Pairs Trading With Statistical and Machine Learning Methods

1 Abstract

The paper discusses the use of statistical and machine learning methods for generating eligible pairs for pairs trading, a popular trading strategy. Pairs trading involves profiting from deviations in prices or returns of two assets that converge to their mean in the long run, typically based on mean reversion and stationary stochastic processes. However, relying on correlation or co-integration assumes linearity, which may not always hold. Thus, non-linear techniques like copulas or machine learning are gaining attention. The statistical methods presented include co-integration and Hurst exponent analysis to shortlist pairs that exhibited mean reverting spreads. The machine learning methods presented include using Cluster Analysis to congregate analogous ETFs. The paper also discusses different trading strategies that can be used on eligible ETF pairs. It involves comparing novel trading approaches based on statistical Copulas and Reinforcement Learning agent to trade pairs and evaluating their performance against a benchmark of baseline strategy based on Bollinger bands.

2 Introduction

Pairs trading is a popular strategy in financial markets where a long position in one stock is taken while simultaneously taking a short position in another highly correlated stock. In this paper, we evaluate the effectiveness of pairs trading using both statistical and machine learning methods.

In statistical pairs trading, there are two moving parts or areas that need a precise understanding of the two securities, their price & return time-series, the distribution of the spread between them, and higher order moments of all of these. This is where we try to replace traditional statistical methods for selecting pairs and trading their spread using Copula and Machine Learning methods. Copulas have the ability to trade pairs when their spreads have varying distributions. Machine Learning and Reinforcement learning is used to mine patterns not visible to traditional statistics and continuously adapt trading strategies to market conditions instead of static trading rules.

3 Data Preparation

The research paper focuses on a universe of 105 equity-based ETFs, categorized according to the type of commodities being tracked. Daily frequency price series data was retrieved for each ETF, and missing value ETFs were discarded. The minimum liquidity requisites were used to remove ETFs that did not meet the criteria, ensuring the bid-ask spread transaction costs were realistic. Occasional outliers were detected in some price series and were analyzed using a simpler technique that involved calculating the return series and identifying points with percentage changes higher than 10 percent. Finally, the data was standardized by dividing each spread by its mean and standard deviation when training the forecasting-based models.

For each algorithm and strategy, we used data from '2011-01-01' to '2016-12-31' as training data and testing dates were from '2017-01-01' to '2019-12-31'.

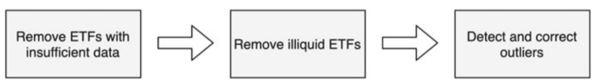

Figure 3.1 Data Processing Steps

4 Pairs selection

Pairs trading is a trading strategy in which a pair of securities are traded together based on the statistical relationship between them. To generate eligible pairs for pairs trading, various statistical and machine learning methods have been developed.

One of the statistical methods used to generate eligible pairs is cointegration analysis. This method aims to identify pairs of securities that have a long-term relationship, which implies that any deviation from this relationship will eventually be corrected. Therefore, two securities are considered eligible for pairs trading if they are cointegrated.

Another statistical method used to generate eligible pairs is the Hurst exponent analysis. The Hurst exponent is a measure of the long-term memory of a time series. In pairs trading, the spread between two securities is analyzed to determine whether it exhibits mean reversion behavior, which is a necessary condition for pairs trading. A spread that has a Hurst exponent of less than 0.5 is indicative of mean reversion, making it a suitable candidate for pairs trading.

Additionally, the time period in which the spread between the two securities diverges and converges is an important consideration. If the divergence and convergence periods are too long, it may not be practical to trade the pair. Therefore, pairs that exhibit divergences and convergences within reasonable time periods are considered eligible for pairs trading.

Finally, it is essential to ensure that the spread between the two securities reverts to the mean with sufficient frequency. A pair that reverts to the mean infrequently may not be profitable for pairs trading.

In addition to statistical methods, machine learning methods have also been developed to generate eligible pairs. One such method is K means clustering, which aims to cluster similar securities together based on their price behavior. Hierarchical clustering is another method that aims to group securities into clusters based on their similarities. Affinity propagation is a third machine learning method that can be used to generate eligible pairs for pairs trading.

