Pairs Trading Using Machine Learning

Anmol Agarwal
Computer Science Department
Delhi Technological University
Delhi, India
anmol22004@gmail.com

Arnav Preet
Computer Science Department
Delhi Technological University
Delhi, India
arnavpreet.bti@gmail.com

Rahul Gupta
Computer Science Department
Assistant Professor,
Delhi Technological University
Delhi, India
Ph.D Candidate at DTU,Delhi,India
rahulgupta100689@gmail.com

***Abstract*— 'Pairs Trading' is an investment strategy used by many Hedge Funds. Consider two similar stocks which trade at some spread. If the spread widens short the high stock and buy the low stock. As the spread narrows again to some equilibrium value, a profit result. This paper provides an analytical framework for such an investment strategy. We propose a mean reverting Gaussian Markov chain model for the spread which is observed in Gaussian noise. Predictions from the calibrated model are then compared with subsequent observations of the spread to determine appropriate investment decisions. The methodology has potential applications to generating wealth from any quantities in financial markets which are observed to be out of equilibrium.**

Keywords*— *Pairs trading; Hedge funds; Spreads; Mean reversion

I. INTRODUCTION

Pairs trading strategy is one of the most popular and successful quantitative methodology developed in 1980s by the team of scientists, mathematicians and computer scientists. The group is formed by the Wall Street quant Nunzio Tartaglia. The method was developed from the statistical model and computer algorithm without the decision from the human. The pairs trading strategy worked very well in the first move, but the performance was not consistent in long-term.

Pairs trading is a rule-based investment strategy exploiting the mispricing behaviour of securities. The statistical arbitrage is one type of mispricing. When the two time series data share the same characteristics such as same operating businesses, listed in the same industry or expose to the similar risk factors, they tend to move together, following the law of one price. When the prices of two securities move diversely from each other, they generate the price difference called "spread". The spread of the pair is considered as stationary and mean reverting process. The spread of the price is assumed to be white noise, which means that if the spread exceeds the statistical threshold, typically 2-standard deviation, the investors can open the position by opening the short position the winner stock and opening the long position the loser stock. When the spread converges to the mean, the position is closed and realized the profit (loss) of the pair.

II. PROBLEM STATEMENT

One of the big reasons that algorithmic trading has become so popular is because of the advantages that it holds over trading manually. The advantages of algo trading are related to speed, accuracy, and reduced costs.

Since algorithms are written beforehand and are executed automatically, the main advantage is speed. The speed at which these trades are made is measured in fractions of a second, faster than humans can perceive.

Trading with algorithms has the advantage of scanning and executing on multiple indicators at a speed that no human could do. Since trades can be analysed and executed faster, more opportunities are available at better prices.

Another advantage to algorithmic trading is accuracy. If a computer is automatically executing a trade, you get to avoid the pitfalls of accidentally putting in the wrong trade associated with human trades. With manual entries, it's much more likely to buy the wrong currency pair, or for the wrong amount, compared to a computer algorithm that has been double checked to make sure the correct order is entered.

One of the biggest advantages of algo trading is the ability to remove human emotion from the markets, as trades are constrained within a set of predefined criteria. Why this is an advantage is because humans trading are susceptible to emotions that lead to irrational decisions. The two emotions that lead to poor decisions that algo traders aren't susceptible to are fear, and greed.

Another advantage to algo trading is the ability to back test. It can be tough for traders to know what parts of their trading system work and what doesn't work since they can't run their system on past data. With algo trading, you can run the algorithms based on past data to see if it would have worked in the past. This ability provides a huge advantage as it lets the user remove any flaws of a trading system before you run it live.

Another advantage of automated trading is the reduced transaction costs. With algo trading, traders don't have to spend as much time monitoring the markets, as trades can be executed without continuous supervision. The dramatic time reduction for trading lowers transaction costs because of the saved opportunity cost of constantly monitoring the markets.

