Dynamic Asset Allocation for Pairs Trading

Franco Ho Ting Lin francohtlin@cs.toronto.edu
Department of Computer Science
University of Toronto
Toronto, ON M5S 2E4, Canada

Abstract

Pairs trading and statistical arbitrage methods have been shown to be highly successful in the past. In this project, we propose a straight forward procedure to construct a portfolio of pairs. Pairs are determined through the Engle and Granger test for Cointegration and our portfolio is constructed through a simple ranking procedure. We also provide a dynamic approach to the asset allocation problem. An adaptation to the Recurrent Reinforcement Learning algorithm is used to provide weightings for each pair. We demonstrate the capabilities of the proposed methodology through empirical results on securities from the New York Stock Exchange (NYSE).

1. Introduction

Statistical arbitrage methods have shown to be profitable in the past [5]. Nunzio Tartaglia first introduced pairs trading in 1987 to great success. The basic principle behind pairs trading is that we want to identify two securities whose prices have moved together historically. When there is an anomaly in the relationship, we sell the overvalued security and buy the undervalued one, with the hope that the mis-pricing will eventually correct itself [11].

There are many factors in determining if our portfolio will succeed. From the selection of pairs, determination of anomalies, management of assets and position sizing; the potential ways of building such a portfolio is not straight forward and may warrant confusion. A review of the methods used in the past is provided by [5]. We used ideas from [5] to provide a different approach for constructing a portfolio. We use Engle and Granger Method for cointegration to create the initial set of candidate pairs. We then perform an iterative search to obtain thresholds that denote when such anomaly has occurred and rank the top pairs based on profit. A dynamic approach to asset allocation is also presented. We propose an adaptation of the Recurrent Reinforcement Learning algorithm by [7;8] to provide dynamic weightings for each pair.

The rest of the paper is organized as follows, in Section 2, we provide the full procedure in constructing our portfolio of pairs. In section 3, we present our model used for asset allocation and derivations for how the model is trained. In Section 4, we define the metrics used to evaluate how well our model and portfolio performs. Section 5, provides a demonstration of the procedure for pair selection and empirical results of our model tested on securities from the New York Stock Exchange (NYSE) against the 3 Month Treasury Bill (T-Bill).

2. Pairs Trading

One of the keys to success for pairs trading is the selection of pairs. We want to find two securities that are similar in nature. When their prices have diverged a certain amount, we short the overvalued security and buy the undervalued one. We then close our position when the prices revert back to their historical mean. In the following section, we will discuss the selection of pairs, determination of trading thresholds and the ranking of pairs.

2.1 Definitions

- Let N be the set of securities we are interested in.
- Let F be our formation period.
- Let T be our trading period.
- Let A, B be the two securities of interest
- Let $X_{A,t}, X_{B,t}$ be the two the price of each security at time t
- Let $r_{A,t} = \frac{X_{A,t}}{X_{A,t-1}} - 1$ be the immediate daily return for the security A.

2.2 Cointegration Approach

Two securities are determined as candidate pairs if they are cointegrated. If there exists some linear combination of the two time series that is stationary, they are therefore cointegrated. We use Engle and Granger method [2] to find securities that are cointegrated and the procedure is listed as follows:

1. Verify the two securities $X_{A,t}, X_{B,t}$ are $I(1)$ non-stationary time series using Augmented-Dickey Fullers (ADF) Test.
2. Regress one series on the other to obtain the cointegration coefficient and intercept β_1, β_0
3. Verify our regression residuals (spread) given by equation 1

$$spread_t = X_{B,t} - \beta_1 X_{A,t} - \beta_0 \tag{1}$$

   is $I(0)$ stationary using ADF.

4. If all the above are satisfied, we can conclude the two securities are cointegrated.

We perform the above procedure for all two security combinations from the NYSE, and collect the pairs that are cointegrated. Let us denote the set of cointegrated pairs as $Coint_N$.

The cointegration approach is valid due to the Granger Representation Theorem [2]. The theorem states that if X_t and Y_t are cointegrated, then there exists an Error Correction Model (ECM) for the two series. The key intuition behind the ECM is that it shows our cointegrated pairs to have a long-run equilibrium. If there is some slight deviation from this equilibrium in the short-run, then one or both of our series will adjust to revert back to the equilibrium [11]. Therefore, we take the above justifications of the cointegration approach to be a good indication that our method is valid.

Another way to test for cointegration would be to use the Johansen Tests. The procedure is provided in Appendix A along with a simple demonstration. The benefit behind using Johansen Test is that the procedure will result in a linear combination which is most stationary. The choice of Engle and Granger Method is because it minimizes variance as compared to Johansen test which maximizes stationarity [6]. The simplicity of implementation allows us to test and verify results quickly.

