Constructing a Tracking Portfolio Using PCA and Graph Techniques

2022 Fall, Final Project Report

Supervisor: Chris Romano
Team Members: Yitian Duan, Shengbo Lang, Yan Sang, Daoming Zhang

1 Introduction

Tracking portfolio construction has always been a fundamental topic for exchange-traded fund (ETF) issuers. The objective is to mimic a portfolio's performance using only a subset of securities, with a desirable small tracking error, defined as the standard deviation of returns' difference. This is important for ETF issuers in order to provide market participants such as market makers with necessary information to hedge risks correspondingly, meanwhile avoid disclosing too many details of their ETFs' actual holdings.

Graph techniques are common in analyzing the structure of underlying objectives. There are well-established graph algorithms like planar maximally filter graph (PMFG) and minimal spanning tree (MST) in designing networks based on edge information. For instance, the edge information can be 'distance' or cost associated with certain edge, contributing to a final goal of minimizing total cost or finding the shortest path.

To build a tracking portfolio, we exploit securities' structure using graph techniques. More specifically, tracking portfolio concerns selection of representative securities to which graph algorithms can be applied, where securities are represented by nodes of a graph and edges can indicate relevance information between nodes or securities.

We shall begin with introducing relevant works, followed by graph theory including edge filtering algorithms. Then we steer to experiments on the stock market index Standard and Poor's 500 (S&P 500), explain how we bridge principal component analysis (PCA) to constructing an alternative relevance measure between securities, test on parameters and present results. After that, we conclude by possible improvements and future directions.

2 Related Work

Graph techniques applied in financial market network and portfolio construction is not merely a recent trend. Graph-derived central and peripheral stocks, determined by centrality rank, play different roles in a portfolio perspective. In a nutshell, portfolio made up of peripheral stocks are perceived to diversify risk, and central stocks are perceived to be influential and potent candidates in tracking major indexes [1][2][4].

Major graph filtering methods used in our project are based on planar maximally filter graph (PMFG) and minimal spanning tree (MST), which have been proven to be effective in constructing stock correlation network [2]. Besides, we propose a new winner-take-all (WTA) after spread sampling (SS) filtering method. This is inspired by results that spread sampling efficiently obtains small diversified samples from a larger graph; and WTA, also understood as threshold method, can select highly influential or dominant stocks from a large universe of pools [3][4].

This project extends previous projects to another dimension. The goal has changed from excess return to constructing a tracking portfolio with stable and small tracking error. To capture information in a richer way, we initiate the principal component analysis (PCA) approach in constructing a relevance measure between securities, working as an alternative to previously and widely used correlation matrix. In graph filtering step, closely related to selecting securities for portfolio construction, we bring in an SS-WTA algorithm (spread sampling then winner-take-all threshold method) in addition to PMFG and MST. As for weights assignment, we incorporate price-adjusted and return-adjusted weights in response to tracking purpose.

3 Methodology: Graph theory

We introduce graph filtering algorithms and centrality measures in this section.

3.1 Edge Filtering Algorithms

To clarify, the starting point is a fully-connected or 'unfiltered' graph where each edge is associated with a relevance score from PCA. The relevance score indicates how significant an edge or connection is. Graph filtering algorithms are applied to this unfiltered graph, to only keep edges of more importance.

3.1.1 Planar Maximally Filtered Graph

A graph is planar if and only if it can be drawn on the plane in a way that its edges intersect only at their endpoints. In other words, it can be drawn in such a way that no edges cross each other.

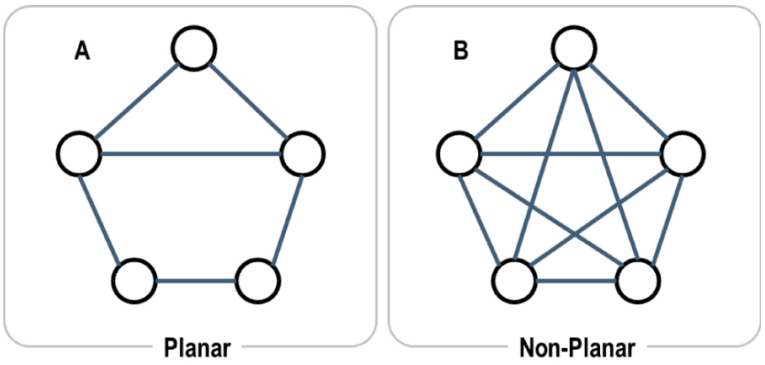


Figure 1: Planar and Non-Planar Graph

If $G(V, E)$ is a Connected Planar Graph with $|V| \geq 3$, then we have

$$|E| \leq 3|V| - 6$$

Since each face has at least three edges and then at least three degrees, we have

$$\sum_{f \in F(G)} deg(f) = 2|E|$$

then

$$3|F| \leq 2|E|$$

Plug the inequation into the Euler's formula, we have

$$3|F| = 3|E| - 3|V| + 6 \leq 2|E|$$

then

$$|E| \leq 3|V| - 6$$

The PMFG algorithm filters a simple weighted graph into a planar graph G with $|E(G)| = 3|V(G)| - 6$ if the number of the original graph has more than 3 vertices. The construction algorithm is given as follows.