Overall, by using statistical and machine learning methods, traders can identify pairs of securities that exhibit the necessary conditions for pairs trading and increase their chances of success in this trading strategy.

4.1 Statistical method

We perform a battery of statistical tests to shortlist a group of eligible pairs as shown below:

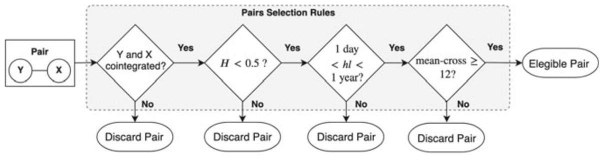

Figure 4.1.1 Pair Selection Criteria

Mean reversion and Stationarity

To fully understand the formalities of pairs trading, it's important to introduce some tools. Stationarity is a key concept in this regard, with a stochastic process being considered stationary if its mean and variance remain constant over time. A stationary time series is mean reverting in nature, with fluctuations around the mean having similar amplitudes. The order of integration, I(d), reports the minimum number of differences required to obtain a stationary series. While financial traders often take advantage of the stationary property by placing orders when the price of a security deviates considerably from its historical mean, non-stationary price series are more common and theoretically unpredictable. Pairs trading is interesting because it creates an artificial stationary time series from the combination of two non-stationary time series. It's worth noting that mean-reversion and stationarity are distinct mathematical concepts, with the former describing the change in a time series being proportional to the difference between the mean and current value, and the latter indicating that the variance of the logarithmic value of the time series increases slower than a geometric random walk. Various tools for detecting and modeling these properties will now be presented.

Augmented Dickey Fuller test

The ADF test, proposed by Dickey and Fuller, is a hypothesis test used to determine if a time series contains a unit root. The null hypothesis assumes the presence of a unit root, while the alternative hypothesis assumes stationarity. The ADF test models changes in the time series as an autoregressive process with lag order k. The test evaluates whether the coefficient λ is equal to zero, which would indicate that the series is a simple random walk. A negative value of λ (calculated with $\Delta y(t-1)$ as the independent variable and $y(t)$ as the dependent variable) suggests mean reversion. The test statistic is calculated as λ divided by the standard error (SE) of the regression fit. The critical values corresponding to the distribution of the test statistic can be found tabulated and used to determine whether to accept or reject the null hypothesis at a given probability level. A consecutive change in time series can be modeled as:

$$\Delta y(t) = \lambda y(t-1) + \mu + \beta t + \alpha_1 \Delta y(t-1) + \cdots + \alpha_k \Delta y(t-k) + \epsilon_t$$

where $\Delta y(t) \equiv y(t) - y(t-1), \mu$ is a constant, β is the coefficient on a time trend and k is the lag order of the autoregressive process.

Cointegration-Engle and Granger test

Cointegration was initially introduced by two econometricians, Engle and Granger. It refers to a set of variables that are cointegrated if there exists a linear combination of those variables, of order d, which results in a lower order of integration, I(d-1). Formally, considering two time series, y_t and x_t which are both I(1), cointegration implies there exist coefficients, and such that

$$y_t - \beta x_t = u_t + \mu \tag{1}$$

The cointegration method involves selecting pairs of securities that are cointegrated, meaning they have a stable long-term relationship. If two securities, Y_t and X_t, are found to be cointegrated, then the resulting series from their linear combination, $S_t = Y_t - X_t$, where is the cointegration factor, must be stationary. Defining the spread series in this way is convenient because it is expected to be mean-reverting. The cointegration approach is considered more rigorous than the distance approach for pairs selection because it identifies more sound equilibrium relationships. The selection criteria for cointegration pairs has been proposed by Vidyamurthy, who also provides heuristics for cointegration strategies. A comparison study by Huck and Afawubo using SP 500 constituents under varying parameterizations, including risk loadings and transaction costs, found that the cointegration approach significantly outperformed the distance method, supporting the hypothesis that the cointegration approach identifies more robust relationships.