2.3 Trading Thresholds

With our set of cointegrated pairs $Coint_N$, we now want to determine when to open and close trades. Let us first define the long-short positions we will take once a certain threshold η is exceeded.

$$I_{AB,t} = \begin{cases} 0, & not \;\; open \\ 1, & Long \;\; A, Short \; B \quad if \; spread_t > \eta \\ -1, & Short \;\; A, Long \; B \quad if \; spread_t < -\eta \end{cases} \tag{2}$$

In other instances, our threshold could be a predefined constant like $\eta = 2$ [9]. For better performance, we opted to use a different threshold for each pair by taking the best threshold $\eta*$ which generates the largest profit. We define profit and wealth as follows:

$$W_{AB,T} = W_0 \prod_{t=1}^{T} \{1 + r_{AB,t}\} \tag{3}$$

where daily return of the pair AB is denoted as $r_{AB,t}$ and the wealth at time $t = T$ is denoted as $W_{AB,T}$. We will take an initial wealth $W_0 = 1$ and define the profit of the pair as:

$$P_{AB,T} = W_{AB,T} - W_0 \tag{4}$$

To find the optimal threshold, we perform an iterative search over 9 evenly spaced thresholds between the mean and maximum of the spread. We will denote these thresholds as η_k where $k \in [1, \cdots, 9]$

$$\text{diff} = \max_{t \in F}(spread_t) - \mu_{spread} \quad \text{and} \quad \eta_k = \mu_{spread} + k * \text{diff} \tag{5}$$

Once we have our thresholds, the optimal $\eta*$ is the threshold that maximizes our profit during the formation period.

$$\eta* = \max_{\eta_k}(P_{AB,F}) \tag{6}$$

We then collect the $\eta*$ and $P_{AB,F}$ for all the cointegrated pairs.

2.4 Portfolio Construction

Each pair from $Coint_N$ is ranked based on their performance in the formation period. We take the top 5 pairs with the largest profit. Once a security has been selected, we omit all the following pairs that also contain that security. This was used to prevent the possibility of taking both long and short positions on the same security. From here on out, we will define the final 5 pairs used in our portfolio as A_5.

Note that this may not necessarily guarantee the performance of our pairs in the trading period. Although this may be a naive way of constructing a portfolio, for the purposes of this demonstration it will suffice. The number of pairs selected in the portfolio is arbitrarily decided and the selected pairs can be found in Table 3.

3. Dynamic Asset Allocation

With the portfolio constructed, we now want to obtain weightings for each pair that change over time as the prices change. The motivation behind the model used is from a family of works [7;8] where they use a recurrent reinforcement learning architecture to execute trades. We use a similar architecture to provide weightings for each pair in our portfolio. The following section will provide the layout of how these weightings are obtained.

3.1 Definitions

- Let D_5 be the top 5 pairs used in our portfolio
- Let $p \in D_5$ be a single pair used in our portfolio
- Let $\theta^{(p)} = [\theta_0, ..., \theta_n]$ be the vector of parameters for pair p
- Let $r_t^{(p)}$ be the return for pair p
- Let $F_t^{(p)}$ be the weighting for pair p

3.2 Architecture for a Single Pair

The architecture for a single pair is simple linear regression with a recurrent connection. Our input variables for a single pair is denoted as:

$$x_t = [1, r_t, r_{t-1}, ..., r_{t-m}, F_{t-1}]^T \tag{7}$$

where $r_t, \cdots, r_{t-m}$ is the immediate daily return of our pair at time $t, t-1, \cdots, t-m$. We will select a window of size m to provide our model with time dependent features. There also exists a single recurrent connection where the previous weighting of the pair F_{t-1} is included as input.

A $Softmax$ function is used for the output of our model to squash the output between $[0, 1]$. This is a natural weighting for each pair, where $F_t = 0$ means our pair is not to be used in the portfolio, and $F_t = 1$ means we place all the capital available into this single pair. Let the $Softmax$ be denoted as:

$$Softmax(z_j) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

We can denote our linear model as a function f parametrized by some θ with output F_t given by the $Softmax(\cdot)$

$$f^{\theta}(x_t) = \theta_0 + \theta_1 r_t + \theta_2 r_{t-1} + \cdots + \theta_{n-1} r_{t-m} + \theta_n F_{t-1} \tag{8}$$

$$F_t = Softmax[f^{\theta}(x_t)] \tag{9}$$

We now have a concise form to represent a single pair and can apply this methodology across multiple pairs.

3.3 Architecture for Multiple Pairs

To simplify the previous single pair architecture, we will adopt the following vector notations.

$$x_t^{(p)} = [1, r_t, r_{t-1}, ..., r_{t-m}, F_{t-1}]^T, \quad \theta^{(p)} = [\theta_0, ..., \theta_n]^T \tag{10}$$

For every pair $p \in D_5$, we have the simplification:

$$Softmax[f^{\theta^{(p)}}(x_t^{(p)})] = F_t^{(p)} \quad \text{where} \quad \sum_{i=1}^{D_5} F_t^{(i)} = 1 \tag{11}$$

