Application of Time-Varying Optimal Copula and Mixed Copula in Pairs Trading

Common practice in Pairs Trading such as Distance, Cointegration, and Stochastic Control approaches are based on an assumption that the relationship between the 2 assets is linear. In reality, such dependence structure is usually nonlinear, especially in the extreme scenarios where the relationship could be drastically different. Copula is one of the most recent techniques which could model the nonlinear dependency. However, transforming inputs into stationary processes, choosing the correct distributions for each asset, and selecting the best type of Copula are the greatest challenges for this approach, especially when it is a known fact in finance that volatility of returns is not constant. Therefore, in this project we aim to apply GARCH Filter, Mixed Copula, and Copula Goodness of Fit Test to improve pairs trading strategies.

1 Introduction

Pairs Trading involves trading opposite positions for 2 co-moving securities and exploiting profit by betting that the "spread" between the two will eventually converge. Most common approaches, including distance and cointegration approaches involve assuming a linear relationship between the two assets.

Linear Relationship assumes that the relationship of 2 co-moving securities can be expressed linearly. Supposed X and Y are co-moving securities, the relationship of the prices or returns between them can be expressed as the following linear equation.

$$Y_t = \alpha X_t + \varepsilon_t$$

The spread can be defined as the following.

$$\varepsilon_t = Y_t - \alpha X_t$$

Through the linearity, where α is the coefficient, one can trade the spread by exploring ε_t. In distance approach, α is equal to 1 and the spread is simply Y_t-X_t. In cointegration approach, α is calculated based on the cointegration coefficient from several tests, such as Johansen test.

In reality, the relationship between the prices or returns of 2 securities is not linear. During extreme events when prices plummet or rocket within a short period of time, the dependence structure between the two would not be the same as that in normal circumstances. This is why the Copula approach is now adopted widely in pairs trading.

There is another alternative which is the machine learning approach. [4]However, it is based on utilizing the fact that the spread between 2 cointegrated assets is a stationary process. Due to this, applying time series forecasting techniques such as LSTM to forecast the spread movement is possible. In other words, the method assumes a linear relationship to construct a portfolio. The forecasting part aims toward solving a lack of signals in cointegration approach.

In contrast to machine learning, Copula directly captures the nonlinearity of the market by modeling the joint cumulative probability function. Instead of modeling the spread, Copula separately generates signals, conditional probability in this case, for each leg. Thus, this eliminates the need for linear assumption.

However, there are 3 main challenges in modeling bivariate Copula:

1. Stationary Assumption

Copula requires both input processes to be stationary. Most common practice to deal with this problem is to use quantiles of prices or use returns as inputs. In this paper, we chose the latter one. Nevertheless, it is an accepted fact in finance that the volatility of returns is not constant over time. Due to this, we apply GARCH Filter to the log return data to eliminate the nonconstant variance and mean.

2. Marginal Distribution Determination

The distribution for each leg needs to be determined as a separate procedure. Assuming symmetric distribution might not always be realistic. Thus, we chose to use the empirical cumulative distribution function fitted during the formation period as our assumed marginal distribution.

3. Optimal Copula Selection

The model performance largely depends on selecting the correct type of Copula.

2 Signal Generation

2.1 GARCH Filter

There is an existing research on time-varying Copula (Liu et al., 2016) which could directly be applied to satisfy the stationary requirement. Its concept is based on the fact that the cumulative distribution function of returns is equivalent to that of the standardized residuals obtained from GARCH model. [3]

$$F_{r_{i,t}}\left(r \mid \Omega_{t-1}\right) = \Pr\left(r_{i,t} \le r \mid \Omega_{t-1}\right)$$

$$= \Pr\left(\varepsilon_{i,t} \le \frac{r-\mu_i}{\sqrt{h_{i,t}}} \mid \Omega_{t-1}\right) = F_{\varepsilon_i}\left(\frac{r-\mu_i}{\sqrt{h_{i,t}}}\right)$$

GARCH model is constructed as follows.

$$\begin{aligned} r_{i,t} &= \mu_i + a_{i,t}, i \in \{X, Y\} \\ a_{i,t} &= \sqrt{h_{i,t}} \cdot \varepsilon_{i,t}, \varepsilon_{i,t} \sim \text{ i.i.d. } F_{\varepsilon_i} \\ h_{i,t} &= \omega_i + \beta_i \cdot h_{i,t-1} + \alpha_i \cdot a_{i,t-1}^2. \end{aligned}$$

The GARCH model is implemented as a rolling window forecast. We fitted the model on 75 data points (approximately 1 trading quarter) and forecast 1 step ahead. We fit the Copula on the standarized residuals $\varepsilon_{i,t}$.