Algorithm 1 Planar Maximally Filtered Graph

Input:
V: vertex set
$s_{i,j}$: weight(or similarity) between the i^{th} vertex and the j^{th} vertex where $i \neq j$ and $i, j = 1, 2, ..., |V|$
Output: $G(V, E)$: planar graph

$G(V, E) \leftarrow$ an empty graph
$S \leftarrow$ sort all $s_{i,j}$ in descending order
for $s_{i,j} \in S$ **do**
  $E \leftarrow E \bigcup s_{i,j}$
  if $G(V, E)$ is planar **then**
    $E \leftarrow E - s_{i,j}$
  end if
  if $|E| == 3|V| - 6$ **then break**
  end if
end for

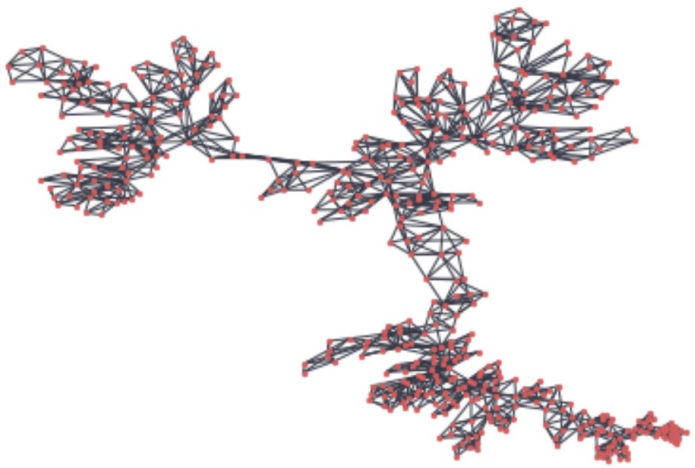


Figure 2: PMFG result, 3-month PCA on 10/01/2022

3.1.2 Minimal Spanning Tree

A spanning tree is a connected subgraph that is a tree which includes all the vertices of the original graph. The minimum spanning tree is a spanning tree with the minimum possible sum of edge weights and $|V| - 1$ edges.

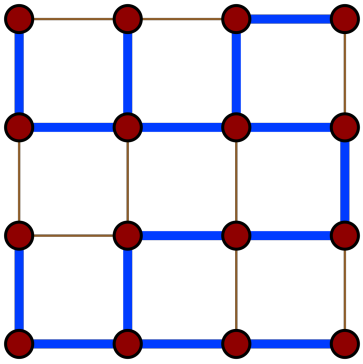


Figure 3: Spanning Tree

We use Kruskal's algorithm ($O(|E|log|V|)$ to construct MST. Because we want to seperate the central stocks and peripheral stocks, the MST should has the highest sum of edge weights. Thus, all the edges will be sorted in a descending order and then we iterate each of them starting from an empty graph. If at least one vertex of this edge is not included in the current graph, the edge will be added into the graph. The iteration would stop until the total number of edges equals to $(|V| - 1)$

Algorithm 2 Minimum Spanning Tree

Input:
V: vertex set
$s_{i,j}$: weight(or similarity) between the i^{th} vertex and the j^{th} vertex where $i \neq j$ and $i, j = 1, 2, ..., |V|$
Output: $G(V, E)$: MST graph

$G(V, E) \leftarrow$ an empty graph
$S \leftarrow$ sort all $s_{i,j}$ in descending order
for $s_{i,j} \in S$ **do**
  if $E \bigcup s_{i,j}$ does not have a cycle in E **then**
    $E \leftarrow E \bigcup s_{i,j}$
  end if
  if $|E| == |V| - 1$ **then break**
  end if
end for

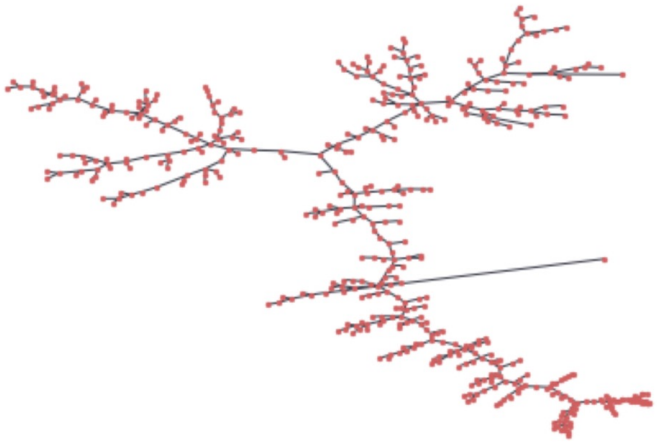


Figure 4: MST result, 3-month PCA on 10/01/2022

3.1.3 Spread Sampling, Winner-Take-All

We implement spread sampling (SS) first to find diversified nodes (stocks in our project) [3]. Adjustments are made for our study. The original SS method as shown in paper is to assign the same selecting probability p for every nodes; we assign re-weighted probability to nodes based on the relation matrix obtained from PCA, namely the selecting probability is proportional to its sum of 'relevance score' with every other nodes. This means the higher 'relevance' one node has with other nodes, more likely will it be sampled. The original SS method removed nodes that have over k connections to sampled set S, which is not applicable to our fully-connected graph. Thus, we set up the average of sum of 'relevance score' for each node as the threshold, nodes whose sum of scores to the sampled set nodes is larger than this threshold are removed, to achieve the idea of 'spread' or diversity.
After spread sampling we take winner-take-all (WTA) method to filter the graph. This is essentially a threshold method, only keeping edges associated with a higher 'relevance score'. In the reference they are using correlation, while we are using PCA-derived score [4]. So this is not that reasonable to use a value threshold. Alternatively, we can only keep a proportion of edges having the highest relevance scores. Below is a graph constructed on 10/01/2022, using 3-month PCA, selecting 400 out of 497 stocks by spread sampling, followed by keeping 2% edges associated with highest scores.