A direct consequence of the $Softmax$ function is that the sum of all of our outputs $F_t^{(p)}$ at every time step will equal one. This is needed as we do not want to weight our portfolio to be more than 1. Let us denote the immediate daily return for our portfolio as:

$$R_{D_5,t} = \sum_{i=1}^{D_5} F_t^{(i)} r_t^{(i)} \tag{12}$$

where $r_t^{(i)}$ is our immediate return of pair i at time t. Let us also define the Sharpe Ratio[10] S_n as a risk-adjusted return to both verify our portfolio performance and to train our model. We adopt the same equation used by[7;8] denoted as follows:

$$S_n = \frac{Average(R_n)}{Standard\ Deviation(R_n)} = \frac{A_n}{K_n(B_n - A_n^2)^{1/2}} \tag{13}$$

with

$$A_n = \frac{1}{n}\sum_{t=1}^{n} R_t \quad B_n = \frac{1}{n}\sum_{t=1}^{n} R_t^2 \quad K_n = \left(\frac{n}{n-1}\right)^{1/2} \tag{14}$$

We use the value K_t as a normalizing factor for our unbiased estimate of the standard deviation and remove the risk free rate of return as proposed by [7;8].

With all the definitions for our full model, we can examine the architecture for dynamic asset allocation in Figure 1. Note that our inputs are not connected to all the nodes. We do not have input variables from one pair being passed to parameters of another pair. This specific architecture was used as an attempt to have interpretable parameters where $\theta_t^{(i)}$ is strictly used for the pair i and inputs $x_t^{(i)}$. The Sharpe Ratio provides us with a value in which our model can optimize on. We want to obtain weightings $F_t^{(i)}$ for each pair which will give us the maximum Sharpe Ratio. We will next describe how the model is trained based on this maximization.

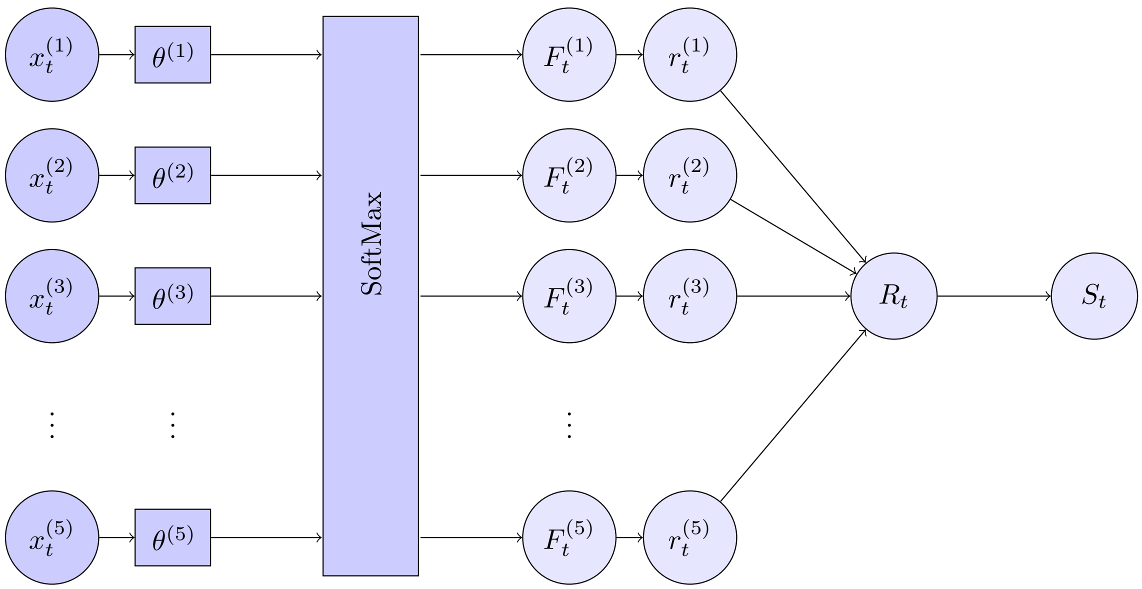

Figure 1: Dynamic Asset Allocation Model for Multiple Pairs

3.4 Training the Model

Maximizing some value corresponds to performing gradient ascent updates on the parameters. Let us denote the gradient step for each pair (p) as follows:

$$Gradient: \ \Delta\theta^{(p)} = \rho \frac{dS_T(\theta^{(p)})}{d\theta^{(p)}} \tag{15}$$

We take the derivative of the *Sharpe Ratio* at time T with respect to our weight parameters $\theta^{(p)}$. This allows us to perform batch updates after each iteration of the formation period as follows:

$$\theta_{t+1}^{(p)} = \theta_t^{(p)} + \Delta\theta^{(p)} \tag{16}$$

Presented below are the equations used to obtain our gradient $\Delta\theta^{(p)}$, we will be simply applying the chain rule from the sharpe ratio all the way back to each set of parameters.