III. DATASET

We obtain the data set from finance.yahoo.com. The timeframe for our data set is 1 day, ranging from 01/01/2015 to 31/12/2019. First, we need to do some data preprocessing because pairs trading requires the data of two securities must be consistent. By consistent we mean that the data and time of every feed of both securities should be an exact match.

The original data set has 1027 securities from NSE. Each security consists data from 01/01/2015 to 31/12/2019 and closing price as their attribute. Another factor involved for calculations is type of Industry they belong to. After processing the data to percent change on daily basis we then used that data for further study.

IV. CONCEPT USED

Pairs trading is a nice example of a strategy based on mathematical analysis.

The principle is as follows: Let's say you have a pair of securities X and Y that have some underlying economic link. An example might be two companies that manufacture the same product, for example Pepsi and Coca Cola. You expect the spread (ratio or difference in prices) between these two to remain constant with time. However, from time to time, there might be a divergence in the spread between these two pairs. The divergence within a pair can be caused by temporary supply/demand changes, large buy/sell orders for one security, reaction for important news about one of the companies, and so on. When there is a temporary divergence between the two securities, i.e. one stock moves up while the other moves down, the pairs trade would be to short the outperforming stock and to long the underperforming one, betting that the "spread" between the two would eventually converge. Pairs trading is a market neutral trading strategy enabling traders to profit from virtually any market conditions: uptrend, downtrend, or sideways movement.

Cointegration between Stocks Because two cointegrated time series (such as X and Y above) drift towards and apart from each other, there will be times when the spread is high and times when the spread is low. We make a pairs trade by buying one security and selling another. This way, if both securities go down together or go up together, we neither make nor lose money—we are market neutral. Going back to X and Y above that follow Y = α X + e, such that ratio (Y/X) moves around it's mean value α, we make money on the ratio of the two reverting to the mean. In order to do this we'll watch for when X and Y are far apart, i.e. α is too high or too low:

- Going Long the Ratio This is when the ratio α is smaller than usual and we expect it to increase. In the above example, we place a bet on this by buying Y and selling X.
- Going Short the Ratio This is when the ratio α is large and we expect it to become smaller. In the above example, we place a bet on this by selling Y and buying X.

Note that we always have a "hedged position": a short position makes money if the security sold loses value, and a long position will make money if a security gains value, so we're immune to overall market movement. We only make or lose money if securities X and Y move relative to each other.

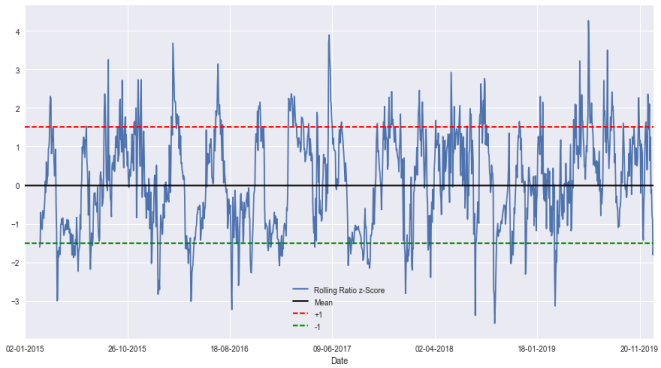

Fig.1. Spread of ratio of two Equities

V. STRATEGY

Go "Long" the ratio whenever the z-score is below -1.5.
Go "Short" the ratio when the z-score is above 1.5.
Exit positions when the z-score approaches zero.
This is just the tip of the iceberg, and only a very simplistic example to illustrate the concepts.
In practice you would want to compute a more optimal weighting for how many shares to hold for S1 and S2
You would also want to trade using constantly updating statistics.
In general, taking a statistic over your whole sample size can be bad. For example, if the market is moving up, and both securities with it, then your average price over the last 3 years may not be representative of today. For this reason, traders often use statistics that rely on rolling windows of the most recent data.
Instead of using ratio values, let's use 1d Moving Average to compute to z score, and the 30d Moving Average and 30d Standard Deviation as the mean and standard deviation.