Algorithm 3 Spread Sampling (SS method)

Input:
p: infection rate
k: removal threshold
N: target sample size
$G(V, E)$: a connected undirected graph
Output: S: A set of sampled nodes
$S \leftarrow$ an empty node set
$C \leftarrow V$
while $|C|! = 0$ and $|S| < N$ **do**
for $u \in V$ **do**,
$b \sim B(1, p)$ a random variable from Bernoulli distribution
if $b == 1$ **then** $S \leftarrow S \cup u$
end if
end for
$V \leftarrow V - S$
$B_k \leftarrow$ an empty node set
for $v \in C$ **do**
if $|N(v) \cap S| \geq k$ **then** $B_k \leftarrow B_k \cup v$
end if
end for
$R \leftarrow R \bigcup B_k$
$C \leftarrow C - R$;
end while

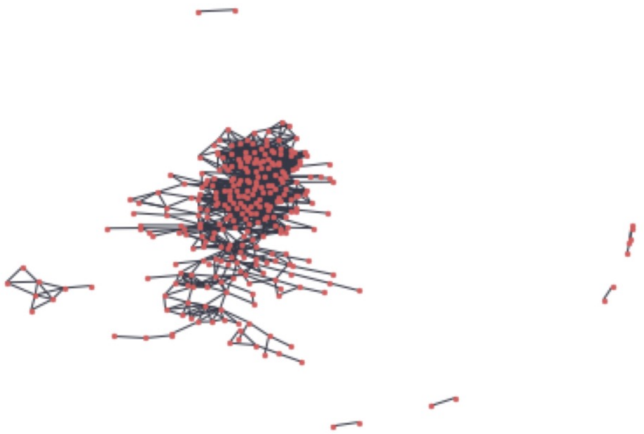


Figure 5: WTA result, top 2% egdes, 3-month PCA on 10/01/2022

3.2 Centrality Measures

Centrality measure aids to rank securities' centrality from the filtered graph. We then select candidates for tracking portfolio based on such centrality rank.

3.2.1 Closeness Centrality Measure

Given a connected graph $G(V, E)$, the closedness centrality $C(v)$ of a vertex $v \in V$ is defined as

$$C(v) = \frac{1}{\sum_w d(v, w)},$$

where $d(v, w)$ is the shortest-path distance between vertices v and w.

For a more 'central' vertex v, its distance to other vertices are generally smaller than a more 'peripheral' vertex, resulting in a higher closedness score $C(v)$.

Note that the cardinal value of a vertex's closeness depends also on the number of connected vertices to itself. That is to say, when comparing across graphs of different sizes, the right metric is normalized closeness measure. This potential issue can be neglected for our study, for we apply graph filtering to a fully connected graph of same number of vertices (number of securities of the original portfolio to track). To supplement, we are using the python package *NetworkX*, where normalized closedness is applied by default.

This closedness centrality measure is used for graphs filtered by PMFG or MST because of their topology structure.

3.2.2 Degree Centrality Measure

In graph theory, degree of a node or vertex refers to the number of neighbors, or the number of edges that are incident to the node or vertex. Given a graph $G = (V, E)$ and any vertex $v \in V$, or for simplicity assume G to be undirected ($(w, v) \in E$ is equivalent to $(v, w) \in E$), degree $D(v)$ is defined as

$$D(v) = \sum_{w \in V} 1_{(w,v) \in E},$$

where 1_A is the indicator function. Vertices of higher degrees are considered to be more 'central' due to its more connections with other vertices.

This degree centrality measure is used for graphs filtered by SS-WTA because of the threshold filtering nature. This makes less sense for PMFG- or MST-filtered graphs, since edges are more aggressively removed for more 'central' vertices to preserve planar or no-cycle property. Also both PMFG and MST give a fixed number of edges (in other words, fixed sum of degrees), limiting the power of degree centrality measure.

4 Empirical Tests

4.1 Data

We use S&P 500 constituents' daily data from the start of 2015 to the end of 2022, source library of the historical data is yfinance (the financial data available on Yahoo Finance), and the components of the S&P 500 constituents were generated by the current components of S&P500 index together with the historical changes during the past 7 years. To test the performance, we set S&P 500 components' price data from 2019-10 to 2020-10 as training set and data from 2020-10 to 2022-10 as test set. We use iShares S&P 100 ETF (OEF) as the reference model. It's built by 100 mega-cap US company stocks. The comparison between OEF and S&P 500

index (SPY) is shown in Figure 6.