Hurst Exponent

The Hurst exponent is a useful metric for assessing the stationarity of a time series by measuring its speed of diffusion from its initial value compared to that of a geometric random walk. This can be quantified by the variance of the logarithmic time series value at time t and an arbitrary time lag , denoted as Var(). The geometric random walk standard speed can be approximated as

$$\left\langle |z(t+\tau) - z(t)|^2 \right\rangle \sim \tau^{2H}$$

where H represents the Hurst exponent. For a price series exhibiting a geometric random walk, H = 0.5, resulting in

$$\left\langle |z(t+\tau) - z(t)|^2 \right\rangle \sim \tau^{2H}$$

As H decreases toward zero, the speed of diffusion reduces, indicating a more mean-reverting price series, while an increasing H value toward 1 suggests an increasingly trending price series. Therefore, the Hurst exponent serves as an indicator of the degree of mean-reversion or trendiness in a time series.

Half life

The half-life of mean-reversion is a useful metric that measures the time it takes for a time series to return to its mean value. To calculate this metric, the discrete-time series from equation below-

$$\Delta y(t) = \lambda y(t-1) + \mu + \beta t + \alpha_1 \Delta y(t-1) + \cdots + \alpha_k \Delta y(t-k) + \epsilon_t$$

can be transformed into its differential form, which expresses the changes in prices as infinitesimal quantities. By ignoring the constant drift in price and the lagged differences in the above equation, the expression can be simplified into equation-

$$dy(t) = (\lambda y(t-1) + \mu)dt + d\varepsilon$$

which describes an Ornstein-Uhlenbeck process with some Gaussian noise ($d\varepsilon$). Using the analytical solution presented in equation -

$$E(y(t)) = y_0 \exp(\lambda t) - \mu/\lambda(1 - \exp(\lambda t)).$$

we can determine that if λ is positive, the time series will not exhibit mean-reversion. However, if λ is negative, the expected value of the time series decays exponentially to the value of $-\frac{\mu}{\lambda}$, and the half-life of decay is $\frac{\log(2)}{\lambda}$. This result implies that a larger absolute value of λ is associated with a faster mean-reversion. Therefore, $\frac{\log(2)}{\lambda}$ can be used as a natural time scale to define many parameters in a trading strategy and avoid brute-force optimization of a free parameter.

Mean-crosses

To ensure sufficient liquidity, we require that every spread crosses its mean at least once per month. It's important to recognize that while there's a negative correlation between the number of mean crosses and the half-life period, as shorter half-lives naturally result in more mean crosses, these properties aren't always interchangeable. Imposing this constraint not only enforces the previous condition, but it may also eliminate pairs that, despite meeting the mean-reversion timing requirements, fail to cross the mean, leaving no exit opportunities. This constraint ensures that our trading strategy is both profitable and practical.

4.2 Machine Learning methods

The statistical methods mentioned in the preceding subsection are meant to identify pairs which follow some mathematical rules and can be traded to create significant profits that outshine any costs and risks associated with it. But this is under certain assumptions, that the spread between co-integrated securities is stationary, it log-returns are distributed normally, and that the spread is almost a mean-reverting process with considerable variance to provide trading windows. This might not be the case always and there can be some securities that provide pair trading opportunities without following these assumptions.

In this subsection, we use Cluster Analysis to congregate securities with analogous trends and risk that would be eligible for a prospective pairs trading strategy. Cluster analysis is an example of an unsupervised learning algorithm, which means that the number of clusters in the data is not known before running the model. Unlike our statistical methods, cluster analysis does not assume any pre-existing relationships within the data. Rather, it identifies associations and patterns in the data without providing information about their meaning or interpretation. Thus it might be capable of capturing relationships from higher order moments that certain statistical methods might be unable to do analytically.

We try three different types of clustering algorithms, each different in the fundamental method used for creating clusters.

K-Means Clustering

K-means clustering is a technique used for grouping data points into K clusters, where K is a predefined number of clusters. The algorithm works by iteratively assigning data points to the closest cluster centroid and updating the centroid until the algorithm converges. K-means clustering has been widely used in finance for various purposes, including identifying patterns in stock returns, clustering mutual funds, and grouping stocks into portfolios.

In the context of pairs trading, this algorithm would use price data to group securities that have similar price patterns into clusters. Once the securities are grouped into clusters, pairs can be selected by finding securities that are in the same cluster, as their price behavior would be similar.