$$
\begin{aligned}
\frac{dS_T(\theta^{(p)})}{d\theta^{(p)}} &= \sum_{t=1}^{T}\left\{\frac{dS_T}{dA_T}\frac{dA_T}{dR_t} + \frac{dS}{dB_T}\frac{dB_T}{dR_t}\right\}\left\{\frac{dR_t}{d\theta^{(p)}}\right\} \\
&= \frac{1}{T}\sum_{t=1}^{T}\left\{\frac{B_T - A_T R_t}{K_T(B_T - A_T^2)^{3/2}}\right\}\left\{\frac{dR_t}{d\theta^{(p)}}\right\}
\end{aligned} \tag{17}
$$

Given that our $F_i^{(p)}$ is an output from the *Softmax* function, we will obtain a *Jacobian* when taking the derivative with respect to our parameters. By looking at the specific case for $\theta^{(p)}$, we have:

$$
\begin{aligned}
\frac{dR_t}{d\theta^{(p)}} &= \frac{\partial}{\partial\theta^{(p)}}\sum_{k=1}^{D_5} F_t^{(k)} r_i^{(k)} \\
&= \sum_{k=1}^{D_5} r_t^{(k)} \frac{\partial F_t^{(k)}}{\partial f_t^{\theta^{(p)}}}\frac{\partial f_t^{\theta^{(p)}}}{\partial\theta^{(p)}}
\end{aligned} \tag{18}
$$

with the *Jacobian* given as follows:

$$
J = \begin{bmatrix}
\frac{\partial F^{(1)}}{\partial f^{\theta^{(1)}}} & \cdots & \frac{\partial F^{(1)}}{\partial f^{\theta^{(D_5)}}} \\
\frac{\partial F^{(2)}}{\partial f^{\theta^{(1)}}} & \cdots & \frac{\partial F^{(2)}}{\partial f^{\theta^{(D_5)}}} \\
\vdots & \ddots & \vdots \\
\frac{\partial F^{(D_5)}}{\partial f^{\theta^{(1)}}} & \cdots & \frac{\partial F^{(D_5)}}{\partial f^{\theta^{(D_5)}}}
\end{bmatrix} \quad \text{where} \quad \frac{\partial F^{(i)}}{\partial f^{\theta^{(j)}}} = \begin{cases} F^{(i)}(1 - F^{(j)}), & i = j \\ -F^{(i)}F^{(j)}, & i \neq j \end{cases} \tag{19}
$$

Given the *Jacobian* we can now take the derivative of R_i with respect to $\theta^{(p)}$

$$
\begin{aligned}
\frac{dR_t}{d\theta^{(p)}} = r_t^{(1)}(-F_t^{(1)}F_t^{(p)})\frac{\partial f_t^{\theta^{(p)}}}{\partial\theta^{(p)}} + \cdots + r_t^{(p)}(F_t^{(p)}(1 - F_t^{(p)}))\frac{\partial f_t^{\theta^{(p)}}}{\partial\theta^{(p)}} + \cdots \\
+ r_t^{(D_5)}(-F_t^{(D_5)}F_t^{(p)})\frac{\partial f_t^{\theta^{(p)}}}{\partial\theta^{(p)}}
\end{aligned} \tag{20}
$$

Since $f_t^{\theta^{(p)}}$ is dependent on the previous value $F_{t-1}^{(p)}$, the derivative will be computed as follows:

$$
\frac{\partial f_t^{\theta^{(p)}}}{\partial\theta^{(p)}} = x_t^{(p)} + \theta_n^{(p)}\frac{\partial F_{t-1}^{(p)}}{\partial\theta^{(p)}} \tag{21}
$$

where

$$
\frac{\partial F_{t-1}^{(p)}}{\partial\theta^{(p)}} = \cancelto{0}{\frac{\partial F_{t-1}^{(p)}}{\partial f_{t-1}^{\theta^{(1)}}}\frac{\partial f_{t-1}^{\theta^{(1)}}}{\partial\theta^{(p)}}} + \cdots + \frac{\partial F_{t-1}^{(p)}}{\partial f_{t-1}^{\theta^{(p)}}}\frac{\partial f_{t-1}^{\theta^{(p)}}}{\partial\theta^{(p)}} + \cdots + \cancelto{0}{\frac{\partial F_{t-1}^{(p)}}{\partial f_{t-1}^{\theta^{(D_5)}}}\frac{\partial f_{t-1}^{\theta^{(D_5)}}}{\partial\theta^{(p)}}} \tag{22}
$$

Since $f_{t-1}^{\theta^{(p)}}$ is the only function parametrized by $\theta^{(p)}$, we can cancel out the derivatives not parametrized by $\theta^{(p)}$, we will end up with the following simple equation:

$$
\frac{\partial F_{t-1}^{(p)}}{\partial\theta^{(p)}} = \frac{\partial F_{t-1}^{(p)}}{\partial f_{t-1}^{\theta^{(p)}}}\frac{\partial f_{t-1}^{\theta^{(p)}}}{\partial\theta^{(p)}} = F_{t-1}^{(p)}(1 - F_{t-1}^{(p)})\frac{\partial f_{t-1}^{\theta^{(p)}}}{\partial\theta^{(p)}} \tag{23}
$$

Note that the parameter update for a single pair in equation 20 is dependent on a previous parameter. This is present because of the recurrent connection. When training the model, we initialize the previous gradients of the parameters in equation 23 to be all zeros and keep a copy of the previous derivatives after each iteration.