The first kind of features in the PCA analysis are stocks' historical market data, including open price, highest price, close price, low price, and transaction volume. The close price we used is the adjusted close price with dividends deducted.

The second kind of the features is technical indicators generated by stocks' historical data. Those are heuristic or pattern-based signals produced by the price, volume, and open interest of security. Various technical indicators were produced to represent one conception (e.g. Multiple stochastic oscillators), and only one among the whole category remained here.

The third kind of the features is associated with the stock fundamental data. Since these data were released at least quarterly, we focused on those that can be applied together with stock price data. For example, the Price-to-earnings ratio and Price-to-sales ratio: although the earnings and sales were released quarterly, we can associate them with the stock price to generate a daily changing feature.

Over 100 features were generated at original trials, but due to practical issues like categorical features, frequency of data, and possible delays from the market, we select numerical features mainly. Some features are generated with the help of Python open resource FinTA (Financial Technical Analysis).

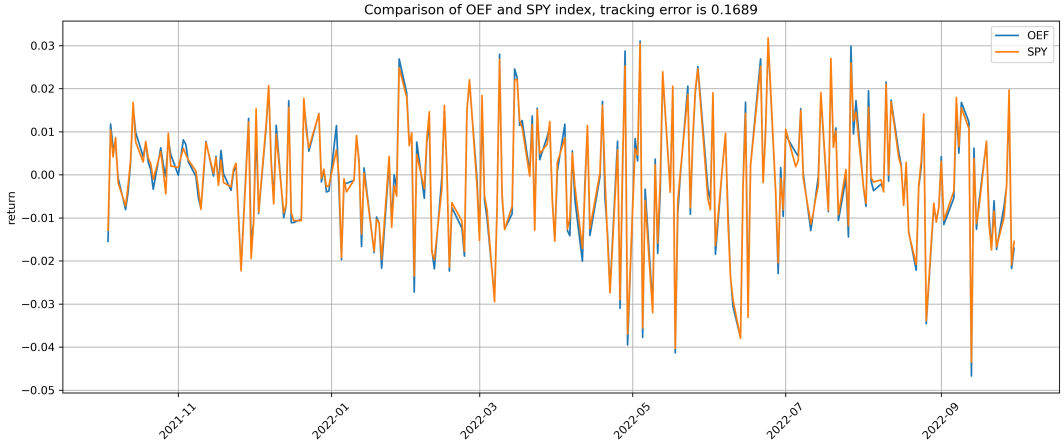


Figure 6: Daily return comparison between OEF and SPY over the period 2020-10 - 2022-10

4.2 PCA Approach for a 'Relevance Matrix'

The graph-generating process starts with a relevance matrix indicating the relations between each component. The previously used correlation matrix was replaced by a relevance matrix derived from PCA. We generate features and then applied daily PCA to obtain coefficients with regard to leading q principle components for each stock at time t.

PCA in stock features:

- At time t, we have N stocks and M features for each: x_1 to x_N (vectors in $\mathcal{R}^M$)
- After PCA was applied, we keep the q leading components of the results based on the explained variance: z_1 to z_N (vectors in $\mathcal{R}^q$)

- Throughout a period, (z_1, t) to (z_N, t) (vector sequences) was generated to represent the results of PCA analysis on the components.

After the vector sequences for each stock were produced, we will focus on how to generate a matrix to measure the relevance between the stocks for the purpose of building graphs. In order to indicate the 'distance' of any two stocks from the vectors, we compute the norm distance between them. A larger difference means they are explained more differently by principal components, meaning they are 'less relevant'. Therefore, the reciprocal of such distance is to replace correlation - a larger number means less norm difference and more similarities to each other.

Relevance measure between stocks:

- Given any two stocks i and j, compute norm difference d_{ij} between (z_i, t) to (z_j, t)
- Larger d_{ij} implies larger discrepancies of coefficients with respect to principle components, 'more different'
- Throughout a period, (z_1, t) to (z_N, t) (vector sequences) were generated to represent the results of PCA analysis on the components.

Here is an example, we do 3-month PCA on 10/01/2022, where 497 stocks are included.