Hierarchical Clustering

Hierarchical clustering involves grouping similar securities into clusters based on their price behavior and other characteristics. Hierarchical clustering has the advantage of providing a dynamic termination criteria, which allows investors to select a group displacement with the desired level of granularity. Hierarchical clustering is based on the top-down approach as opposed to the bottom-up approach of K-mean clustering.

This algorithm involves grouping data points or objects it is meant to cluster using tree like relationships which can produce sub-optimal results and create high bias in certain scenarios.

Affinity Propogation Clustering

Affinity Propagation Clustering is a type of clustering method used in machine learning that does not require a specified number of clusters in advance. Instead, it creates clusters by measuring how well one instance suits another as a representative. The process is similar to instances messaging each other on how much they suit one another, and a sender will send back the revised value of attractiveness to each receiver until an agreement is reached.

Once a sender gets associated with a receiver, the receiver becomes the exemplar, and all data points with the same exemplar form a cluster. This method has the advantage of not requiring a predefined number of clusters, and it can identify exemplars that represent the most important data points in a cluster. Additionally, this method is suitable for data sets with a high dimensionality, which can make it difficult to visualize and analyze the data in a traditional sense.

Comparing Clustering algorithms

We tried to create clusters of ETFs in our data universe based on their mean returns and volatility throughout our training and testing periods using all the above-mentioned clustering methods. We measure the quality of clusters identified by each algorithm using the Silhouette scores for labels created by the models. The K-means and hierarchical clustering methods need a preset number of clusters to be defined. We use the elbow method and the dendogram graph for determining the optimal k number of clusters for K-means and Hierarchical algorithms respectively. The affinity propagation method is self-sufficient and does not require a preset number of clusters. In our case, we get 4 clusters from K-means, 3 clusters from Hierarchical clustering and 11 clusters from the affinity propagation method.

The Silhouette scores for the three methods are as follows:

Figure 4.2.1 K-Means Clustering: 0.59

K-Means works best in this case because of a strong euclidean similarity between some ETFs' mean returns and volatility.

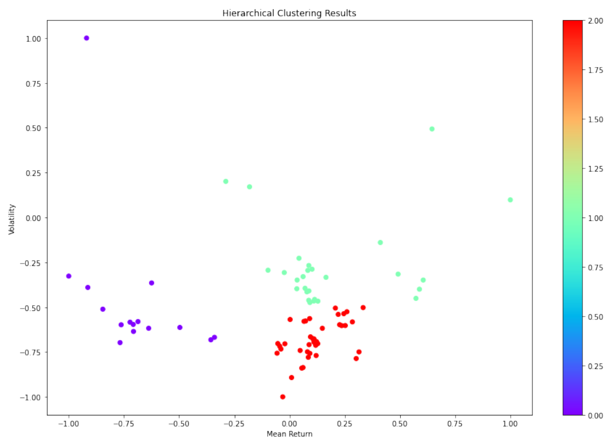

Figure 4.2.2 Hierarchical Clustering: 0.42

Hierarchical Clustering does not perform well relative to the K-means algorithm as the tree-based approach seems to induce great bias.

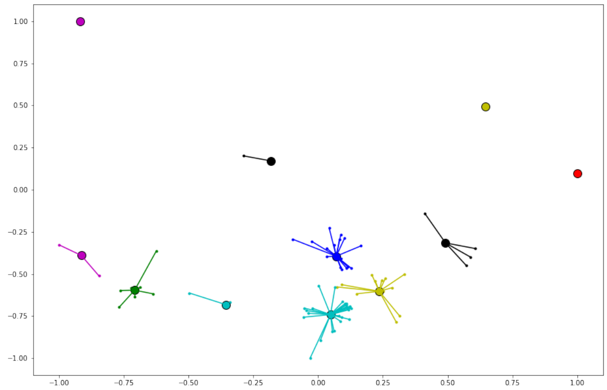

Figure 4.2.3 Affinity Propagation Clustering: 0.49

Also, the Affinity Propagation methods cannot match K-means simplicity as it works best with high-dimensional data.

5 Trading setup

This section given an overview of the three 3 trading strategies we have used, namely, a cointegration based baseline strategy which serves as a benchmark to our copula and machine learning based trading strategies.

