A Kernel Density Based Approach to Portfolio Optimization

Lawrence Liu

June 2023

Abstract

In this paper we propose a new approach to portfolio optimization. We show that the returns of the assets are not normally distributed, and thus we develop a new approach that does not rely on this assumption, unlike Modern Portfolio Theory. We show that this method provides a higher Sharpe Ratio and a higher Year over Year return compared to Modern Portfolio Theory.

1 Introduction

Portfolio optimization can be characterized as a constrained optimization problem, where we want to maximize the expected return of a portfolio while minimizing the risk. The risk is usually measured by the standard deviation of the portfolio. If we allow margin trading, meaning that we can borrow money to invest, and thus have a position larger than our initial capital, then the optimal portfolio becomes a scaling of the portfolio that maximizes the Sharpe ratio, this is what we call the efficient frontier. We define the sharpe ratio as the ratio of the expected return of the portfolio to the standard deviation of the portfolio.

$$\text{Sharpe Ratio} = \frac{\mu_p}{\sigma_p} \tag{1}$$

Where μ_p is the expected return of the portfolio, and σ_p is the standard deviation of the portfolio. If we take the μ_p as the average day to day return of the portfolio, and σ_p as the standard deviation of the day to day returns, then we can also define the annualized Sharpe ratio as

$$\text{Sharpe Ratio} = \frac{\mu_p}{\sigma_p}\sqrt{252} \tag{2}$$

Where 252 is the number of trading days in a year. From now on we use the annualized Sharpe ratio and all further references to Sharpe ratio will be the annualized Sharpe ratio. From now on we will only consider the case where we buy and hold, ie at the beginning of the period we allocate all of our capital into the portfolio and then leave it there for the entire period.

1.1 Modern Portfolio Theory

The most common approach to create the portfolio that maximizes the Sharpe ratio is Modern Portfolio Theory (MPT). MPT was first introduced by Harry Markowitz in 1952 [3]. Let us define the sample mean of the returns of the assets as μ, and the sample covariance matrix of the returns of the assets as Σ. Then μ_p and σ_p can be written as

$$\mu_p = w^T\mu \tag{3}$$

$$\sigma_p = \sqrt{w^T\Sigma w} \tag{4}$$

Where w is the weight vector of the portfolio. The optimization problem can then be written as

$$\begin{aligned} \underset{w}{\text{maximize}} \quad & \frac{w^T\mu}{\sqrt{w^T\Sigma w}} \\ \text{subject to} \quad & \sum_{i=1}^{n} w_i = 1 \\ & w_i \geq 0 \quad \forall i \in \{1, \ldots, n\} \end{aligned} \tag{5}$$

Where n is the number of assets in the portfolio. This is a fractional programming problem which we can solve in the following manner.[1] If we let $y = \alpha w$, where $\alpha = 1^T y$ is a scalar, then the problem becomes

$$\begin{aligned} \underset{y}{\text{minimize}} \quad & y^T\Sigma y \\ \text{subject to} \quad & \mu^T y = 1 \\ & y_i \geq 0 \quad \forall i \in \{1, \ldots, n\} \end{aligned} \tag{6}$$

This is a convex optimization problem, which we can solve with convex optimization methods. We then get that the optimal weights are given by $w^* = \frac{y^*}{1^T y^*}$, where y^* is the optimal solution to the convex optimization problem.

At the central core of MPT are two assumptions.

- The returns of the assets are normally distributed.
- The returns of the assets are stationary.

In this paper we will first show that the returns of the assets are not normally distributed. Then we will propose a new approach to portfolio optimization that does not rely on this assumption. We will show that this method provides a superior sharpe ratio compared to MPT.

2 Data

We used the data from the S&P 500 index in a 10 year period from 2010-01-01 to 2020-01-01. The data was downloaded from Yahoo Finance. Because certain companies were not traded during the entire period, we only used the companies that were traded during the entire period. As a result we had 428 companies in our dataset.

We used the adjusted close price of the stocks to calculate the returns. The data was split into a training set and a test set. the training set was the first 10 years of the data, and the test set was the last year of the data. We fit our model on the training set, and evaluated it on the test set.