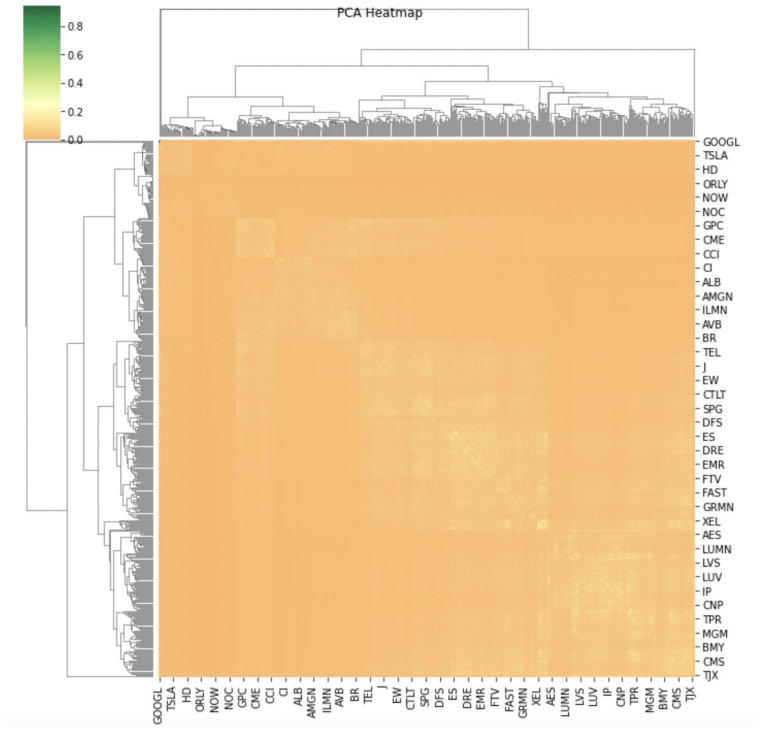


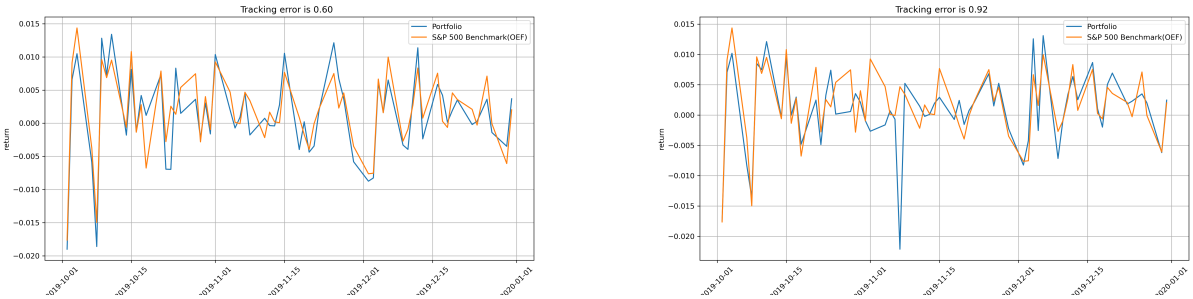
Figure 7: Relevance Matrix by PCA

4.3 Tests on Parameters

The aim of our study is to construct an index-tracking portfolio with fewer components and achieve smaller tracking errors. For the concern of time complexity, we need to select the optimal parameters and then conduct the whole backtest on the model. Finding the optimal parameters in the model is a challenge since there are multiple options at each step, and every

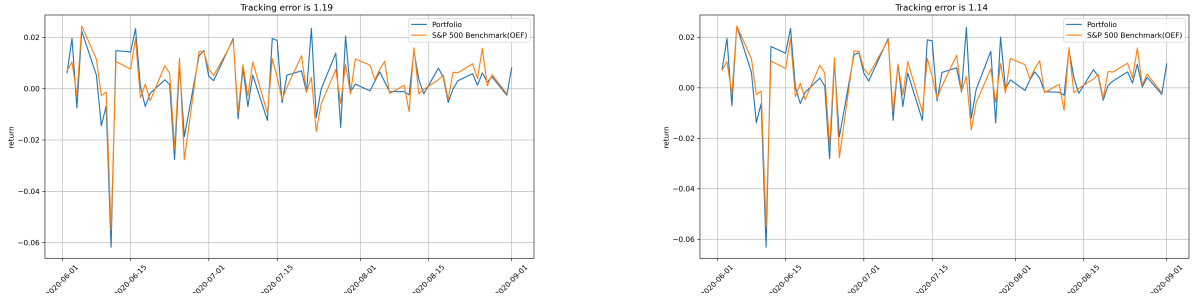
change can trigger instabilities in the model backtest results. Below is a list of all the parameter options we have in the model and comparisons are shown in Figure 7-12.

- Holding periods (1 months): the period we hold our portfolio, also the time interval between calculations in the backtest.
- PCA windows (3/6 months): we applied PCA to the past n months' data at each time of calculation.
- Graph filtering methods (PMFG/ MST/ WTA-SS): reduce the edges and nodes of the original graph, see details in **3.1**.
- Number of Stocks(30/50/100): number of stocks in the portfolio
- Stock Selection methods (central/ peripheral/ uniform/ random): stock selection methods based on the centrality information.
- Weight generating methods (Equal/ Markowitz/ low vol/ price-adjusted/return adjusted): the method we assign weight to each stock after selection



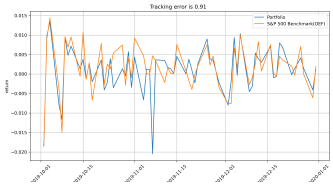
(a) 3 Months (b) 6 Months

Figure 8: Daily return comparisons between different PCA windows and S&P 100 portfolio over the period 2019-10-01 - 2019-12-31

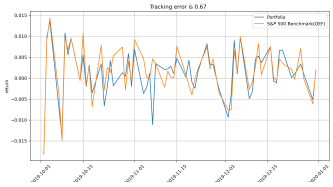


(a) MST (b) PMFG

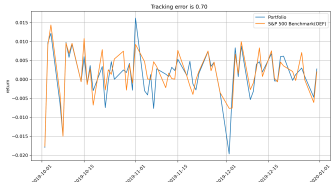
Figure 9: Daily return comparisons between MST/PMFG and S&P 100 portfolio over the period 2019-10-01 - 2019-12-31