2.2 Single Copula Estimation and Selector

Copula estimation consists of two steps: Marginal Distribution Fitting, and Copula Fitting. We choose to use the nonparametric estimation method to model the marginals, which is closely related to empirical distribution function, defined as:

$$\hat{F}_{j,n}(x) = \frac{1}{n+1} \sum_{i=1}^{n} I_{[x_{i,j} \le x]}$$

After the transformation is applied, the pseudo samples are used for further copula fitting as if they had uniform marginal distributions. We then use a semi-parametric approach called Canonical Maximum Likelihood Estimation (CMLE) to estimate the Copula parameters θ, and the likelihood function is given by

$$L(\theta|u_1, .., u_j, .., u_d) = \prod_{i=1}^{n} c(u_{i1}, .., u_{ij}, .., u_{id}|\theta)$$

The parameter theta is estimated by maximum likelihood:

$$\hat{\theta} = \arg\max_{\theta \in \Theta} \sum_{i=1}^{n} \log c(\hat{u}_{i1}, \dots, \hat{u}_{im}; \theta)$$

In terms of implementation, we use copula API provided by statsmodels and rewrite density function for elliptical copulas to help estimation. After fitting each copula family (Gaussian, Clayton, gumbel, Frank, StudentT), we select copula with the highest likelihood. The result of the top 5 pairs selected from the shortest distance method is shown in Figure 1. We can see that for most pairs, Gaussian Copula is selected.

pair	copula	parameter	loglikelihood
^AORD-^AXJO	Gumbel	15.340341	2847.090699
^N100-^FCHI	Gaussian	0.978139	2007.226723
^N100-^STOXX50E	Gaussian	0.972995	1838.308258
^STOXX50E-^FCHI	Gaussian	0.969519	1763.294400
^STOXX50E-^GDAXI	Gaussian	0.962432	1631.493444

Figure 1: Single Copula Fitting

2.3 Mixed copula

Our candidate copula families for mixed copula consists of four commonly used copulas, Gaussian, Clayton, Gumbel, and Frank, as all possible combinations of these four copulas have an ability to capture most of the possible dependence structures. The working formula for our mixed copula is defined as:

$$\begin{aligned} C(u, v; \boldsymbol{\theta}) = \lambda_1 C_{Ga}(u, v; \theta_1) + \lambda_2 C_{Cl}(u, v; \theta_2) \\ +\lambda_3 C_{Gu}(u, v; \theta_3) + \lambda_4 C_{Fr}(u, v; \theta_4) \end{aligned}$$

We define the joint density function of mixed copula according to the following equation:

$$f(\mathbf{x}; \boldsymbol{\phi}) = \prod_{j=1}^{p} f_j(x_j; \alpha_j) \sum_{i=1}^{n} \lambda_k c_k(F_1(x_1; \alpha_1), ..., F_p(x_p; \alpha_p); \theta_k)$$

To estimate parameters, CMLE approach is used in mixed copula situation as well. We maximize the logarithm of likelihood function with respect to the weights and copula parameters. Since we are implementing an iterative algorithm, initial value for the parameter of each copula formula is defined separately. Moreover, we delete the corresponding copula functions with very small values of weight parameters and re-fit the model with other more significant copula components.

pair	Cl-w	Cl-theta	Fr-w	Fr-theta	Gu-w	Gu-theta	Ga-w	Ga-rho	logl
^AORD-^AXJO	0.303090	28.766371	0.0	19.095699	0.688546	18.668031	0.008364	0.164089	3035.532141
^N100-^FCHI	0.194981	9.365752	0.0	12.903271	0.242956	5.435778	0.562063	0.989937	2065.137562

Figure 2: Mixed Copula Fitting

We observe a better performance of mixed copula compared to choosing the optimal individual copula, as the log likelihood improves for the same pair. Additionally, the weight on Frank Copula is zero. This means that Gaussian Copula is enough to characterize the central dependence for these two pairs.

3 Pairs Selection

Pairs selection is a critical step in pairs trading and it can be implemented through any of the following steps.

3.1 Cointegration tests

We applied Johansen test mainly to identify pairs for the benchmark cointegration model. However, we also decided to test Copula models on the same pair to be able to compare their performance. Cointegration is the long-term equilibrium relationship between two assets. Two nonstationary timeseries are considered to be cointegrated if a linear combination of them results in a stationary timeseries. The presence of cointegration suggests that any deviation from the equilibrium will eventually be corrected. Checking for cointegration entails a two-step procedure. The **first step** is to determine whether the timeseries are indeed non-stationary. We employ the augmented Dickey-Fuller (ADF) test for the purpose. The ADF test is an extension of the Dickey-Fuller test, which tests the null hypothesis that a unit root is present in the time series. A unit root indicates that the time series has a stochastic trend and is non-stationary. The ADF test adds additional lagged difference terms to the standard Dickey-Fuller test to account for any serial correlation in the data.

For the purpose of our exercise we select 29 indices on exchanges across the world which have data from year 2000 onwards. We employed the ADF test on log prices of the indices to check for non-stationarity. Since prices are usually non-stationary, as per our expectation, all 29 indices pass this test.

Once it is determined that the timeseries are indeed non-stationary, the **second step** is to check whether the timeseries cointegrate. The Johansen test is a multivariate test that estimates the number of cointegrating vectors present in a system of time series. The test uses maximum likelihood estimation to estimate the parameters of a vector error correction model (VECM). The VECM model is a multivariate model that describes the long-run equilibrium relationships between variables while accounting for short-run dynamics. The Johansen test uses the estimated parameters of the VECM to test for the presence of cointegration. Carrying out the Johansen test requires careful consideration of the lag length and the specification of the VECM model that is being fit via maximum-likelihood. In our experiments, while testing for cointegration, we set the lag period to 1, and assume a linear specification.