Fig. 2. Signal on chart with mean reversion technique

VI. PROPOSED PAIRS SELECTION FRAMEWORK

At this stage, we aim to explore how one investor may find promising pairs without exposing himself to the adversities of the common pairs searching techniques. On the one hand, if the investor limits its search to securities within the same sector, he is less likely to find pairs not yet being traded in large volumes, leaving a small margin for profit. But on the other hand, if the investor does not impose any limitation on the search space, he might have to explore excessive combinations and possibly find spurious relations. We intend to reach an equilibrium with the application of an Unsupervised Learning algorithm, on the expectation that it will infer meaningful clusters of assets from which to select the pairs.

A. **Dimensionality reduction-** The first step towards this direction consists in finding a compact representation for each asset, starting from its price series. The application of PCA is proposed. PCA is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of linearly uncorrelated variables, the principal components. Each component can be seen as representing a risk factor. Using the price series may not be appropriate due to the underlying time trends. The number of principal components used defines the number of

features for each asset representation. Considering that an Unsupervised Learning algorithm will be applied to these data, the number of features should not be large. High data dimensionality presents a dual problem. The first being that in the presence of more attributes, the likelihood of finding irrelevant features increases. Additionally, there is the problem of the curse of dimensionality, caused by the exponential increase in volume associated with adding extra dimensions to the space. According to Berkhin (2006), this effect starts to be severe for dimensions greater than 15. Taking this into consideration, the number of dimensions is upper bounded at 15 and chosen empirically.

B. **Unsupervised learning-** Having constructed a compact representation for each asset, a clustering technique may be applied. To decide which algorithm is more appropriate, some problem-specific requisites are first defined: No need to specify the number of clusters in advance. No need to group all securities. Strict assignment that accounts for outliers. No assumptions regarding the clusters' shape. By making the number of clusters data-driven, we introduce as little bias as possible. Furthermore, outliers should not be incorporated in the clusters, and therefore grouping all assets should not be enforced. In addition, the assignment should be strict, otherwise the number of possible pair combinations increases, which is conflicting with the initial goal. Finally, due to the nonexistence of prior information that indicates the clusters should be regularly shaped, the selected algorithm should not assume this. Taking into consideration the previously described requirements, a density-based clustering algorithm seems an appropriate choice. It forms clusters with arbitrary shapes, and thus no gaussianity assumptions need to be adopted. It is naturally robust to outliers as it does not group every point in the data set. Furthermore, it requires no specification of the number of clusters. The DBSCAN algorithm is the most influential in this category. Briefly, DBSCAN detects clusters of points based on their density. To accomplish that, two parameters need to be defined: e, which specifies how close points should be to each other to be considered ‘‘neighbours”, and minPts, the minimum number of points to form a cluster. From these two parameters, in conjugation with some concepts that we omit here2 , clusters of neighbouring points are formed. Points falling in regions with less than minPts within a circle of radius e are classified as outliers, hence not affecting the results. Nevertheless, DBSCAN still carries one drawback. The algorithm is appropriate under the assumption that clusters are evenly dense. However, if regions in space have different densities, a fixed e may be well adapted to one given cluster density, but it might be unrealistic for another.