(a) 30 stocks

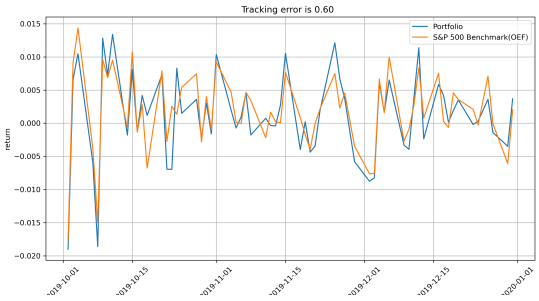


(b) 50 stocks

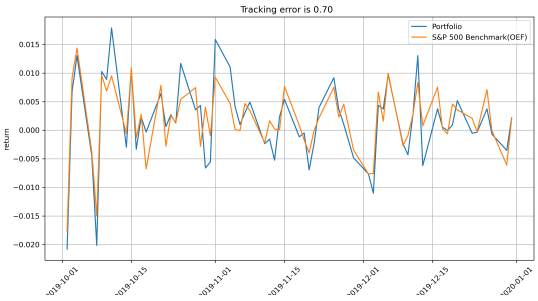


(c) 100 stocks

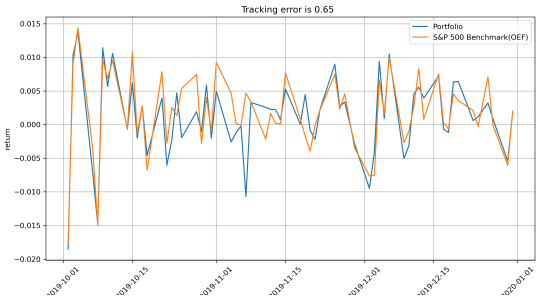
Figure 10: Daily return comparisons between portfolios with different number of stocks and S&P 100 portfolio over the period 2019-10-01 - 2019-12-31



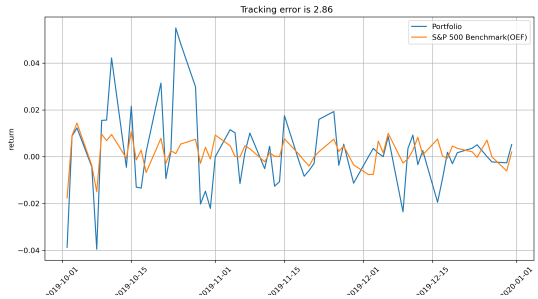
(a) Central



(b) Peripheral

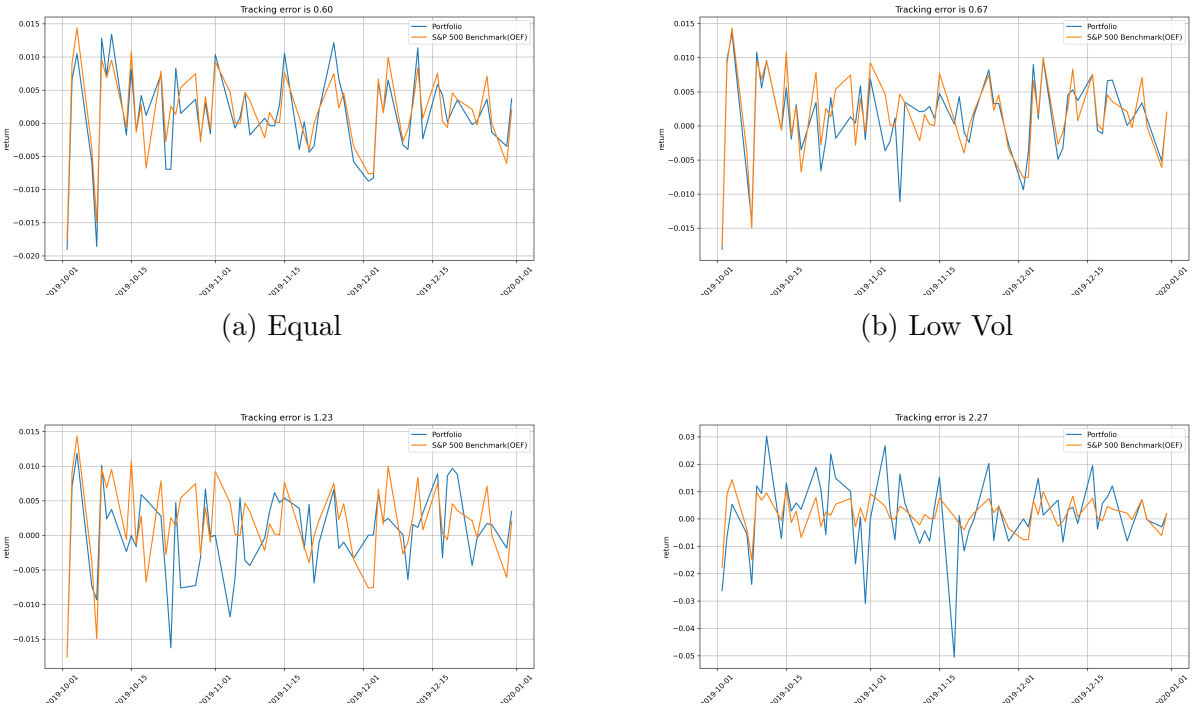


(c) Uniform



(d) Random

Figure 11: Daily return comparisons between 4 selection method and S&P 100 portfolio over the period 2019-10-01 - 2019-12-31