In[7;8], they derived an online-learning approximation for the recurrent connection, allowing the algorithm to use stochastic gradient ascent. This method should be investigated as we can perform updates at a faster speed, thus resulting in a model that can be updated after each day of trading. For the purposes of this project, the above equations are sufficient for batch gradient ascent updates after T. Now that we have derived all the equations needed, we can train our model by taking gradient ascent updates that maximize $SharpeRatio$.

4. Evaluation

In order to verify that our model is performing well, we will look into the three different metrics. We will compare our model with even weighting of pairs and the baseline T-Bill.

We use profit to verify that our dynamic weighting is in fact performing better than even weightings. We first compute the accumulated wealth of the portfolio similar to Equation 3:

$$W_{D_5,T} = W_0 \prod_{t=1}^{T} \{1 + R_{D_5,t}\} \tag{24}$$

where $R_{D_5,t}$ is the immediate daily return of our portfolio computed by 12. The accumulated wealth for even weighting is computed as follows:

$$W_{E,T} = W_0 \prod_{t=1}^{T} \{1 + R_{E,t}\} \quad \text{where} \quad R_{E,t} = \sum_{k=1}^{K} \frac{1}{K} r_t^{(k)} \tag{25}$$

Note that K denotes the set of pairs that are currently open. We need to distinguish this since we want to utilize all of our capital at each time frame and there may be situations where certain pairs are not being traded. We will take $W_0 = 1$ to be the initial investment in both cases and the profit will be computed as $P_{(\cdot),T} = W_{(\cdot),T} - W_0$ where $W_{(\cdot),T}$ is the accumulated wealth for dynamic weighting or even weighting.

The same Sharpe Ratio given by Equation 13 will also be used to measure the performance of our model. We will substitute the excess returns $\tilde{R}_t = R_t - R_t^f$ with simply R_t, where R_t^f is the risk free rate of portfolio returns[7;8]. This is used to simplify the analysis of our model. It allows us to not have our model be dependent on some risk free rate. The higher this value is, the better our portfolio is performing.

The third measure used is the Information Ratio defined as follows:

$$IR = \frac{E(R_{D_5} - R_b)}{\sigma(R_{D_5} - R_b)} \tag{26}$$

where R_p is the return of our portfolio and R_b is the return of some benchmark. We take the expected value of the active return divided by the standard deviation of the active return. The interpretation of this ratio is to gauge how well our model is performing compared to if we simply invested in some benchmark. The benchmark is the T-Bill rate. According to[3], a value of $IR > 0.5$ will put us in the top-quartile of fund managers. If obtain a negative IR, it means that we are worse off than just investing in the T-Bill.

5. Results

5.1 Cointegration and Threshold Search Demonstration

In the following section, we provide a full demonstration of Section 2 using the securities TE Connectivity Ltd. (TEL) and United Technologies Corporation (UTX). The results from the cointegration tests and optimal threshold search are shown below.

Security	Test-Statistic
TEL	1.1301
UTX	1.0733
Residuals	-4.6780

Table 1: ADF Test-Statistics Output for the TEL, UTX and Regression Residuals for lm(TEL-UTX)

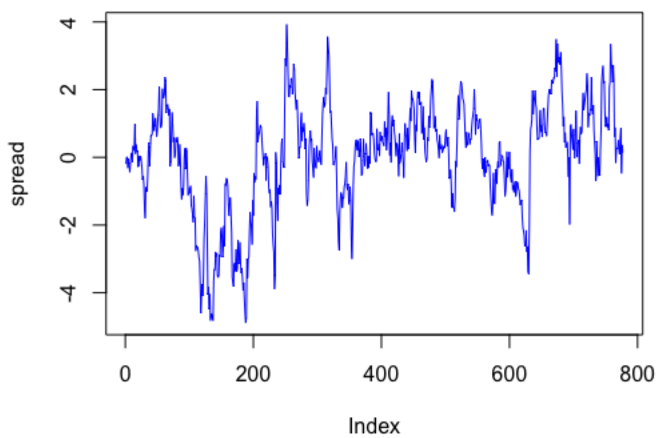

Figure 2: Regression residuals for TEL and UTX. The R function lm(TEL UTX) is used to obtain our coefficient of cointegration $\beta_1 = 0.6034$ and intercept $\beta_0 = -13.5047$

Let us first check that our two securities are cointegrated using Engle and Granger Method. By looking at Table 1 we see that both test-statistics lie within the critical values of $-2.58, -1.95$ and -1.62. Therefore the two securities are non-stationary $I(1)$. Next we regress one on the other using Ordinary Least Squares (OLS) to obtain our coefficients $\beta_1 = 0.6034, \beta_0 = -13.5047$. The spread is shown in Figure 2.