5.1 Baseline Strategy

For the purpose of this paper, we implement a mean-reverting pairs trading strategy that takes advantage of the co-integration relationship between the two stocks to act as a benchmark to copula and machine learning based trading strategies. If the spread between the two stocks deviates from the fair value, the strategy takes a position to profit from the expected reversion to the mean.

This strategy is based on the assumption that over the long-term, the two stocks in a co-integrated pair will move together and maintain a certain relationship to each other, such that any divergence from this relationship will eventually revert to the mean.

5.2 Copula Strategy

To implement this strategy, the relevant copula function, marginal distributions, and conditional probability distribution functions are required, which may involve the use of copulas. During the formation period, data of the stocks is utilized to estimate the marginal distribution functions and their respective parameters based on the value of the cumulative log-returns. This estimation can be done by standard statistical analysis software to determine the best-fitting marginal distribution. Once the marginal distributions and their parameters are estimated for each stock return, the cumulative distribution function values obtained for each stock, u and v, are utilized to choose a relevant copula function. The final important information required for this strategy is the conditional probability functions. These functions are defined as the derivatives of the copula with respect to v and u, respectively, and are used to calculate the following conditional probabilities

$$P(U \leq u \mid V = v) := \frac{\partial C(u, v)}{\partial v},$$

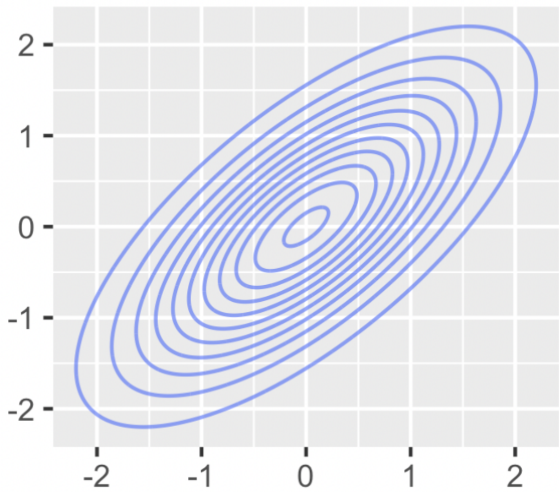

Figure 5.2.1 Gaussian Copula Contour Plot

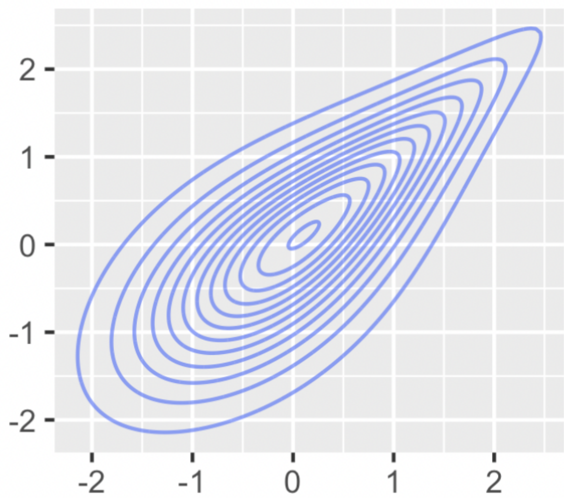

Figure 5.2.2 Gumbel Copula Contour Plot

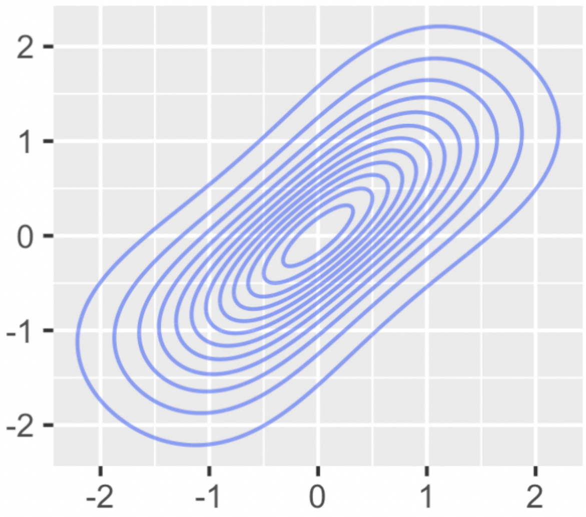

Figure 5.2.2 Frank Copula Contour Plot

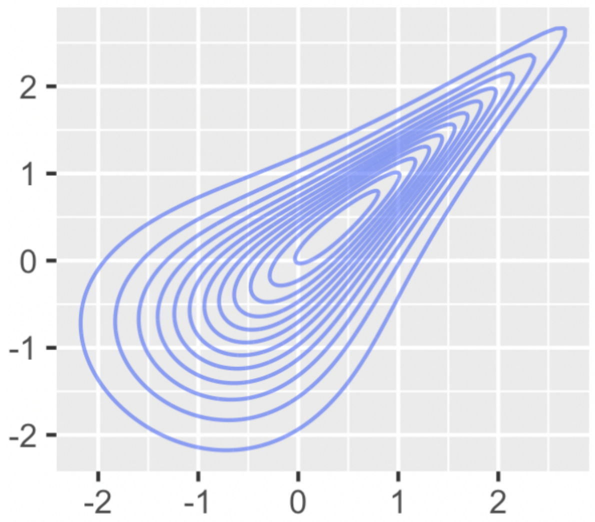

Figure 5.2.2 Joe Copula Contour Plot

5.3 Reinforcement Learning Trading Agent

Reinforcement learning is a type of machine learning in which an agent learns to make decisions based on a sequence of experiences and feedback from its environment. The goal of reinforcement learning is to find the optimal policy or set of actions that maximizes a long-term reward.

In our scenario, reinforcement learning is used to optimize trading strategies by learning from past experiences and feedback. The agent in reinforcement learning represents the trading algorithm, and the environment represents the market. The agent takes actions, such as going long or short on the spread between two ETFs, and receives rewards based on the profitability of those actions. We create a hidden Markov State Space with seven features that is updated using a Neural Network to determine the optimal policy for trading the spread between ETF pairs. This method is capable to recognizing and adapting to market changes, which could result in alternate optimal trading strategies.

