rely on a higher computational power, because when we wanted to be more ambitious in the length of the prediction, we ran out of computational memory.

References:

Jiayu Wu, *A Pairs Trading Strategy for GOOG/GOOGL Using Machine Learning*, December 9, 2015

Ziping Zhao, Zhongju Wang, and Daniel P. Palomar ; *Optimal Portfolio Design for Statistical Arbitrage in Finance*, March 2018

Alejandro Álvarez Roa, *PAIRS TRADING: OPTIMAL THRESHOLD STRATEGIES*, June 2018

George J. Miao, *High Frequency and Dynamic Pairs Trading Based on Statistical Arbitrage Using a Two-Stage Correlation and Cointegration Approach*, February 2014

Fabozzi, Frank J. *The Basics of Financial Econometrics: Tools, Concepts, and Asset Management Applications*. Wiley, 2014.

Krauss, Christopher, 2015. "*Statistical arbitrage pairs trading strategies: Review and outlook*," FAU Discussion Papers in Economics 09/2015, Friedrich-Alexander University Erlangen-Nuremberg, Institute for Economics.

Pole, Andrew. *Statistical Arbitrage: Algorithmic Trading Insights and Techniques*. J. Wiley & Sons, 2007.

Ruppert, David. *Statistics and Data Analysis for Financial Engineering: with R Examples*. Springer, 2015.