Fig. 3. Clusters with varying density

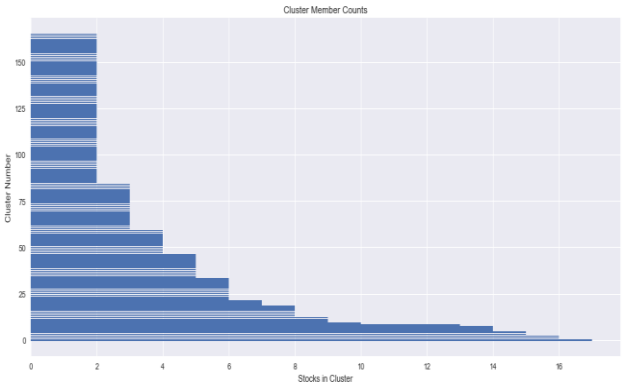

Fig. 4. Clusters Member counts

C. **Pairs selection criteria-** Having generated the clusters of assets from which to find the candidate pairs, it is still necessary to define a set of rules to select those eligible for trading. It is critical that the pairs' equilibrium persists. To enforce this, we propose the unification of methods applied in separate research work. According to the proposed criteria, a pair is selected if it complies with the four conditions described next. First, a pair is only selected if the two securities forming the pair are cointegrated. To test this condition, we propose the application of the Engle-Granger test, Engle and Granger (1987), due to its simplicity. To deal with the test reliance on the dependent variable, we propose that the test is run for both possible selections of the dependent variable and that the combination generating the lowest t-statistic is selected. Because two cointegrated time series (such as X and Y above) drift towards and apart from each other, there will be times when the spread is high and times when the spread is low. We make a pairs trade by buying one security and selling another. This way, if both securities go down together or go up together, we neither make nor lose money—we are market neutral.

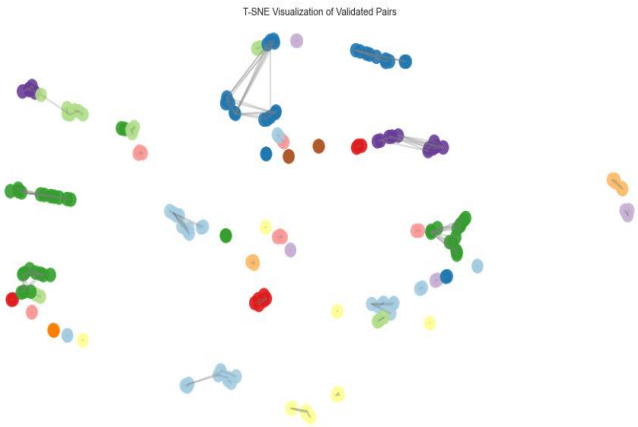

Fig. 5. T-SNE Visualization of valid clusters

Stocks Time Series For Some Cluster :-

PGHH and HINDUNILVR are highly cointegrated as they both belongs to FMCG sector and passed our pair test to get traded.

Fig. 6. PGHH VS HINDUNILVR

CANBK, INDIANB AND PNB are PSUs bank and are higly cointegrated to get traded.

Fig. 7. CANBK VS INDIANB VS PNB

HCLTECH, INFY, OMAXE AND WIPRO are from IT sector and are cointegrated with each other. With this 6 pairs are formed to be traded according to our strategy.

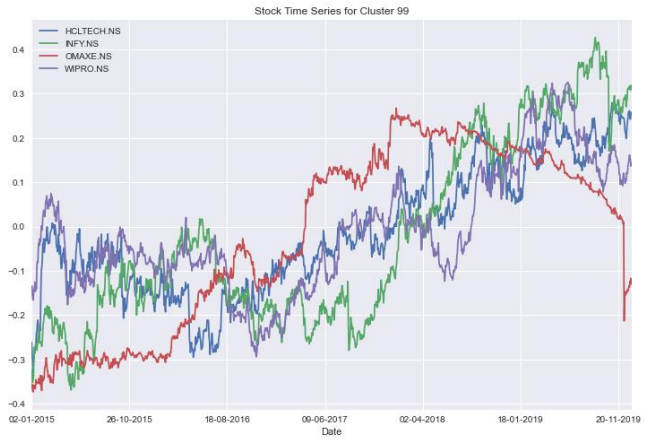

Fig. 8. WIPRO VS INFY VS OMAXE VS HCLTECH

VII. SYSTEM REQUIREMENTS

Hardware Requirement:

Minimum 128 MB of RAM, 256 MB recommended.

110 MB of hard disk space required; 40 MB additional hard disk space required for installation (150 MB total).

Software Requirement:

This project has been designed for windows and other platforms. Python Language is used.