(a) Equal (b) Low Vol

(c) Markowitz (d) Return Ajusted

Figure 12: Daily return comparison between weight generation methods S&P 100 portfolio over the period 2019-10-01 - 2019-12-31

4.4 Results and Discussion

Choice of PCA windows and Graph filtering method do not have much impact on performance. We first test PCA windows of 3 and 6 months. They yield to similar short term and long term results. Considering that S&P components may change every quarter, a longer PCA window not only results in longer running time, but also reduces the pool of securities. Thus, we think 3 month window is a reasonable default choice. As for graph filtering, PMFG and MST have similar performances as well. We choose PMFG for our following experiments since PMFG-filtered graph has more edges where possibly less information is lost through the filtering process.

Number of stocks in the tracking portfolio is chosen to be 50. Experiments show that selecting more stocks results in smaller tracking errors. This is intuitive since the more constitutes of the tracking portfolio with other conditions unchanged, more information of the original portfolio is captured. Since our goal is tracking portfolio construction using a rather small subset of securities, we set the number of tracking securities to be 50, roughly 10% of the original portfolio or S&P 500.

Central stocks play significant roles in S&P 500 index. We investigated different ways of selecting stocks from centrality rank, which is obtained by calculating centrality scores of the filtered graph. Selecting central stocks outperforms all other strategies. Selecting random stocks gives much worse results compared to others, which demonstrate the effectiveness of PCA graph filtering. Also, peripheral selected stocks give worse results than central ones. This verifies the idea that tracking portfolio should be constructed by most influential stocks.

Equal weighted portfolio performs the best. Interestingly, assigning each stock an equal weight is better than all other 'fancy' methods. It is likely because we selected most central stocks and all of them are important. Low volatility strategy gives a little different weight distribution from the equal one, so their performance are similar in the long run. Markowitz is inferior for its goal is to maximize excess return rather than tracking a portfolio. Both price-adjusted and return-adjusted portfolio produces unstable results. One key reason is S&P 500 is weighted by market capitalization and assigning weights based on prices or returns does not comply with market rules, for large capitalization and high return or price are not always positive correlated.

Finally, we select 50 most central stocks with equal weight and 3 months PCA window, using PMFG method

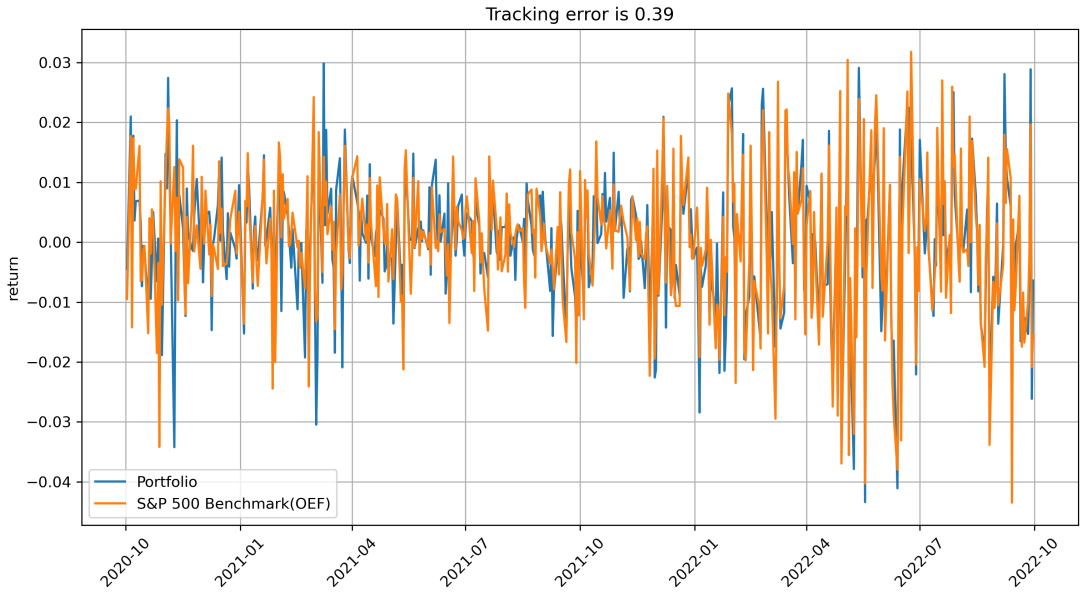


Figure 13: Daily returns of portfoilio and iShares S&P 100 ETF from 2020-10 to 2022-10

4.5 Conclusion

We have shown that relevance matrices obtained from stock feature PCA analysis can be used to construct a filtered list of stocks, which is appropriate in constructing an index-tracking portfolio. We finally selected 50 most central stocks with equal weight and 3 months PCA window, using PMFG method performed efficiently in the backtest with a tracking error under 0.5%.