In our application of Johansen test, we consider a test to have passed if either the trace statistic or maximum eigenvalue statistic pass the critical value threshold. Using this test we are able to identify 80 out of 406 pairs that successfully pass the Johansen's test and can be considered cointegrating. The numbers presented in Table 1 depict the trace-value test statistic. Critical value of this test statistic at 95% is 15.49. Thus, any sample test statistic that exceeds 15.49 can be considered as evidence to reject null hypothesis and to confirm the that the pair is cointegrating.

Table 1: Results of the Johansen's cointegration test evaluated on pairs of indices. Presented results correspond to top 5 pairs in the sense of the trace value test statistic. Spread evaluation period: December 2010 to January 2017.

Pair	Sample test statistic **Critical value** (95%): 15.49
(FTSE, RUT)	27.71
(GSPTSE, TWII)	26.17
(AORD, FCHI)	25.95
(AXJO, FCHI)	24.39
(FTSE, N225)	24.04

3.2 Clustering

Identify assets whose returns are highly correlated. Two assets that have a high correlation coefficient means that they tend to move in the same direction. You can use statistical tools such as correlation analysis to identify the level of correlation between two assets.

For the purpose of our paper, we choose to use Cointegration and ML based approach for pair selection.

3.3 OPTICS

OPTICS is a useful algorithm for pair selection in trading or other applications where you want to group similar items together. By identifying clusters of similar items, pairs of assets that have similar behavior can be potentially identified.

OPTICS (Ordering Points To Identify the Clustering Structure) is a density-based clustering algorithm that is used to identify clusters of points in a dataset. Unlike other clustering algorithms, such as k-means or hierarchical clustering, OPTICS does not require the number of clusters to be specified in advance. Instead, the algorithm automatically identifies clusters based on the density of the data.

One advantage of OPTICS is its robustness to outliers. Outliers are data points that are significantly different from the other data points in the dataset and can have a significant impact on the results of clustering algorithms. OPTICS identifies clusters based on the density of the data, so outliers are less likely to affect the clustering results.

Another advantage of OPTICS is its ability to handle clusters with varying density. In some datasets, clusters may have different densities, with some clusters having a high density of points and others having a lower density of points. OPTICS can identify these clusters, even if they have different densities.

Using OPTICS 6 clusters are generated shown in the figures3 to figure8.

From the clusters we can arrive at a list of pairs:

```
Cluster 0:(000001.SS,399001.SZ)
Cluster 1:(BFX,FCHI),(BFX,N100),(FCHI,N100)
Cluster 2:(DJI,GSPC),(DJI,IXIC),(GSPC,IXIC)
Cluster 3:(NYA,RUT)
Cluster 4:(AORD,AXJO),(AORD,MERV),(AXJO,MERV)
Cluster 5:(GSPTSE,TWII)
```

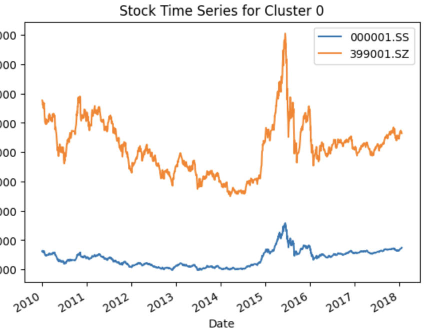

Figure 3: Cluster 0

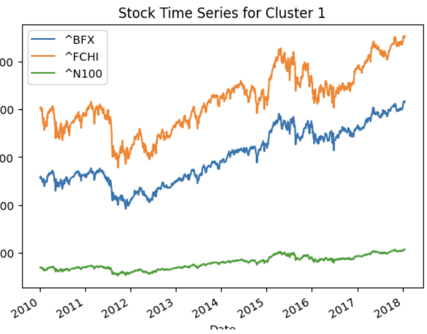

Figure 4: Cluster 1

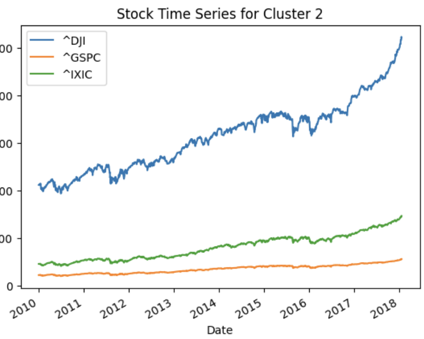

Figure 5: Cluster 2

3.4 Non Linear cointegration analysis using $\epsilon - SVR$

Although in finance cointegration normally refers to a relationship where a linear combination of 2 nonstationary processes