Important Library Used: -

- Yfinance
- Scipy
- Statsmodels
- Sklearn

Tools Used: -

- Jupyter Notebook
- MS Word
- MS PowerPoint

VIII. IMPLEMENTATION

Trading Algorithm

```
def trade(S1, S2,money, window1=1, window2=30):
    # If window length is 0, algorithm doesn't make
sense, so exit
    if (window1 == 0) or (window2 == 0):
        return 0
    # Compute rolling mean and rolling standard devi
ation
    ratios = S1/S2
    ma1 = ratios.rolling(window=window1,
                                 center=False).mean()
    ma2 = ratios.rolling(window=window2,
                                 center=False).mean()
    std = ratios.rolling(window=window2,
                            center=False).std()
```

```
    zscore = (ma1 - ma2)/std
    # Simulate trading
    # Start with no money and no positions
    countS1 = 0
    countS2 = 0
    first_time = [0,0]
    profit=0
    c1=0
    c2=0
    mon_list = pd.Series(index=returns.index)
    mon_list[mon_list.index==returns.index[0]] = mon
ey
    for i in range(len(ratios)):

        # Sell short if the z-score is > 1,5
        if zscore[i] > 1.5 and money>max(S1[i],S2[i]
) and c1==0:
            countS1 -= int((money/2)/S1[i])
            countS2 += int((money/2)/S2[i])
            first_time = [S1[i],S2[i]]
            money -= -countS1*S1[i] +  countS2*S2[i]
            c1=1
        # Buy long if the z-score is < -1.5
        elif zscore[i] < -
1.5 and money>max(S1[i],S2[i]) and c2==0:
            countS1 += int((money/2)/S1[i])
            countS2 -= int((money/2)/S2[i])
            first_time =  [S1[i],S2[i]]
            money -= countS1*S1[i] - countS2*S2[i]
            c2=1


        # Clear positions if the z-score between -
.1and .1
        elif abs(zscore[i]) < 0.1:
            profit= countS1*(S1[i]-
first_time[0]) + countS2*(S2[i]-first_time[1])
            if(c1==1):
                money += -
countS1*first_time[0] + countS2*first_time[1] + prof
it
                c1=0
            if(c2==1):
                money += countS1*first_time[0] -
 countS2*first_time[1] + profit
                c2=0
            countS1 = 0
            countS2 = 0
            mon_list[mon_list.index==data.index[i]]
= money

    return mon_list
```

IX. Result

After applying this pairs strategy on NSE equities over the span of 5 years from 2015 to 2019 we got some handsome profits. From this technique we have found 48 pairs outperforms NIFTY-50(buy and hold for 5 years) with maximum profit of 467% on single pair.