We then perform a final ADF test on the regression residuals. Referring to Table 1, we have a test-statistic of -4.6780 which is outside of our smallest critical bound of -2.58 indicating our residuals are in fact $I(0)$ stationary.

Next, we will look for the optimal threshold $\eta*$ by computing the return of 9 evenly spaced candidate thresholds. The values of each η_k and the optimal η_* is determined by equations 5 and 6 respectively. With a $\eta* = 0.0619$, our profit during the formation period is 1.4842. In Table 2, we see that this threshold performs well during the formation period. However, this does not guarantee the return during the testing period will be the same. A visualization of the Price, Spread, Trading Signals and Cumulative Return of the pair (UTX, TEL) is shown in Figure 3.

Now we can construct a portfolio of the top 5 pairs shown in Table 3 and weight each pair according to our dynamic asset allocation system.

5.2 Dynamic Asset Allocation

The asset allocation model was tested on the top 5 pairs using the procedure in Section 2.4. We used a sliding window of size $m = 200$ days, a single recurrent connection and bias, resulting in an input

Threshold	Profit
0.0619	1.4842
0.1239	1.3823
0.1858	1.4515
0.2477	1.1756
0.3097	0.7032
0.3716	0.6045
0.4336	0.6319
0.4955	0.6586
0.5574	0.3607

Table 2: Profit of 9 different thresholds η_k, we see that the optimal threshold is $\eta* = 0.0619$ with a cumulative return of $P_{AB} = 1.4842$

Pair	Optimal Threshold	Profit
(SIG, UAL)	0.1674	41.1803
(RSG, VRTX)	0.5276	32.8422
(ALK, EXPE)	0.2377	31.9745
(ILMN, TAP)	0.1080	30.9623
(TGNA, VRTX)	0.7432	25.2231

Table 3: Top 5 pairs, with their corresponding Optimal Threshold and Profit at the end of the Formation Period. All combinations of the 467 different securities listed on NYSE stock exchange were tested and ranked by Profit.

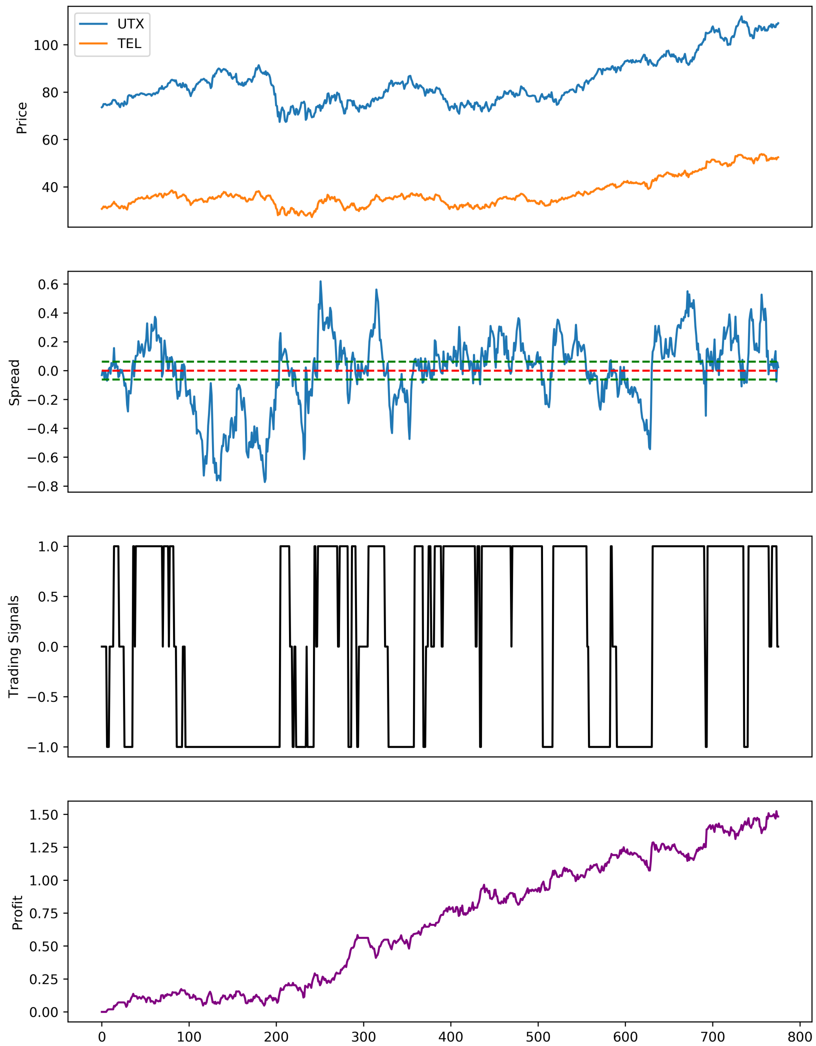

Figure 3: Formation Period Price, Spread, Signal and Return for pair (UTX, TEL). The green dotted line in the Spread graph represents the optimal threshold $\eta* = 0.0619$ and the point where we close the spread is denoted as the red line. Trading Signals correspond to the positions we take based on equation 2. Profit is calculated using equation 4