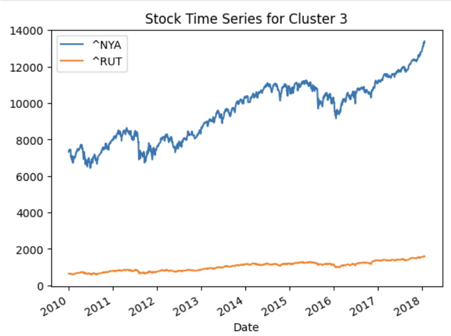

Figure 6: Cluster 3

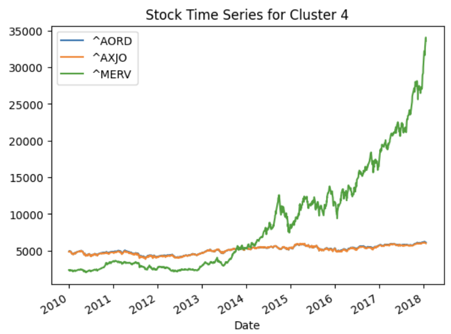

Figure 7: Cluster 4

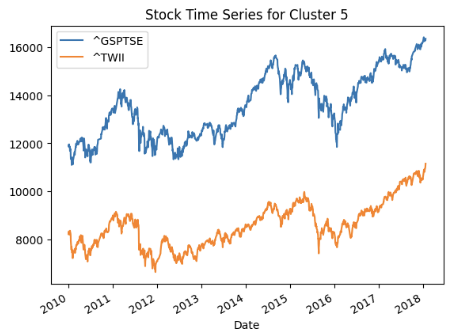

Figure 8: Cluster 5

is stationary. In other discipline such as economics, there exists a concept called nonlinear cointegration. Cointegration theory is divided into two classes: linear cointegration theory and nonlinear cointegration theory. In linear cointegration theory, cointegration reflects the degree of stationarity variability of various non-stationary stochastic series after linear combination. Engle and Granger (1987) introduced the concept of cointegration, which provided the theoretical basis for seeking equilibrium relations among two or more non-stationary stochastic series and setting up error correction models for variables with cointegration relations. This approach integrates the advantages of short-term and dynamic models used by time series analysis methods to determine the long-term stable relations in econometrics. However, nonlinear cointegration methods are being increasingly appreciated due to the strong non-stationary characteristics of financial time series.

In this context, nonparametric methods, such as the support vector machine, are being discussed, with a focus on improving the discriminate analysis using $\epsilon - SVR$. The definition of nonlinear cointegration is formulated based on this approach.

ϵ-support vector regression (ϵ-SVR) helps to overcome limitations in estimating nonlinear cointegration relations in financial time series. The steps involved in this method are as follows:[2]

- Choose financial variable time series as input vectors (Xi) for the ϵ-SVR model. Use a known short-memory time series (Di) as the supervised output for the ϵ-SVR model, i.e., the output vector $\hat{y}$.
- Use the ϵ-SVR model to estimate the nonlinear function between input vector (Xi) and output vector $\hat{y}$ Obtain the estimated output value $\hat{y}$ using the estimated nonlinear function.
- Test for the existence of the nonlinear cointegration relation among the component vectors of Xi by verifying the long memory characteristic in the time series $\hat{y}$.
- If the long memory characteristic still exists in the time series $\hat{y}$, it implies that the nonlinear cointegration relation among the component vectors of Xi does not exist. Otherwise, this kind of relation exists.

In summary, the ϵ-SVR model is used to estimate the nonlinear function between financial variables and short-memory time series. The estimated output values are then tested for the presence of a long memory characteristic, which indicates the existence of a nonlinear cointegration relation among the financial variables.

Following the above and using Federal Funds Effective Rate (DFF)(economic daily indictor) as $\hat{y}$ and daily closing prices as (Xi) of the selected pairs, applying Dickey Fuller Test to check cointegration by observing p-values allows one to understand if the pair is non-linearly cointegrated. The lower the p-value, the higher non-linear integration is among the pair.

Table 2 mentions the p-value results of performing ϵ - SVR on each selected pair.

Pair	P-value
`^GSPC` and `^IXIC`	8.10×10^{-29}
`^GSPTSE` and `^TWII`	7.59×10^{-29}
`^DJI` and `^GSPC`	6.49×10^{-29}
`^AXJO` and `^MERV`	6.12×10^{-29}
`^FCHI` and `^N100`	5.64×10^{-29}
`^DJI` and `^IXIC`	5.57×10^{-29}
`^AORD` and `^AXJO`	5.48×10^{-29}
`^AORD` and `^MERV`	5.29×10^{-29}
`^BFX` and `^FCHI`	2.33×10^{-29}
`^NYA` and `^RUT`	1.29×10^{-28}
`^BFX` and `^N100`	1.21×10^{-29}
`000001.SS` and `399001.SZ`	4.66×10^{-28}

Table 2: Results of ϵ - SVR with corresponding p-values from Dickey-Fuller test

3.5 Dependence Test

For this test, it can be divided into two parts.

The first part would be the distance calculation. We first use the pairs from Clustering and calculate daily spreads of each pair and aggregate all of them. The reason for doing first part is to identify which pair has the smallest average difference. By doing so, we can assume that two indexes in each pair is likely to be co-monotonic. However, the number of samples in each pair might be different. In order to make the results comparable, we divide the total spread of each pair by the number of its samples, and to use the average results to compare with each other.

By doing the first part, we can arrive a table of top five pairs that have the smallest distance.

Pair	Average Distance
^AORD and ^AXJO	9.92×10^{-3}
^N100 and ^FCHI	5.49×10^{-2}
^N100 and ^STOXX50E	6.51×10^{-2}
^STOXX50E and ^FCHI	6.52×10^{-2}
^STOXX50E and ^GDAXI	9.11×10^{-2}

Table 3: Results of Dependence Test with Average Distance

The second part is to use Kendall's Tau to verify the dependency of two indexes of the selected pair. By using Kendall's Tau, we then see its p-value to verify the dependency and co-monotonicity. By using the results from the first part, we can arrive a table of Kendall's Taus and its p-value of each pair.