Non Gaussanity of the Returns

To demonstrate that the day to day changes in the price of the assets are not normally distributed, let us first the plot the histogram of the day to day changes of the S&P 500 index over the timeframe of our dataset, and compare it to a normal distribution. This is plotted in figure 1. We can see that the histogram compared to the fitted normal distribution has a higher peak and fatter tails, but its "shoulder" sags more.

To more quantifiablely demonstrate that the returns are not normally distributed, let us perform a Kolmogorov-Smirnov test at a significance level of 0.001. The null hypothesis of the Kolmogorov-Smirnov test is that the distribution of the day to to day change in the price of the asset is normally distributed according to the fitted normal distribution. At this confidence level, we can reject the null hypothesis of normallity for all but 6 of the 428 assets.

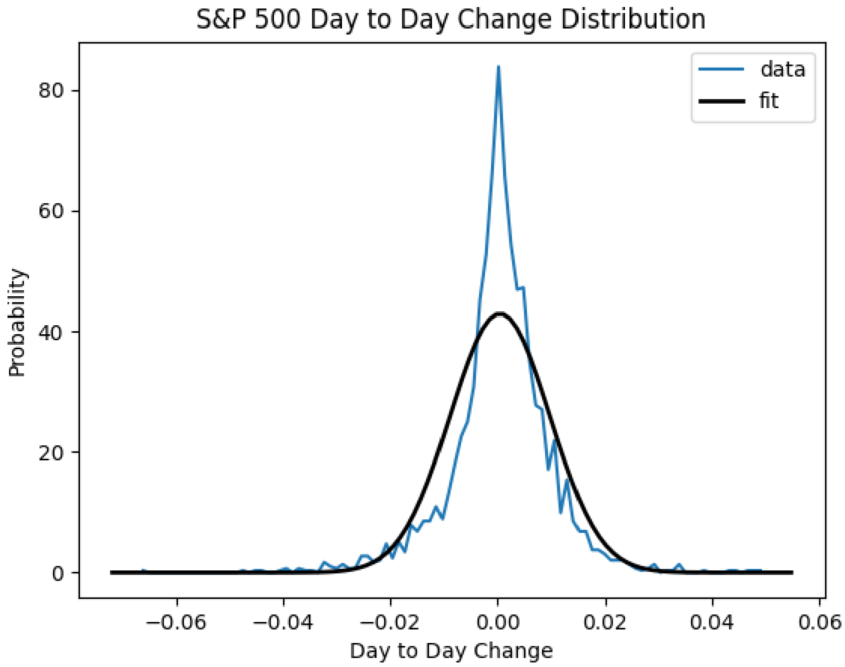

Figure 1: Histogram of the day to day changes of the S&P 500 index over the timeframe of our dataset, and a fitted normal distribution.

3 Method

Our method consists of two parts. The first part is to model the returns of a set of assets with a nonparametric Kernel Density Estimation with a multivariate Gaussian Kernel. The second part is to isolate the assets into smaller subsets that are highly correlated with itself through spectral clustering.

3.1 Kernel Density Estimation

Effectively with Kernel Density Estimation what we try to do is with a kernel function, we smooth a normalized histogram of the data. The kernel function is a function that is centered around a point, and is symmetric around that point. Mathematically, we can write the kernel density estimation for the the *pdf* of a random variable X as

$$\hat{f}_\theta(x) = \frac{1}{n}\sum_{i=1}^{n} K_\theta(x - x_i') \tag{7}$$

where K_θ is the kernel function, and θ are the parameters of the Kernel, and x_i' is the ith training point. In our case x_i' is a m dimensional vector representing the daily change for each of the m stocks in our dataset. We assume that that each kernel function is normalized $\int_{\mathbb{R}^d} K_\theta(x)dx = 1$

3.1.1 Optimizing the Kernel Parameters

We would want to find the optimal Kernel Parameters, we would want to minimize the log likelihood function, ie we would want to minimize:

$$\mathcal{L}(x_1, \ldots, x_k) = -\sum_{i=1}^{k} \log(\hat{f}_\theta(x_i)) \tag{8}$$

Where $x_1, \dots, x_k$ are the test points.

