CSE6242 2024Spring - Final Report
SPONGE and Beyond: Signed Weighted Graph Stock Clustering for Dynamic Market Forecasting

Team 77 - Jonathan Maniery, Andres Quinonez, Cleo Zhang, and Dhaval Lokagariwar

1 INTRODUCTION

With over 10,000 individual stocks, it can be daunting for individuals to select an appropriate blend for their portfolio. Through this project, we aim to implement an equity investment strategy that uses clustering on correlation matrices to identify clusters of securities driven by similar economics. Our proposed strategy then uses the cluster level returns to form price indices and implements a time-series momentum strategy using supervised learning. Prior contributions [3] to the statistical arbitrage literature use correlation graph clustering to form mean reverting portfolios, but to our knowledge none consider trend following strategies based on the mean cluster returns. Our approach treats clusters as individual portfolios of stocks and hopefully captures some element of sector timing where the underlying driver is the economic environment.

2 PROBLEM DEFINITION

Instead of investing in the over-all stock market we propose investing in a cluster based investment strategy identified by the SPONGE clustering algorithm [1]. Specially, we propose taking an equal weight in the top quartile of clusters selected by supervised machine learning algorithms using trend following features as inputs and outputs as the next day's forecasted risk-adjusted return. Our hope is this strategy will capture time-dependent dynamics driven by both economic cycles and idiosyncratic evolutions. Examples of these could be out performance of cyclical stocks in an economic expansion, or the current out-performance of AI stocks. Intuitively instead of trying to make a long term forecast of a given sector or groups of stocks we just wait until trends surface and ride the wave. Traditionally similar sector momentum strategies rely on using broad classifications, such as energy or retail while we hope to capture more granular similarities automatically with clustering.

3 LITERATURE SURVEY

Our literature review synthesizes contributions to statistical arbitrage, trend following and clustering in quantitative investing strategies. Initially, it examines pairs trading's profitability and its foundational strategies[5], tracing back to the 1980s at Morgan Stanley [6], and underscores its alignment with the Law of One Price [2]. Subsequent studies delve into comparing pairs trading strategies [12], where distance and co-integration [5] approaches are favored over the copula method, particularly under market volatility. Innovative research applies machine

learning to arbitrage in the SP 500 [9], revealing that ensemble methods surpass single-model performances, thus questioning market efficiency. Investigations using PCA [1] to dissect stock returns highlight strategies that yield significant alphas, thriving amid market fluctuations. A novel method employing deep reinforcement learning[8] to optimize pairs trading by adaptive boundary settings shows superior results over conventional strategies. The addition of a paper on graph clustering algorithms for portfolio construction [15] marks a significant leap, demonstrating over 10 percent annual returns and notable Sharpe ratios by grouping correlated stocks for mean-reverting arbitrage strategies. In portfolio construction, clustering outperforms traditional methods by revealing complex company relationships and balancing risk and performance[4][15][14]. It taps into advanced algorithms, notably machine learning, to advance statistical arbitrage. Wu[15] and Soleymani[14] utilized K-means for portfolio creation, with Wu identifying optimal clusters for Sharpe ratio and returns when $k > 30$, and Soleymani optimizing CVaR with a 12-stock portfolio. Our method extends K-means to eigenvectors of graphs, enhancing forecast accuracy by using cluster information, which contains less noise compared to using data from all stocks[13], highlighting the importance of data selection in trading strategies. With regards to trend following we take inspiration from [11] which show persistence in returns over 1 to 12 months across equity indices, currencies, commodities and bond futures consistent with theories of initial under reaction and delayed over-reaction to news. Additionally they find that trend following speculators profit at the expense of hedgers. AQR[7] does an extensive study to examine whether the strong performance of trend following is a statistical fluke of recent decades or a robust persistent phenomenon. They gather data back to 1880 and find that the strategy was profitable consistently over the next 110 years. And finally recent approaches[10] in trend-following use deep learning to output asset level position sizes directly using custom loss functions.

4 PROPOSED METHOD

In line with existing research [3] we plan to estimate clusters of stocks based on the correlation matrix of residual returns using the SPONGE algorithm. SPONGE is used specifically because our correlation matrix is a signed weighted graph. The output of this step is a matrix of daily returns of shape (N, k) where N are the daily time-stamps and k is the number of clusters. After aggregating daily cluster returns we plan to develop time-series momentum strategies on a cluster level. Additionally we will have a visualization layer to explore the output. The steps are as follows and we view 5 and 6 as our key innovations:

1. Compute residual returns for our universe of stocks. Specifically, let y_i^t be the residual return of stock i at time t and mkt be the return of the universe and ra_i^t the return of stock i. Then $y_i^t = ra_i^t - \theta mkt_i^t$ where theta is estimated using ordinary least squares over some formulation period τ. If time permits also experiment computing residual returns using PCA.