Another advantage of reinforcement learning in pairs trading is its ability to explore new trading strategies. The agent can try different actions and observe their outcomes, allowing it to discover new and potentially profitable strategies. We primarily leverage Deep Q-Networks to train our trading agent.

Deep Q networks

Deep Q-Networks (DQNs) are neural networks meant to approximate the Q-value function which represents the expected future reward of taking a certain action in a given state. For each input state, the neural network outputs Q-values for all possible actions. The agent then chooses the action with the highest Q-value.

In pairs trading, DQNs can be used to optimize trading strategies by learning from past experiences and feedback, similar to other reinforcement learning algorithms. The DQN is meant to reward the agent on a successful action and penalize on a negative action. It uses two neural networks, namely Eval net and Target net to train the agent in which the Eval net estimates a policy, validates it on the Traget net and back-propagates the error to update state change probabilities.

6 Executing Trades

6.1 Baseline Strategy Execution

To identify the fair value of the spread, we use a linear regression model to estimate the parameters of a cointegrating relationship between the two stocks. The model then predicts the fair value of the spread, based on the current price of the first stock given by:

$$fairvalue = \hat{a} + \hat{b}x_t$$

If the actual spread diverges from the fair value by a certain threshold, the strategy takes a position to profit from the expected reversion to the mean.

By taking a long position in the stock that is expected to increase in price and a short position in the stock that is expected to decrease in price, the strategy aims to profit from the convergence of the spread back to its fair value.

6.2 Copula Based Strategy Execution

We begin performing a distribution fitting analysis on the returns of a set of selected stocks. We use four different probability distributions, namely Normal, Student's t, Logistic, and Extreme, to fit each stock's returns distribution and select the best fit.

We compute Akaike Information Criterion score(measure of the relative quality of different models), Bayesian Information Criterion(the Bayesian Information Criterion score for the best-fit distribution, which is similar to AIC but applies a different penalty for model complexity) score for the best-fit distribution, Kolmogorov-Smirnov test's p-value (test for goodness of fit, which assesses how well the fitted distribution matches the empirical data.) for fit analysis.