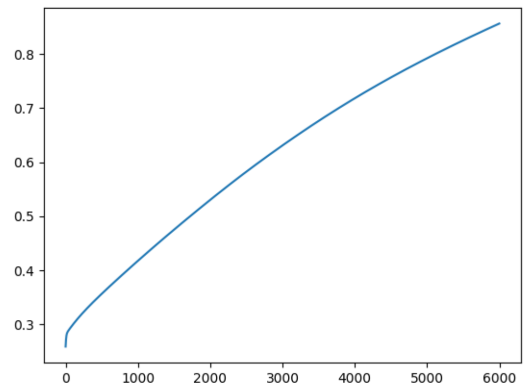

Figure 4: Sharpe Ratio for the top 5 pairs trained for 6000 epochs. We have a strictly increasing positive curve meaning after every iteration, our new weights are performing better on the formation data.

	Train
SR	0.8565
IR	0.8146

Table 4: Train-Test evaluation metrics for the top 5 pairs.

feature size of $n = 202$ for each pair. For model validation, we used a 70-30 train-test split on the historical data. The model is trained on the formation period of $F = 777$ days. Due to limitations in computational power, we did not test larger network architectures. With only 5 pairs and 200 features, the training of 6000 iterations took ≈ 1 hour. Through quick inspection, we found that the T-Bill rate (our baseline) in the year 2016 period is very different from that of the prior 6 years. For the purposes of this experiment, we omitted all of 2016 from our data.

Let us first inspect the Sharpe Ratio after each iteration to verify we have an increasing value. We do in fact see a strictly increasing slope in 4, and it seems to reach a good value of 0.8565 at the end of training. Although our model is performing well on our training data, we need to verify our performance on the testing data.

To truly test our model, we hold our parameters $\theta^{(p)}$ fixed and observed the performance of our system in the trading period of $T = 333$ days. If we look at the metrics in table 4, the results are far lower. We see that the cumulative return of our dynamic weightings is better than even weightings. With an $IR > 0$, it means we are performing better than the baseline T-Bill.

These results are disappointing, the difference in training and testing makes us believe that our model is over-fitting to the training data. A deeper investigation must be performed. Let us now take a closer look at the profit during our testing period. 5 We see that the dynamic weighting model is better than the even weightings at most times during the trading period. The worrying part is the long periods where both systems are at a loss. We will be observing losses in the period from 100-200 days before being able to see substantial gain. Only after 200 days are we able to continually beat the baseline. Referring to 5, an interesting observation is that if we were to use even weights for each pair, we would end up with less profit than investing in the T-Bill. This is a good indication that our model is in fact outperforming even weighting and is able to mitigate some of the losses.

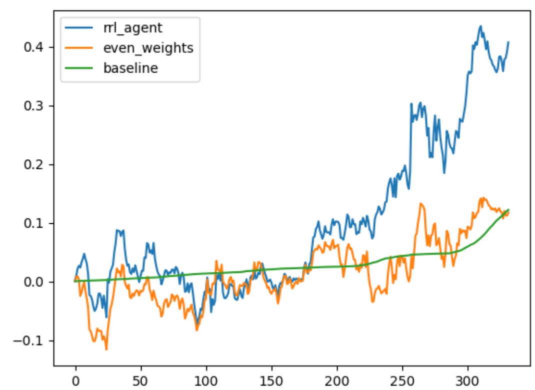

Figure 5: Profit of Dynamic Weighting(Blue), Even Weighting(Orange) and Baseline(T-Bill) over the testing period

Investment	Profit
Dynamic Weight	0.4071
Even Weight	0.1173
Baseline T-Bill	0.1217

Table 5: Profit Returns at the end of the trading period

6. Conclusion

We proposed and tested a methodology for selecting candidate pairs to construct a portfolio and a dynamic asset allocation model to provide better weightings of the pairs. The empirical results obtained in this project were competitive with the baselines provided.

We were able to apply a simple procedure to find a set of 5 pairs that were profitable in the formation period, but if applied to the trading period with even weights would have resulted in a loss. Further investigation of pairs selection could provide us with a better subset of pairs. A possible method could be to select pairs that satisfy both Engle and Granger Method and Johansen Test. A time-series approach to provide changing thresholds for opening positions proposed by [1]. This could result in a massive increase in return since the thresholds are not static and can vary based on the spread. Also, testing with smaller formation and trading periods may be interesting to look at. The number of pairs used could also be investigated further, 5 pairs were chosen for ease of computation.

For the dynamic asset allocation system, we were able to adapt the works of [7;8] to our pairs trading system, beating the even weightings of pairs and staying competitive with the 3-month Treasury Bill baseline. A different asset allocation technique could have been used as a benchmark instead of just the even weighting of our pairs. We could compare our model to the analytic solutions of the value and policy functions presented by [4]. Improvements on the model could be employing a more complex architecture, a neural network with multiple layers could result in a more generalizable model. The online implementation of the algorithm could have helped in performance since direct updates after each day during the trading period could have resulted in better results. With this online architecture, we could try out testing on different periods much easier. State of the art reinforcement learning methods like DDPG and A3C could result in a significant improvement.