Pair	Kendall's Tau	P-value
^AORD and ^AXJO	0.9398	2.3×10^{-60}
^N100 and ^FCHI	0.8651	1.6×10^{-60}
^N100 and ^STOXX50E	0.8605	1.8×10^{-60}
^STOXX50E and ^FCHI	0.8462	2.7×10^{-60}
^STOXX50E and ^GDAXI	0.8269	3.4×10^{-60}

Table 4: Results of Dependence Test with Kendall's Tau and p-value

From the results of the second part, we can verify that these 5 pairs not only have the smallest distances among all pairs, but they also have significant co-monotonic movement.

4 Trading Strategy

4.1 Mean Reversion

Trading rules of all models are based on mean reversion strategy. For Copula, the signal is calculated based on the conditional probabilities.

$$P\left(U_1 \leq u_1 \mid U_2 = u_2\right) := \frac{\partial C\left(u_1, u_2\right)}{\partial u_2}$$

$$P\left(U_2 \leq u_2 \mid U_1 = u_1\right) := \frac{\partial C\left(u_1, u_2\right)}{\partial u_1}$$

Trading rules are applied as appeared in [5]

The mispricing index is calculated as

$$MI_t^{X|Y} = P\left(R_t^X < r_t^X \mid R_t^Y = r_t^Y\right) - 0.5.$$

$$MI_t^{Y|X} = P\left(R_t^Y < r_t^Y \mid R_t^X = r_t^X\right) - 0.5$$

Since we used log returns as input, the flags are calculated as their cumulative sum.

$$FlagX(t) = \sum_{s=0}^{t} \left(MI_s^{X|Y}\right)$$

$$FlagY(t) = \sum_{s=0}^{t} \left(MI_s^{Y|X}\right)$$

The entry criteria is when one of the flags reaches an upper or lower threshold and another flag reaches the opposite threshold. In our experiment, we are using 0.3 as upper threshold and -0.3 as lower threshold for the flags. The exit criteria is when at least one flag reach the zero cutoff.

For cointegration approach, the entry and exit rules follow standard mean reversion strategy with the spread as a signal.

4.2 Deep Reinforcement Learning

The RL system described in this scenario utilizes a Deep Q-Network (DQN) to implement a pairs trading strategy. This strategy involves trading in a pair of stock indices that are cointegrated(chosen from the pair selection method), which means that they have a long-term stable relationship between their prices.

The DQN interacts with a reinforcement learning (RL) environment by taking actions on the spread of the stock pair. Specifically, the DQN can choose to long, short, or take no position on the spread. The spread is the difference between the prices of the two stocks in the pair, and it is used as an indicator of the potential profitability of the trading strategy.

The DQN produces a Q-function(robust in nature), which is a mathematical function that estimates the expected future reward for each possible action in each possible state of the environment. The Q-function learns the trading parameters that will maximize the profit of the pairs trading strategy. [1]

It overcomes the problem of manually defining trading actions based on stop-loss condition/ executing a trade action when threshold value for spread is touched instead the DQN approach directly determines trading actions based on the spread and reward system.

The pairs trading strategy in this RL system is based on 10 trading parameters that represent the spread between the two stocks in the pair. In this approach, the daily prices data for the training set from 2018-2021 is pre-processed to generate the 10 input features for each day, which represent the state of the environment for that day. The DQN takes these input features and outputs an action of long (Buy Stock index-1 and short Stock index-2), short (Short Stock index-1 and Buy Stock index-2), or no position on the spread of the pair.

The DQN utilizes RELU non-linear activation functions and an Adam optimizer to update network weights. After training for 100 episodes, the DQN is able to converge on a near-optimal set of weights for the input features, and produce a Q-function to maximize reward for any given state.

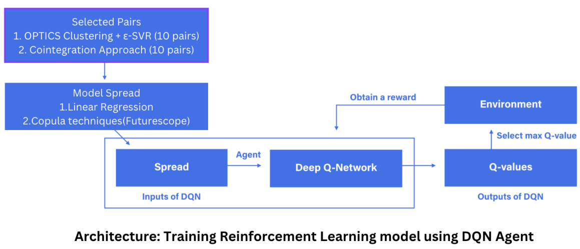

Figure 9: DQN Architecture Diagram

Trading period/Testing the performance is done on the period 2022-2023. The initial amount of money allocated for the trading strategy. In this case, it is set to 100,000.