2. Compute a rolling correlation matrix of residual returns over window $\tau = 60$ days. This is our graph where stocks are nodes and edges are pairwise correlations. Using the methodology from lecture we plan to store the graph in a sqlite3 database.
3. Compute cluster assignments on the graph using SPONGE, and store as a column in our database. We should have a cluster assignment for each stock on each date. We will experiment with fixed numbers of clusters and the eigenvalue cumulative variance methodology for cluster number selection. Ultimately we chose $K = 20$, a fixed number of clusters to insure stability over-time vs a dynamically changing number of clusters.
4. Compute time-series of average cluster returns using the full total returns on a stock level. In this step we group by cluster assignment and date, compute the mean on the returns.
5. Set up the forecasting problem described in section 4, specifically we apply machine learning to forecast one-day forward risk adjusted returns and formulate portfolios and returns based on the time-series momentum equation.
6. Visualize the signed weighted graph using a d3.js application. Ideally we use node.js to connect our sqlite3 database and allow the user to select a date and then view the graph. We plan to color nodes with cluster assignments and allow selection for a subset of clusters as the graph will have thousands of stocks. Regarding this item our final visualization solution is to visualize the correlation structure of cluster correlation matrix and not the individual stocks while having functionality to view the stocks inside each cluster. Ultimately this is consistent with our methodology of forecasting on a cluster level and not a stock level thus reducing dimensionality from NxN to KxK.
7. Using streamlit visualize the results of our clustering strategy over time. Specifically we show the line plots of strategy returns for each machine learning algorithm with a multi-selector. This allows users to compare strategies across models and across quartiles. Recall that we rank clusters on expected returns from the models and go long the top quartile (out of $K = 20$), thus our expectation is that higher forecasted cluster returns result in higher annual total returns.

5 EXPERIMENTS AND EVALUATION

A key part of our innovation is building a cluster investment strategy. Instead of trying to find cheap and expensive stocks within clusters in the typical statistical arbitrage spirit we wish to take over/weight positions on a cluster level. This will likely be similar to a sector rotation strategy with the hope of dynamically capturing the changing market environment. In-line with prior research we plan on setting up the cluster mean return forecasting problem as a time-series momentum problem[10]. We will predict one-day forward risk adjusted mean cluster returns using time-series features such as trailing risk adjust returns and moving average differences over various past frequencies. Formally, the combined returns of a time series momentum strategy (TSMOM) can be expressed by a trading rule X_t learned by a statistical model. Below,

N is the total number of clusters and the left hand side is the return of the strategy from day t to $t+1$. σ_{tgt} is a volatility target and σ_t^i is the volatility of cluster i at time t.

$$r_{t,t+1}^{TSMOM} = \frac{1}{N_t} \sum_{i=1}^{Nt} X_t^i \frac{\sigma_{tgt}}{\sigma_t^i} r_{t,t+1}^i$$

X_t is the trading rule (long or short) and is the sign of a trend estimation Y_t^i produced by a supervised learning model. Let Y_t^i be the trend estimate for cluster i and time t. The trend estimation is = $Y_t^i = f(u_t^i; \theta)$ where f is the output of a machine learning model and u_t^i is a feature vector of cluster i at time t, θ are the model parameters resulting in predictions.

Ultimately we decided to take a simple portfolio construction approach different than equation 1 for several reasons. Firstly, since we are operating in a single asset class volatility tends to be similar amongst clusters. Thus the volatility scaling results in approximately equal weight between clusters. With this in mind we create equal weight cluster portfolios by compute cross-sectional quartiles based on the model predictions, and go long the top five clusters with the highest expected risk-adjusted return over the next day with daily rebalancing.

Experiment and evaluation steps:

1. Hypothesis-Over/under weight mean cluster forecasting strategy can outperform the mean return of our investment universe while taking the same amount of risk.
2. Design matrix X are time-series momentum features[10] for each cluster on each day and targets are their associated one-day forward risk adjusted return. This is a regression problem.
3. Expanding window walk-forward cross validation at 5 year increments using a 90/10 split for training and validation (tuning hyper-parameters), testing set five years forward. For parameter search we use randomized grid-search in sklearn. This method results in four splits across our dataset with predictions being saved for the next five years before re-tuning.
4. Modeling-Use linear models as a baseline and compare with more complex machine models (e.g ElasticNet vs Random Forest)
5. Feature selection - we use SelectFromModel in sklearn for feature selection. Specifically we add the object in our pipelines. For RandomForest we take the features above the mean in the built in feature importance method. For ElasticNet we chose the features with an absolute value of the linear coefficient greater than zero.
6. We will have multiple evaluation criteria, for tuning in the walk-forward step we will use mean-squared error (MSE). For investment performance we will evaluate metrics like total return and Sharpe ratio. We expect to see a linear relationship between the accuracy of the forecasting models and investment performance.