References

[1] Robert J. Elliott, John Van Der Hoek *, and William P. Malcolm. Pairs trading. *Quantitative Finance*, 5(3):271–276, 2005.

[2] Robert F. Engle and C. W. J. Granger. Co-integration and error correction: Representation, estimation, and testing. *Econometrica*, 55(2):251–276, 1987.

[3] Richard C Grinold and Ronald N Kahn. *Active Portfolio Management: A Quantitative Approach for Producing Superior Returns and Controlling Risk; 2nd ed.* McGraw-Hill, New York, NY, 2000.

[4] Jakub W. Jurek and Halla Yang. Dynamic portfolio selection in arbitrage. *SSRN Electronic Journal*, 2007.

[5] Christopher Krauss. Statistical arbitrage pairs trading strategies: Review and outlook. *Journal of Economic Surveys*, 31(2):513–545, 2017.

[6] Jen-Wen Lin. Cointegration lecture notes - time series analysis, October 2017.

[7] J. Moody and M. Saffell. Learning to trade via direct reinforcement. *IEEE Transactions on Neural Networks*, 12(4):875–889, Jul 2001.

[8] John Moody, Lizhong Wu, Yuansong Liao, and Matthew Saffell. Performance functions and reinforcement learning for trading systems and portfolios. *Journal of Forecasting*, 17(5-6):441–470, 1998.

[9] Hossein Rad, Rand Kwong Yew Low, and Robert Faff. The profitability of pairs trading strategies: distance, cointegration and copula methods. *Quantitative Finance*, 16(10):1541–1558, 2016.

[10] William Sharpe. Mutual fund performance. *The Journal of Business*, 39, 1965.

[11] G. Vidyamurthy. *Pairs Trading: Quantitative Methods and Analysis*. Wiley Finance. Wiley, 2011.

Appendix A. Johansen Test

We will perform the Johansen Test on the same securities TEL, UTX as above to verify that they are in fact cointegrated. For the Trace Test (λ_{trace}) we want to verify that the test statistic lies outside the critical values of rank 0 and lies within $r \leq 1$. The same criteria apply for Maximum Eigenvalue Test (λ_{max}) as well.

We see from 6 that our two criteria are in fact verified. For Trace Test, we have that $25.49 > 24.60$, rejecting $r = 0$ and $1.80 < 7.52$ therefore it lies within our critical values, meaning our pair is rank 1. We can also verify from Maximum Eigenvalue Test that $23.69 > 20.20$ rejecting $r = 0$ and $1.80 < 7.52$ therefore accepting the pair is $r = 1$. Therefore, we have that both Johansen Test and Engle and Granger Method arrive at the same conclusion. This may not be the case for other pairs.

```
##
## ######################
## # Johansen-Procedure #
## ######################
##
## Test type: trace statistic , without linear trend and constant in cointegration
##
## Eigenvalues (lambda):
## [1]  3.010366e-02  2.323523e-03 -5.204170e-18
##
## Values of teststatistic and critical values of test:
##
##           test 10pct  5pct  1pct
## r <= 1 |  1.80  7.52  9.24 12.97
## r = 0  | 25.49 17.85 19.96 24.60
##
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
##
##              TEL.l1      UTX.l1    constant
## TEL.l1    1.0000000   1.0000000    1.000000
## UTX.l1   -0.6206601  -0.1247955    1.315821
## constant 15.0789673 -40.5396746 -141.273271
##
## Weights W:
## (This is the loading matrix)
##
##            TEL.l1       UTX.l1      constant
## TEL.d -0.00402142 -0.001926308 -4.557782e-18
## UTX.d  0.08917078 -0.002396918  1.739606e-16
```

(a) Trace Test

```
##
## ######################
## # Johansen-Procedure #
## ######################
##
## Test type: maximal eigenvalue statistic (lambda max) , without linear trend and 
##
## Eigenvalues (lambda):
## [1]  3.010366e-02  2.323523e-03 -5.204170e-18
##
## Values of teststatistic and critical values of test:
##
##           test 10pct  5pct  1pct
## r <= 1 |  1.80  7.52  9.24 12.97
## r = 0  | 23.69 13.75 15.67 20.20
##
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
##
##              TEL.l1      UTX.l1    constant
## TEL.l1    1.0000000   1.0000000    1.000000
## UTX.l1   -0.6206601  -0.1247955    1.315821
## constant 15.0789673 -40.5396746 -141.273271
##
## Weights W:
## (This is the loading matrix)
##
##            TEL.l1       UTX.l1      constant
## TEL.d -0.00402142 -0.001926308 -4.557782e-18
## UTX.d  0.08917078 -0.002396918  1.739606e-16
```

(b) Maximum Eigenvalue Test

Figure 6: Johansen Test Output