Theoretically, an index fund should have a tracking error of zero relative to its benchmark, and enhanced index funds typically have tracking errors in the 1%-2% range. The backtesting performance of the model showed an optimistic result. It highlights the significance of PCA graph filtering techniques in constructing the tracking portfolio. We also find that stable stocks which stay in S&P 500 components for a long period are given more weight in the index.

In the present work, we explored several parameter combinations. However, it can be observed from previous results, we did not achieve huge improvements by choosing different parameters. This is expected because we mainly use price data. Currently we have no access to many useful features. Market capitalization plays a key role in S&P 500 index weight assignment. Other

financial data such as financial sector, P/E ratio and Debt-to-Equity ratio can play important roles in measuring centrality among component stocks.

5 Future Directions

PCA Features Currently we rely heavily on numerical features to conduct daily PCA, but it is well-known that categorical features and more tricky numerical features like financial sector and earnings are not negligible, since financial sector adds additional information to underlying securities' inherent structure. But how to encode categorical features into our PCA-graph framework is a bottleneck. Another bottleneck is how to process less frequent features and realistically deal with time lag (delay of announcements) to avoid look-ahead bias. This requires more careful treatments.

Graph Generation Works till now are mainly about filtering by PMFG and MST. Even though we propose an SS-WTA approach, it is still at early phase which gives unstable results and brings in many additional parameters so that not thoroughly tested yet. Future efforts can be put into other graph generation methods.

Parameter Calibration As discussed in experiment section, there are too many combinations of choices or parameters in our project so that we haven't tested them thoroughly. We might seek to improve code efficiency in the future.

Interpretability and Robustness A drawback of graph techniques applied here is that we blindly rely on numerical information without considering interpretability. As mentioned before, financial sector can effect results and increase interpretability, which should be prioritized for future study in our opinion. Robustness is another issue, not only in tracking error but also in components of the tracking portfolio. A practical remedy for robustness in security representatives could be on the step of stock selection from centrality rank. From experiments, selecting top central ones or uniformly yields to similarly good small tracking error. Future works can give stocks that are both top-ranked in centrality and exist in last portfolio priority in updating the tracking portfolio.

Risks and Data Due to limited access to data, we have not incorporated market capitalization data into our analysis. Since S&P 500 is weighted based on capitalization, and market capitalization includes size information in nature, our absence of that can lead to sub-optimal results. Depending on structure of the original portfolio to be tracked (in our study it is S&P 500), ad hoc further analysis is required. Another practical issue is securities' status changes, where active components are not constant over time from various reasons. Sometimes a ticker is removed from acquisition by another company listed, or from bad performance. This is similar to manage an ETF with components changes. How to process such information is a challenge. Also, we cannot download historical data for tickers that no longer exist from yahoo finance. This could also bias results.

References

[1] Peralta, G., & Zareei, A. (2016). A network approach to portfolio selection. *Journal of Empirical Finance, 38,* 157-180.

[2] Pozzi, F., Di Matteo, T., & Aste, T. (2013). Spread of risk across financial markets: better to invest in the peripheries. *Scientific reports, 3*(1), 1-7.

[3] Wang, Y., Bandyopadhyay, B., Chakrabarti, A., Sivakoff, D., & Parthasarathy, S. (2018). Spread sampling for graphs: Theory and applications.

[4] Chi, K.T., Liu J., & Lau, F.C. (2010). A network perspective of the stock market. *Journal of Empirical Finance, 17*(4), 659-667.

[5] D., Koutra, A. Parikh, A., Ramdas, J., Xiang (2011). Algorithms for Graph Similarity and Subgraph Matching.

[6] D., Song, M., Tumminello, W., Zhou, R.,N., Mantegna (2011). Evolution of worldwide stock markets, correlation structure, and correlation-based graphs. *Physical Review E, 84*(026108), 1-9.

[7] T., Yin, C., Liu, F., Ding, Z Feng, Bo Yuan, Ning Zhang (2022).Graph-based stock correlation and prediction for high-frequency trading systems. *Pattern Recognition, 122*(108209), 1-11.

[8] W., Xu, et al. (2017). HIST: A Graph-based Framework for Stock Trend Forecasting via Mining Concept-Oriented Shared Information.

[9] Bonanno, G., Lillo, F., Mantegna, R.N.(2001). High-frequency cross-correlation in a set of stocks. *Quantit. Finan.*(1), 96–104.

[10] Bonnanno, G., Caldarelli, G., Lillo, F., Mantegna, R.N.(2003). Topology of correlation-based minimal spanning trees in real and model markets. *Phys. Rev. E 68*, 046103.

[11] Campbell, R.A.J., Forbes, C.S., Koedijk, K.G., Kofman, P.,(2008). Increasing correlations or just fat tails? *Empirical Finance 15*, 287-309.

[12] F. Hu, Y. Zhu, S. Wu, W. Huang, L. Wang, T. Tan,(2020). graph representation learning with neighborhood aggregation and interaction, *Pattern Recognit*, 107745.

[13] Wei Bao, Jun Yue, and Yulei Rao.(2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one12, 7* .

[14] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen(2009). Pearson correlation coefficient *Noise reduction in speech processing*. Springer, 1–4.

[15] Qiyuan Gao(2016). Stock market forecasting using recurrent neural network. Ph.D. Dissertation. University of Missouri–Columbia