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

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

Georgia Tech

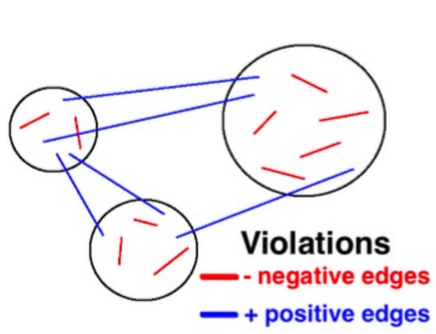
INTRODUCTION

What is the problem? As a popular trading strategy, statistical arbitrage has evolved over decades, which typically rely on correlation-based approaches and may overlook negative relationships or fail to distinguish between positive and negative interactions. The SPONGE algorithm, a graph clustering implementation for constructing portfolios, marks a major step-forward to the traditional statistical arbitrage by addressing those limitations. In addition, the algorithm implementation is difficult for many people to follow, and thus creates a high technical barrier to keep potential investors out of the market.

Why is it important and why should we care? SPONGE's superior returns to traditional statistical arbitrage are certainly a good thing for investors. And our project is important in lowering the investment threshold and encouraging potential investors to enter the market. More investors mean more money in the flow, which will be of great benefit to the of the financial development.

APPROACHES

What are they and how do they work?

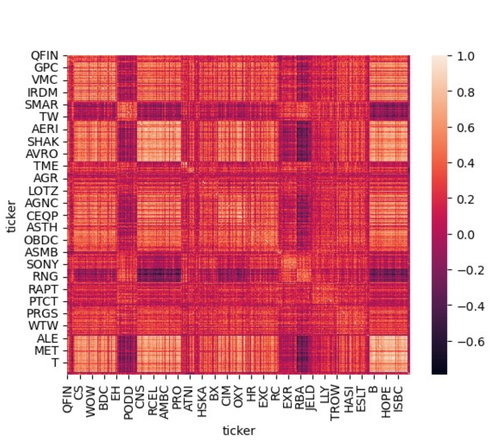


Step 1: Estimate clusters of stocks based on the correlation matrix of residual returns using the SPONGE algorithm, which is an effective algorithm that can find meaningful clusters of similar companies form our stock data by taking both negative and positive relationships between companies into consideration. 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.

Step 2: Compute a rolling correlation matrix of residual returns over some time window. This is our graph where stocks are nodes and edges are pairwise correlations. Then, we apply machine learning to forecast one-day forward risk-adjusted returns and formulate portfolios and returns based on the time-series momentum equation. We store the data in SQL and apply pandas and scikit-learn in this process.

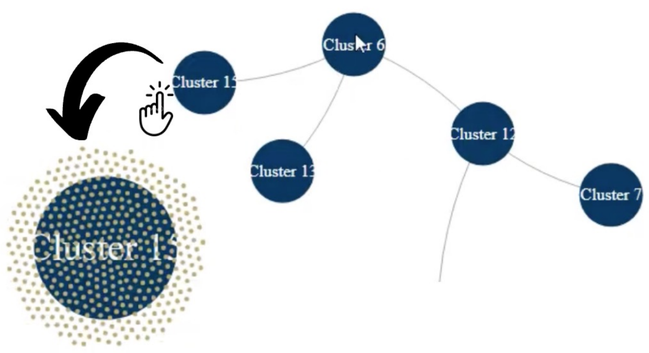
Step 3: Present the final project on the web using Flask and Streamlit. The result visualization and user interaction are implemented in D3.js.

Why do you think they can effectively solve your problem?



SPONGE offers advantages over traditional statistical arbitrage strategies by formulating the clustering problem as a generalized eigenvalue problem and considering both positive and negative interactions between nodes. It can effectively identify clusters, enhance the applicability to diverse networks, promote robustness to noise, and provide interpretable clustering results (as shown on the left).

We then transformed this matrix results into a more understandable graph (as shown on the right). If you click on the cluster, it will show the members within the cluster as smaller yellow dots. If you hover over the yellow dot, a tooltip will show up with the ticker, residual, and raw returns from the end date.



What is new in your approaches? In addition to replicating the SPONGE algorithm, we visualize the stock residual returns correlation matrix as a network using d3.js. Additionally, we build a time-series forecasting model to choose the best cluster for long-term investors to invest in over the next period.

DATA

How did we get it? The stock data is provided by an industry practitioner and can also be collected publicly.

What are the data characteristics?

- Size: 1.73GB
- Record Number: (37104728, 10)
- Permanent data and updated till March 2024

EXPERIMENTS AND RESULTS

How did you evaluate your approaches?

Hypothesis: Over/underweight mean cluster forecasting strategy can outperform the mean return of our investment universe while taking the same amount of risk.

- Design matrix X are time-series momentum features for each cluster on each day, and targets are their associated one-day forward risk-adjusted return. This is a regression problem.
- 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.
- Model ling: Use linear models as a baseline and compare with more complex machine models (e.g. ElasticNet vs Random Forest)
- Feature selection: We use SelectFromModel in sklearn for feature selection. Specifically, we add the object to 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.
- We will have multiple evaluation criteria. For tuning in the walk-forward step, we will use mean-squared error (MSE). We will evaluate investment performance metrics like total return and Sharpe ratio. We expect to see a linear relationship between the accuracy of the forecasting models and investment performance.

What are the results?

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

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

How do your methods compare to other methods? The findings demonstrate that the non-linear RandomForest model surpasses ElasticNet by almost 1 percent annually, indicating the presence of beneficial non-linear interactions within the feature set. RandomForest with feature selection ranks third, suggesting that excluding features does not yield advantages. These top three models consistently outperform the market by 1-2 percent, underscoring the benefits of dynamic modeling in identifying clusters that outperform the broader market. Annual volatility is noted to be similar across all strategies, facilitating reasonable return comparisons, and the Sharpe ratio validates these findings. RandomForest shows lower average Mean Squared Error (MSE) compared to ElasticNet . At the same time, RandomForest with feature selection records the lowest MSE in testing, potentially due to variability in returns impacting cumulative performance differently across models. The analysis employs cross-validation across four time-series split folds to substantiate these conclusions.

CONCLUSION

Our research confirms that machine-learning cluster strategies, focusing on five daily clusters using an equal-weighted approach, consistently outperform the market. Quartile sorts based on predictive rankings reveal strong trends, with top - quartile strategies significantly outperforming the bottom quartiles. Our findings suggest cyclical and idiosyncratic factors within market sectors, providing value to investors who strategically allocate towards these sectors using a trend-following forecasting system. Although implementation requires technical expertise and overlooks turnover and transaction costs, there is potential to package this strategy into an ETF for retail investors. Future work could optimize our approach further with deep learning techniques and custom loss functions to enhance returns and Sharpe ratios, leveraging methodologies like SPONGE for statistical arbitrage to forecast residual returns.