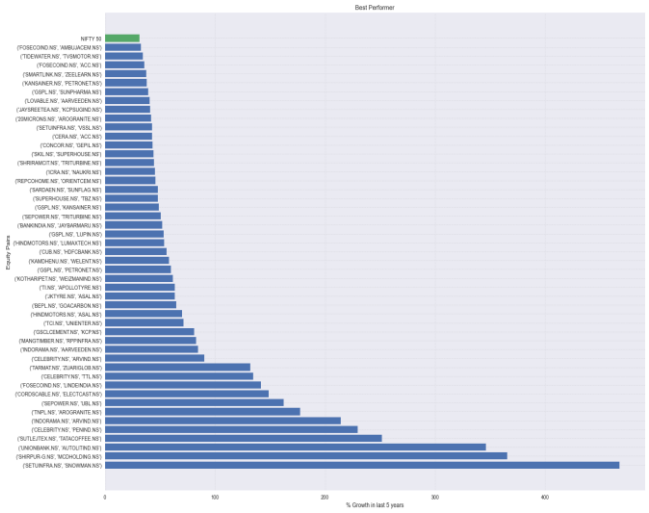

Fig. 9. 48 pairs outperform NIFTY-50

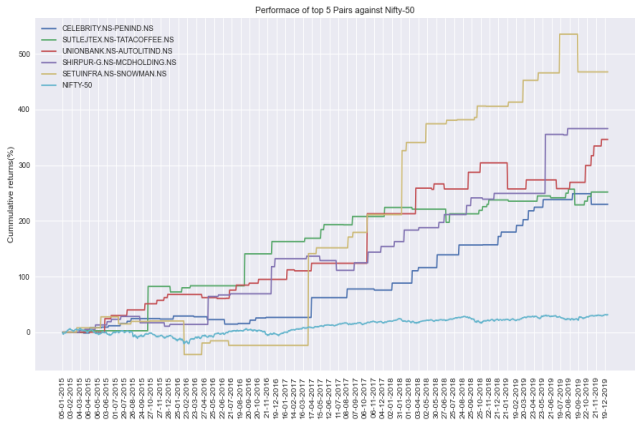

Fig 10. Top 5 performer against Nifty-50

X. Conclusion

In this project, we utilize both price and technical indicators for our pairs trading strategy while most of previous work only uses price to build pairs trading strategy. We also devise two new metrics to measure our strategy because the traditional back-testing for return prediction is not our focus and can be not very useful since it involves much more practical problems, such as slippage and transaction cost. Instead, we want to focus more on directional prediction. The results show that our strategy is moderately predictive, and has slightly better performance on predicting spread change direction than predicting profitable trades

XI. Future Work

We have found a nice number of pairs to use in a pairs trading strategy. We will need to take some special precautions in the Portfolio Construction stage to avoid excessive concentration of any one stock. Although all pairs does not perform as expected and showed erosive nature to our capital . There is need to optimised the algorithm with additional information on financial health or future expansion or quarterly results to cluster down the stocks more accurately.

Happy hunting for pairs!

REFERENCES

[1] Vidyamurthy, G. (2004) Pairs Trading - Quantitative Methods and Analysis, John Wiley & Sons, Inc., New JerseyJ.

[2] Gatev, E., W.N. Goetzmann, and K.G. Rouwenhorst "Pairs trading: Performance of a relative-value arbitrage rule." Review of Financial Studies, Vol. 19 (2006), pp. 797–827

[3] Ghosh, Indranil and Chaudhuri, Tamal and Singh, Priyam, Application of Machine Learning Tools in Predictive Modeling of Pairs Trade in Indian Stock Market (April 10, 2018). The IUP Journal of App

[4] Riedinger, Stephanie, Demystifying Pairs Trading: The Role of Volatility and Correlation (June 7, 2017)

[5] Caldeira, Joao and Moura, Guilherme Valle, Selection of ˜ a Portfolio of Pairs Based on Cointeg

[6] Rudy, J., Dunis, C., Giorgioni, G. and Laws, J. (2010) Statistical Arbitrage and High-Frequency Data with an Application to Eurostoxx 50 Equities. Social Science Electronic Publishing, Rochester.

[7] Huang, C. F., Hsu, C. J., Chen, C. C., Chang, B. R., & Li, C.- A. (2015). An intelligent model for pairs trading using genetic algorithms. Comp

[8] Dunis, Christian and Laws, Jason and Rudy, Jozef, Mean Reversion Based on Autocorrelation: A Comparison Using the S&P 100 Constituent Stocks and the 100 Most Liquid ETFs (January 31, 2011). ETF Risk, 2013, October, 36-41.

[9] Dingli, Alexiei & Fournier, Karl. (2017). Financial Time Series Forecasting – A Deep Learning Approach. International Journal of Machine Learning and Computing. 7. 118-122. 10.18178/ijmlc.2017.7.5.632.

[10] Perry J. Kaufmann, Alpha Trading: Profitable Strategies That Remove Directional Risk, Wiley; 1 edition (March 8, 2011)

[11] Sinclair, E. 2013. Volatility Trading, Hoboken, New Jersey, John Wiley and Sons.

[12] Board of Governors of the Federal Reserve System (US) (2019). 3-Month Treasury Bill: Secondary Market Rate. https://fred.stlouisfed.org/series/TB3MS. Accessed: 2019-07-11.

[13] Sutskever, I., Vinyals, O., & Le, Q.V. (2014). Sequence to sequence learning with neural networks. CoRR, abs/1409.3215