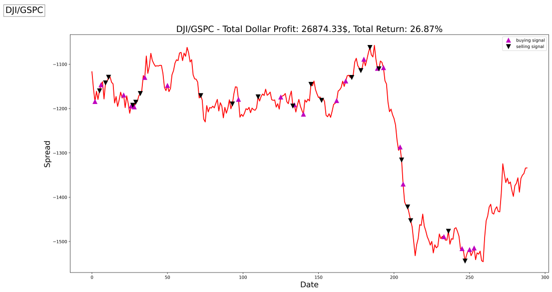

Figure 10: DJI GSPC actions over spread

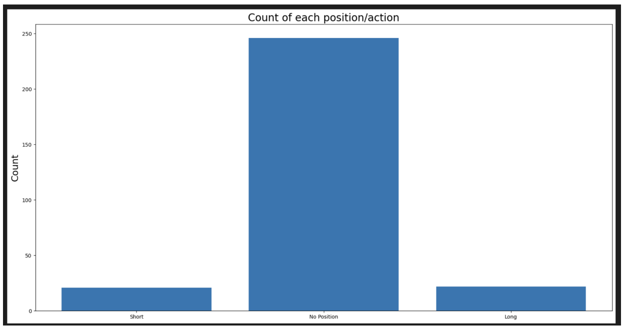

Figure 11: DJI GSPC Position/Action count

5 Experiment Design

We conducted the experiment on rolling training and testing periods to account for strategy updating in practice.

5.1 Model 1 : Cointegration

As discussed in the prequel, cointegration-based pairs trading is a market-neutral trading strategy that involves identifying two financial instruments that are co-integrated. Pairs trading using cointegration involves identifying two co-integrated assets, and then looking for periods when the two assets deviate from their long-term equilibrium relationship. When one asset is trading at a premium to the other, the trader would short the overpriced asset and long the underpriced asset, with the expectation that the prices will eventually converge back to their long-term relationship. The key idea of the strategy is that if two time series are cointegrating, then it means they remain close over time. In other words, the spread $z_t = y_2t - \gamma y_1t$ is mean-reverting. This property can be exploited for trading by suitably short/sell the over-priced index and long/buy the under-priced index, effectively long/short the 'spread'.

In this strategy, we fit an ordinary linear regression (OLS) model on 63 trading days (one quarter) to construct the mean-reverting spread. Next, we set up trading rules using fixed entry and exit thresholds on the constructed spread. We use these thresholds for trading during the subsequent 21 days (one trading month). The aforementioned steps are repeated after every trading month, i.e., the spread is recalculated through regression and used for the following trading month. We refer to this strategy as "dynamic threhsold" based pairs trading strategy, because the thresholds are adjusted monthly based on the updated spread.

We evaluate this strategy on the 80 identified pairs from the (Johansen) cointegration test as discussed earlier. The results of the top-10 best performing pairs are presented in Table 7. We observe that 4 pairs achieve a Sharpe ratio greater than 1. We construct an additional baseline of a portfolio consisting of an equally-weighted combination of top 10 best performing pairs, referred to as EW portfolio in Table 7. Since constituent pairs are relatively independent to each other, the equally-weighted portfolio of the pairs resulted in an improved Sharpe ratio of 1.63 surpassing the performing of the best constituent pair.

5.2 Model 2 : Cointegration + LSTM

To provide a comparison for the Copula based trading strategy, we decided to also implement a deep learning model using the popular LSTM model to try and implement a nonlinear trading strategy. LSTM, or Long Short-term Memory, is a neural network design that uses a input gate, output gate, and forget gate in an attempt to learn from useful long term patterns in the data that traditional recurrent neural networks are unable to.

One hurdle is that deep learning models like LSTMs require extremely large data sets to train on. As we are constrained to using daily price data, there simply isn't enough samples to build a large and robust deep learning model without overfitting to the existing data. Additionally, the data we do have from yahoo finance is incomplete from many over seas indices with either missing data points or simply do not enough history. Because of this, we limit ourselves here to the us equity indices: GSPC (SP500), RUT (Russell 2000), IXIC (Nasdaq 100), and DJI (Dow Jones Industrial Average). We observed that the OHLCV data for these are complete and have a daily price history going back more than 10 years. Additionally, the pairs GSPC/IXIC, GSPC/RUT, IXIC/DJI, DJI/GSPC were all found to be suitable candidates for pairs trading by the OPTICS algorithm described prior.

For each pair of indices we start be using a OLS regression to regress the log price of the one onto the log price of the other. Doing so we get beta which we can use to model the spread of the two indices as one might to in a traditional linear pairs trading strategy. At the point we use the popular TA-LIB python package to create a series of technical indicators applied to the spread to capture the momentum, volatility, and volume of the spread. After applying a standard scaler, we than have our "X" data set. For our "Y" data set, our target variable, we use the one period future shifted Close spread of the two equities. I.e. we are trying to use the spread and technical information we have at time t to predict where the spread will be at time t+1.

We use 10 Years, of historical data for each pair, using the first 8 years for training and the last 2 years for validation. Due to the small nature of the data set, 2500 samples for each pair, we are using a small neural network to try and prevent over fitting. We use two LSTM layers with 8 neurons each and finally a dense layer with a single node with a linear activation to predict the next spread.

Table 5: Top Performing pairs for Deep Reinforcement Learning - DQN Results

Pair	Annual return	Annual volatility	Sharpe ratio	Total return
(DJI, GSPC)	1.325%	0.492%	2.690	**26.87%**
(DJI, IXIC)	0.557%	0.309%	1.802	19.98%
(AORD, MERV)	**28.297%**	5.637%	**5.019**	18.50%
(NYA, RUT)	0.283%	0.203%	1.393	16.18%
(000001.SS, 399001.SZ)	0.175%	0.139%	1.259	10.13%
(AORD, NYA)	0.105%	0.126%	0.829	7.69%
(FCHI, N100)	0.070%	0.126%	0.833	7.58%
(AXJO, BFX)	0.032%	0.051%	0.635	3.68%
(GSPC, IXIC)	0.093%	0.124%	0.748	1.90%

Table 6: Backtesting results of dynamic-threshold based pairs trading strategy on the top 10 performing pairs. Out of sample evaluation period: January 2018 to January 2023. EW-portfolio denotes the results of an equally-weighted portfolio consisting of the presented top 10 pairs.