Below are the results of the strategy, which are based on the out-of-sample returns previously

mentioned. The findings are promising and align with our initial hypothesis. The top three models, all from the upper quartile of ranked predictions (with the index starting at 0), show that the non-linear RandomForest outperforms ElasticNet by nearly 1 percent per annum. This indicates that there are beneficial non-linear interactions within our feature set. Coming in third place is RandomForest with feature selection, which suggests that omitting features is not advantageous. The leading three strategies surpass the market by 1-2 percent, indicating that a dynamic modeling approach offers advantages and can identify clusters that typically outperform the overall market. In our analysis, the market is represented by an equally weighted portfolio comprising all 20 clusters. Consistent with previous observations, we note that annual volatility is similar across all strategies, thus facilitating reasonable return comparisons. The Sharpe ratio is also provided for further validation (return/risk). Moreover, cross-validation results are presented across four time-series split folds. In line with the returns, RandomForest exhibits a lower average MSE than the linear ElasticNet. However, differing from the returns, the RandomForest with feature selection registers the lowest MSE in our testing. We hypothesize that the randomness in returns may be a contributing factor. Two models with comparable accuracy may differ significantly if one model misses the largest single-day return while the other maintains a long position, thereby exerting a substantial impact on cumulative returns.

strategy	Annual_Return	Annual_Volatility	MDD	Sharpe	Sortino
preds_RF_Q_3.0	0.1095	0.2207	-0.5797	0.5800	0.8653
preds_eNet_Q_3.0	0.1006	0.2216	-0.5362	0.5423	0.8078
preds_RF_FS_Q_3.0	0.0931	0.2170	-0.5820	0.5180	0.7647
preds_eNet_Q_2.0	0.0916	0.2113	-0.6483	0.5198	0.7531
preds_eNetPCA_Q_1.0	0.0864	0.2121	-0.6183	0.4972	0.6934
preds_eNetPCA_Q_3.0	0.0843	0.2166	-0.5344	0.4810	0.7050
preds_RF_Q_1.0	0.0839	0.2098	-0.6401	0.4891	0.6905
market	0.0823	0.1938	-0.6106	0.5053	0.7020
preds_eNetPCA_Q_2.0	0.0781	0.2165	-0.6611	0.4553	0.6556
preds_RF_FS_Q_0.0	0.0773	0.2140	-0.5570	0.4543	0.6480
preds_RF_FS_Q_2.0	0.0740	0.2103	-0.6792	0.4451	0.6223
preds_eNet_Q_1.0	0.0734	0.2120	-0.5877	0.4405	0.6054
preds_RF_FS_Q_1.0	0.0632	0.2093	-0.6359	0.3979	0.5598
preds_RF_Q_0.0	0.0632	0.2166	-0.6733	0.3901	0.5620
preds_eNetPCA_Q_0.0	0.0624	0.2124	-0.6516	0.3904	0.5591
preds_RF_Q_2.0	0.0558	0.2079	-0.5613	0.3654	0.5119
preds_eNet_Q_0.0	0.0465	0.2112	-0.6818	0.3204	0.4518

Model	MeanCV_Score	StdCV_Score	Coef_of_Variation
RandomForestFS	1.4476	0.0494	29.2996
RandomForest	1.4488	0.0508	28.5246
ElasticNetFS	1.4512	0.0526	27.5778
ElasticNet	1.4518	0.0533	27.2314

Table 1—Cross-Validation Results

6 CONCLUSION AND DISCUSSION

In line with our hypothesis, we have identified machine-learning cluster strategies that outperform the market by holding an equal weight of just five clusters each day over time. Additionally, we have observed that quartile sorts based on predictions across strategies tend to be mostly monotonic. The top quartiles consistently outperform the bottom quartiles by a substantial margin. Our results suggest the presence of cyclical and idiosyncratic factors affecting subsets of stocks in the market, offering potential value to investors who tactically tilt towards those sectors at opportune times through a systematic forecasting system anchored in trend-following features. Whether through statistical models or a discretionary process, our findings indicate benefits in over-weighting or under-weighting various sectors as opposed to maintaining a static long position in the overall market. However, our approach's limitations include not accounting for turnover and transaction costs in our preliminary analysis. Moreover, this strategy demands a high level of technical expertise for implementation, rendering it unsuitable for the average investor. Nevertheless, there is a prospect for asset managers to package this strategy into an ETF, which could make it more accessible to retail investors. Future developments could involve applying deep learning and custom loss functions to our forecasting challenges, aiming to directly optimize returns and Sharpe ratios, and using the methodology from the original paper introducing SPONGE for statistical arbitrage to forecast residual returns.