Then, we iterate over the four distributions and fit each of them to the stock's returns using the maximum likelihood method. The log-likelihood is used to compute the AIC and BIC scores for each distribution fit, and the distribution with the lowest AIC score is chosen as the best fit for the stock.

Once we have the marginal distributions, we calculate the cumulative distribution function values for each stock pair using the marginal distribution functions and apply the probability integral transform.

We then fits four different copula models (Gaussian, Clayton, Gumbel, Frank, Joe) to the transformed data and calculates the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) for each copula model. The copula model with the lowest AIC is selected as the best fitting copula model for that pair of stocks.

Finally, we calculate the KS-pvalue, which measures the goodness of fit between the copula model and the transformed data. Then, using the fitted copula, we calculate the conditional probabilities of observing certain values for one stock given the value of the other.

Based on these conditional probabilities, the strategy decides whether to open a long, short or no position. If both conditional probabilities are outside a certain range (in our case 70 basis points), the strategy opens a position with equal weighting in each stock in the pair. If one of the conditional probabilities falls inside the range, we close the position in that stock. If both conditional probabilities fall outside the range, the code does nothing and holds the existing position.

6.3 Reinforcement Learning Trading Agent Execution

This system trains and tests a DQN on 94 ETF pairs. We generate pairs from the top three clusters generated by the K-means algorithm, test each pair's variance to make sure it is tradeable and then pass them on to the DQN. We do this test by dividing each ETF's volatility by its mean returns and making sure to trade pairs for which both ETF's have these values to be more than 0.5.

We generate 10 features from daily ETF closing price data of the pair. These features are the spread between the two ETFs, change in spread from previous day, and multiple moving averages for the spread value and returns. These features represent the state of the trading environment as a hidden Markov state space in Open-AI gym environment's step function. After receiving the current state of features, the DQN outputs an action which is long, short or hold on the spread. The DQN keeps repeating this process to update its initial weights and maximize its reward. We also use a negative multiplier to heavily penalize incorrect actions. The DQN utilizes RELU non-linear activation functions and an Adam optimizer to update network weights.

The rewards in the OPENAI Gym environment are calculated as follows:

Training Rewards = Action $\times$ $SpreadReturns \times NegativeReturnsMultiplier$

Testing Rewards = Action $\times SpreadReturns$

The DQN utilizes two Pytorch Neural Networks consisting of an input layer of 10 features, a fully connected layer of 50 nodes, another fully connected layer of 50 nodes, utilizing a RELU non-linear activation function, and an output layer of 3 nodes.

7 Results

For comparing results of our different trading startegies, we use the following prevalent metrics:

Returns

We use simple arithmetic mean as a metric as it is widely used by everyone.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Sharpe Ratio

Sharpe Ratio is a commonly used metric in finance to measure the performance of an investment. It is a risk-adjusted measure that takes into account the average return earned in excess of the risk-free rate per unit of risk. In pairs trading, the Sharpe ratio is used to measure the excess return earned by the trading strategy relative to the risk taken.

$$SharpeRatio = \frac{E[R_p - R_f]}{\sqrt{Var(R_p - R_f)}}$$

where:

$E[R_p - R_f]$ is the expected excess return of the investment over the risk-free rate R_f.

$Var(R_p - R_f)$ is the variance of the excess return of the investment over the risk-free rate R_f.

The Sharpe ratio is calculated as the ratio of the excess return of the trading strategy over the standard deviation of the excess returns. The excess return is the return earned by the trading strategy over the risk-free rate. The standard deviation of the excess returns is a measure of the risk or volatility of the trading strategy.

Maximum Drawdown

Maximum drawdown is a risk metric used in finance and trading to evaluate the largest peak-to-trough decline in the value of an investment or portfolio over a specific period. It measures the maximum loss from the peak value of an investment or portfolio to its trough value. Maximum drawdown is important for investors and traders because it indicates the maximum amount of capital that could have been lost if an investment was held throughout its period of decline.

The formula for maximum drawdown is: Maximum Drawdown = $\max_{i,j:i<j}\left(\frac{V_i - V_j}{V_i}\right)$

where:

V_i is the value of the investment at time i V_j is the value of the investment at time j $\max_{i,j:i<j}$ denotes the maximum value taken over all i, j pairs where $i < j$. In other words, maximum drawdown is the maximum percentage drop in the value of an investment or portfolio from a previous peak to a subsequent trough.