Pair	Sharpe ratio	Annual return (%)	Annual volatility (%)
(AXJO, RUT)	1.43	19.35	13.77
(AORD, NYA)	1.19	14.47	12.00
(AXJO, NYA)	1.04	13.67	13.37
(AXJO, N100)	1.00	10.79	10.50
(AXJO, FCHI)	0.99	11.53	11.85
(AORD, RUT)	0.98	14.69	13.92
(AORD, FCHI)	0.97	11.56	11.99
(AXJO, FTSE)	0.89	9.73	10.11
(AORD, N100)	0.73	9.82	12.67
(AORD, GDAXI)	0.72	9.83	12.48
EW portfolio	**1.63**	12.03	7.35

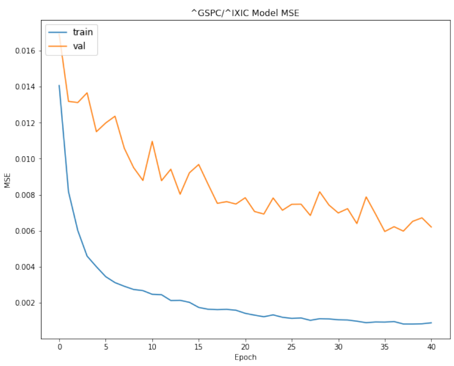

Figure 12: LSTM MSE Loss

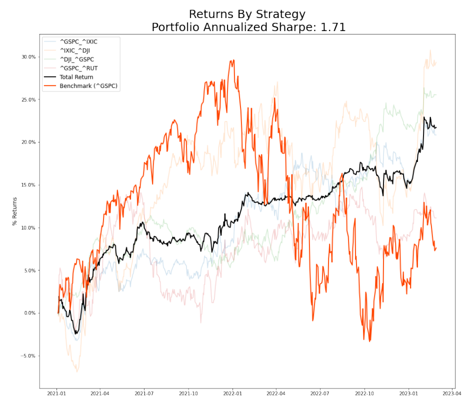

Figure 13: LSTM Strategy Out of Sample Returns

After generating the predictions for the next spread, we can then formulate a trading strategy. If we predict the spread between index A and index B to be higher in the next period, we buy index A and short index B and vice versa. We do this for all four of the index pairs and the generate a combined portfolio as the sum of the positions taken by all four of the strategies. Doing so gives and annualized Sharpe ratio of 1.71, handily beating our benchmark (GSPC).

5.3 Model 3 : Single Copula

In this model, we are trying to identify the Copula that best-fits the data. First and foremost, the data we are using will not be processed through GARCH Filter. Instead, we are using the log-returns of 2 securities directly. In order to implement Copula method, we then transform the data of log-returns into uniform marginals. Further, to select the best-fitted Copula, we first plot the joint-distribution of CDFs (Cumulative Distribution Function) of two marginals, and select the possible choices of Copulas based on the graphical plot. This step can be considered as an initial filter. However, in order to capture the tail movements (the extreme values), Elliptical Copulas, which are Student-t Copula and Gaussian Copula, would not be considered. We only take Archimedean Copulas into consideration.

The second step is to use CMLE to estimate the parameters for Copula and select the copula that has the maximum pseudo-logarithm-likelihood among all. By using marginal CDFs, which can be estimated by empirical CDFs, we try to fit the Copula and find the parameters of Copula by trying to maximize its the sum of pseudo-logarithm-likelihood of each sample. We then compare the pseudo-logarithm-likelihood of

Table 7: Back-testing results of LSTM pairs trading strategy on the 4 US equity pairs. Out of sample evaluation period: January 2021 to February 2023. EW-portfolio denotes the results of an equally-weighted portfolio consisting of the presented 4 pairs.

Pair	Sharpe ratio	Total Return (%)
(GSPC, IXIC)	1.07	21
(IXIC, DJI)	.92	29
(DJI, GSPC)	1.83	26
(GSPC, RUT)	.44	11
Benchmark	.166	8
EW Portfolio	**1.71**	22

each Copula and select the one which has the maximum value.

Lastly, in order to confirm the results, we implement Goodness of Fit Test. The principle of the test is to see whether a null hypothesis is true or not. We conduct the test by building a null hypothesis based on the results from the previous step, denoted as C_0. That is, we hypothesize the true Copula is the Copula that has the maximum pseudo-logarithm-likelihood value ($C \in C_0$), and the alternative hypothesis would be that the true Copula is not the Copula that has the maximum value of pseudo-logarithm-likelihood.