All team members have contributed a similar amount of effort.

REFERENCES

[1] Marco Avellaneda and Jeong-Hyun Lee. "Statistical arbitrage in the US equities market". In: *Quantitative Finance* 10.7 (2010), pp. 761–782.

[2] John Baffes. "Some further evidence on the law of one price: The law of one price still holds". In: *American Journal of Agricultural Economics* 73.4 (1991), pp. 1264–1273.

[3] Álvaro Cartea, Mihai Cucuringu, and Qi Jin. "Correlation Matrix Clustering for Statistical Arbitrage Portfolios (September 3, 2023)". In: *4th ACM International Conference on AI in Finance,Available at SSRN: https://ssrn.com/abstract=4560455* (2023).

[4] Marcos Lopez De Prado. "Building diversified portfolios that outperform out of sample". In: *The Journal of Portfolio Management* 42.4 (2016), pp. 59–69.

[5] Robert F Engle and Clive WJ Granger. "Co-integration and error correction: representation, estimation, and testing". In: *Econometrica: journal of the Econometric Society* (1987), pp. 251–276.

[6] Evan Gatev, William N Goetzmann, and K Geert Rouwenhorst. "Pairs trading: Performance of a relative-value arbitrage rule". In: *The Review of Financial Studies* 19.3 (2006), pp. 797–827.

[7] Brian Hurst, Yao Hua Ooi, and Lasse Heje Pedersen. "A century of evidence on trend-following investing". In: *Available at SSRN 2993026* (2017).

[8] Taewook Kim and Ha Young Kim. "Optimizing the pairs-trading strategy using deep reinforcement learning with trading and stop-loss boundaries". In: *Complexity* 2019 (2019), pp. 1–20.

[9] Christopher Krauss, Xuan Anh Do, and Nicolas Huck. "Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500". In: *European Journal of Operational Research* 259.2 (2017), pp. 689–702.

[10] Bryan Lim, Stefan Zohren, and Stephen Roberts. *Enhancing Time Series Momentum Strategies Using Deep Neural Networks*. 2020. arXiv: 1904.04912 [stat.ML].

[11] Tobias J Moskowitz, Yao Hua Ooi, and Lasse Heje Pedersen. "Time series momentum". In: *Journal of financial economics* 104.2 (2012), pp. 228–250.

[12] Hossein Rad, Rand Kwong Yew Low, and Robert Faff. "The profitability of pairs trading strategies: distance, cointegration and copula methods". In: *Quantitative Finance* 16.10 (2016), pp. 1541–1558.

[13] Javier Vásquez Sáenz. *Data vs. information: Using clustering techniques to enhance stock returns forecasting*. Elsevier, 2023, pp. 1–10.

[14] Fazlollah Soleymani and Mahdi Vasighi. "Efficient portfolio construction by means of CVaR and k-means++ clustering analysis: Evidence from the NYSE". In: *International Journal of Finance & Economics* 27.3 (2022), pp. 3679–3693.

[15] Dingming Wu, Xiaolong Wang, and Shaocong Wu. "Construction of stock portfolios based on k-means clustering of continuous trend features". In: *Knowledge-Based Systems* 252 (2022), p. 109358.

7 FEATURE DEFINITIONS (NORMALIZED WITH ROBUST-SCALER)

- Feature 1d RA(risk adjusted) - Daily return normalized by trailing 63-day standard deviation of returns
- Feautre 1M RA(risk adjusted) - Monthly return normalized by trailing 63-day standard deviation of returns scaled by the square root of time (20 days)
- Feautre 1Q RA(risk adjusted) - Quarterly return normalized by trailing 63-day standard deviation of returns scaled by the square root of time (63 days)
- Feautre 6M RA(risk adjusted) - Semi-annual return normalized by trailing 63-day standard deviation of returns scaled by the square root of time (124 days)
- Feautre 12M RA(risk adjusted) - Annual return normalized by trailing 63-day standard deviation of returns scaled by the square root of time (252 days)
- Feautre MACD Short - Difference between the 8 day moving average of price and the 24 day moving average of price normalized by the 63 day stdev
- Feautre MACD Medium - Difference between the 16 day moving average of price and the 48 day moving average of price normalized by the 63 day stdev
- Feautre MACD Long - Difference between the 32 day moving average of price and the 96 day moving average of price normalized by the 63 day stdev