7.1 Evaluating Results

For evaluating the performance of our pair-selection models and trading strategies, we compare them with benchmarks statistical methods and a baseline strategy respectively. The performance is evaluated by running the trading strategies for all pairs picked by statistical & ML methods and measuring average % returns, sharpe ratio and max drawdown across pairs.

As we can see in the tables below, the numbers performance of all the trading strategies is significantly better for the pairs picked by Cluster Analysis than by Statistical Methods. This tells us that clustering is able to mine patterns of higher order moments which traditional statistical methods might be unable to do.

When we compare the performance of trading strategies for a given set of 98 ETF pairs, the Copula based strategy does a decent job of defeating the baseline algorithm. But we see significant gains in the Reinforcement strategy proving that it had ability to adapt better to market and spread changes throughout the three years of testing period.

Trading Strategy	Returns (%)	Sharpe Ratio	MaxDD
Baseline	-0.09%	-0.0009	-0.0319
Copula	-0.22%	0.1818	-0.0267
Reinforcement Learning	5.85%	-1.3037	-1.158

Table 7.1.1 Results for Pairs picked by Statistical Methods

Trading Strategy	Returns (%)	Sharpe Ratio	MaxDD
Baseline	0.2388%	0.0006	-0.1213
Copula	1.5190%	0.4054	-0.0168
Reinforcement Learning	15.65%	1.489	-0.0802

Table 7.1.2 Results for Pairs picked by Cluster Analysis

8 Conclusion

In this paper, we have tried to implement pairs trading strategies on ETFs instead of traditionally using stocks. In our research and experiments, we tried novel approaches for selecting opportunistic pair of ETFs and building dynamic trading strategies for trading those ETF pairs. From evaluating the results, the ML method using Cluster Analysis is clearly better than the statistical method involving co-integration, stationarity and normality tests. This shows us that there are opportunities for pairs trading even when pair spreads don't fall into exact simple geometrical distributions.

We see how the Copula strategy is clearly an upgrade to the baseline bollinger-band trading strategy as it is capable of handling various possible distributions of the mean-reverting spread.

But the highlight of our work is the Reinforcement Learning Trading agent trained on a Deep Q-Network which seems capable of quickly adapting to spread trends and volatility and can even trade when the spread doesn't exactly follow a mean reverting process but follows a significant trendline.

9 References

Gatev, E., Goetzmann, W. N., Rouwenhorst, K. G. (2006). Pairs trading: performance of a relative-value arbitrage rule. *The Review of Financial Studies*, 19(3), 797-827.

Engle, R. F., Granger, C. W. (1987). Co-integration and error correction: representation, estimation, and testing. *Econometrica: Journal of the Econometric Society*, 55(2), 251-276.

Chen, Y. F., Huang, C. L. (2017). Application of machine learning algorithms to pairs trading strategy. *Journal of Finance and Economics*, 43(3), 73-84.

Avellaneda, M., Lee, J. H. (2010). Statistical arbitrage in the US equities market. *Quantitative Finance*, 10(7), 761-782. Chan, W. H., Chui, C. W. (2010). An empirical comparison of moving average envelopes and Bollinger bands in stock market. *Investment Management and Financial Innovations*, 7(3), 17-24.

Hastie, T., Tibshirani, R., Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer.

Hong, J., Lee, C. J., Li, Y. (2019). Exploring the effectiveness of machine learning for pairs trading. *Expert Systems with Applications*, 137, 300-313.

Hu, J., Chen, J., Wu, Y. (2018). Combining machine learning with pairs trading strategy in Chinese stock market. *Journal of Intelligent Fuzzy Systems*, 35(4), 4645-4654.

Khandani, A. E., Lo, A. W. (2007). What happened to the quants in August 2007? *Journal of Investment Management*, 5(4), 5-18.

Zhang, G., Patuwo, B. E., Hu, M. Y. (1998). Forecasting with artificial neural networks: the state of the art. *International Journal of Forecasting*, 14(1), 35-62.