We then calculate the distance between the empirical Copula and the Copula that we are testing. To elaborate, we select samples from the original data with the increment of 5 and build an empirical Copula. The reason for building an empirical Copula is that empirical Copula is an entirely non-parametric Copula, which can thus be the most objective benchmark for testing. The empirical Copula can be expressed as the following, where n is the number of samples, U_i is the uniform distribution of each variable, and d is the number of variables.

$$C_n(\mathbf{u}) = \frac{1}{n}\sum_{i=1}^{n} I(U_{i,1} \le u_1, \ldots, U_{i,d} \le u_d),$$

$$\mathbf{u} = (u_1, \ldots, u_d) \in [0,1]^d$$

Based on the samples, we then calculate the CDF of each Copula that we want to test, compare it with the empirical Copula, and calculate the distance. Lastly, large value of the distance indicates small p-value and thus leads to the rejection of the null hypothesis.

By following the steps, we can identify and verify the 'true' Copula.

5.4 Model 4 : Mixed Copula

In this model, we do not apply GARCH Filter to process the data. Instead, we use the logarithm-returns of 2 securities as data input directly.

To build a mixed Copula model, we will be using Clayton, Frank, Gumbel and Gaussian Copula, and assigning initial weight parameters for each Copula. The optimization technique we will be using is CMLE, to tune the weight parameters to maximize the pseudo-logarithm-likelihood.

The model will ignore the copula which has small value of weight parameter and re-fit the model with other significant copula components.

5.5 Model 5 : GARCH Filter +single copula

The model applies GARCH Filter to process the log-returns of 2 securities. We use the residuals from GARCH Filter as data input and redo the selection process of **5.3 : Single Copula**.

5.6 Model 6 GARCH filter + mixed copula

The model applies GARCH Filter to process the log-returns of 2 securities. We use the residuals from GARCH Filter as data input and redo the maximization approach of **5.4 : Mixed Copula** to tune the weights and Copula parameters.

6 Conclusion

Based on the results of our research, we can conclude that copula, deep reinforcement learning, and LSTM are effective tools for generating alphas in finance.

Firstly, copula allows for the modeling of complex dependencies between variables, which is crucial in finance where various factors can influence asset prices. By incorporating copula models into alpha generation strategies, we can gain a better understanding of how different factors affect asset prices and use this information to make more informed investment decisions.

Based on the results of our research, we have found that copula performs worse than Linear modeling/cointegration in generating alphas, which is not what would typically be expected. However, we have also discovered that the performance of copula can be greatly improved by using GARCH filters.

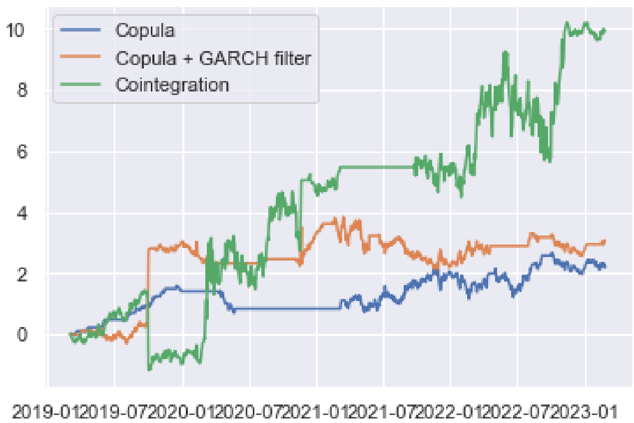

Figure 14: PnL

Our findings suggest that cointegration may be a more reliable method for generating alphas than copula. However,

```
Annualized Sharpe Ratio:
Copula:  0.5891805790369806
Copula+GARCH filter:  0.4145771710424504
Cointegration:  0.5864915252851512
Position Counts:
Copula:  21
Copula+GARCH filter:  14
Cointegration:  10
```

Figure 15: Sharpe Ratio and Position Counts

the inclusion of GARCH filters can significantly enhance the performance of copula. GARCH filters are useful in modeling the volatility of financial time series data, and by incorporating them into copula-based alpha generation strategies, we can better capture the dynamics of financial markets and improve the accuracy of our predictions.

Secondly, deep reinforcement learning is a powerful technique that has been shown to outperform traditional methods in various applications, including finance. By using deep reinforcement learning to generate alphas, we can leverage its ability to learn complex patterns in financial data and make predictions based on this knowledge.

Lastly, LSTM, a type of neural network that is well-suited for sequential data, has also shown promise in alpha generation. By using LSTM to analyze historical financial data and identify patterns, we can generate alphas that capture trends and make more accurate predictions.

7 Future works

We can try to investigate the use of DQN or Double DQN reinforcement learning algorithms for trading pairs based on underlying copula techniques, such as single copula and mixed copula.

References

[1] Andrew Brim. Deep reinforcement learning pairs trading. Utah State University, All Graduate Plan B and other Reports, 1425., 2019.

[2] Dufrénot G and Mignon V. Recent developments in nonlinear cointegration with applications to macroeconomics and finance. *New York: Kluwer Academic Publishers*, 2002.

[3] Ji Q. Fan Y. Liu, B.-Y. A new time-varying optimal copula model identifying the dependence across markets. quantitative finance. 2016, 17(3).

[4] Simão Moraes Sarmento and Nuno Horta. Enhancing a pairs trading strategy with the application of machine learning. Expert Systems with Applications, vol. 158, Nov. 2020, p. 113490.

[5] Hudson Thames. Copula for pairs trading: Overview of common strategies. 2021.