Now let us restrict our considerations to the case of a gaussian multivariate kernel. We have that

$$K_\Sigma(x) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2} x^T \Sigma^{-1} x\right)$$

Where Σ is the covariance matrix for the kernel. Because the covariance matrix is symmetric positive semidefinite we can express Σ as $\Sigma = R^T R$ through Cholesky decomposition. We have that the derivative of the weighted log likelihood function is given by:

$$\frac{\partial}{\partial R} \mathcal{L}(x_1, \dots, x_k) = -\sum_{i=1}^{k} \frac{1}{\hat{f}_\theta(x_i)} \frac{\partial \hat{f}_\theta(x_i)}{\partial R}$$

We have that

$$\begin{aligned} \frac{\partial |\Sigma|}{\partial R} &= \frac{\partial |R^T R|}{\partial R} \\ &= 2|\Sigma| R^{-T} \end{aligned}$$

Therefore we have that

$$\frac{\partial}{\partial R} \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} = -R^{-T} \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \tag{9}$$

We also have that:

$$\frac{\partial}{\partial R} e^{\frac{1}{2} x^T \Sigma^{-1} x} = \frac{1}{2} e^{\frac{1}{2} x^T \Sigma^{-1} x} \frac{\partial}{\partial R} x^T (R^{-1} R^{-T}) x \tag{10}$$

$\frac{\partial}{\partial R} x^T (R^{-1} R^{-T}) x$ is very difficult to calculate, so we must approximate it. First we note that for a function $f(\mathbf{x})$ that takes in a vector $\mathbf{x}$ we have that the first order taylor expansion is given by:

$$f(\mathbf{x} + \Delta \mathbf{x}) \approx f(\mathbf{x}) + \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \Delta \mathbf{x} \tag{11}$$

Where $\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$ is a $1 \times d$ vector, if $\mathbf{x}$ is a d dimensional vector. Therefore we argue that a generalization to a function of a matrix $\mathbf{X}$ is given by:

$$f(\mathbf{X} + \Delta \mathbf{X}) \approx f(\mathbf{X}) + \mathbf{1}^T \left(\frac{\partial f(\mathbf{X})}{\partial \mathbf{X}} \circ \Delta \mathbf{X} \right) \mathbf{1} \tag{12}$$

Where $\circ$ is the Hadamard product, and $\mathbf{1}$ is a $d \times 1$ vector of ones. We note that $\mathbf{1}^T \left(\frac{\partial f(\mathbf{X})}{\partial \mathbf{X}} \circ \Delta \mathbf{X} \right) \mathbf{1}$ equals to $\operatorname{tr}\left(\left(\frac{\partial f(\mathbf{X})}{\partial \mathbf{X}} \right)^T \mathbf{X} \right)$. We have for our specific case:

$$x^T ((R + \delta R)^{-1} (R + \delta R)^{-T}) x \approx x^T (R^{-1} R^{-T}) x + \operatorname{tr}\left(\left(\frac{\partial (x^T R^{-1} R^{-T}) x}{\partial R} \right)^T \delta R \right)$$

We note that for small pertubations, $(R + \delta R)^{-1} \approx R^{-1} - R^{-1} \delta R R^{-1}$, and therefore:

$$x^T (R^{-1} - R^{-1} \delta R R^{-1})(R^{-T} - R^{-T} \delta R^T R^{-T}) x \approx x^T (R^{-1} R^{-T}) x + \operatorname{tr}\left(\left(\frac{\partial (x^T R^{-1} R^{-T}) x}{\partial R} \right)^T \delta R \right)$$

Only keeping the zeroth order and first order terms, we have that:

$$\begin{aligned} x^T (R^{-1} R^{-T}) x - x^T \left(R^{-1} \delta R R^{-1} R^{-T} + R^{-1} R^{-T} \delta R^T R^{-T} \right) x \approx & x^T (R^{-1} R^{-T}) x \\ & + \operatorname{tr}\left(\left(\frac{\partial (x^T R^{-1} R^{-T}) x}{\partial R} \right)^T \delta R \right) \end{aligned}$$

$$-x^T\left(R^{-1}\delta R R^{-1}R^{-T} + R^{-1}R^{-T}\delta R^T R^{-T}\right)x \approx \operatorname{tr}\left(\left(\frac{\partial(x^T R^{-1}R^{-T})x}{\partial R}\right)^T \delta R\right) \tag{13}$$

Because the left side is a scalar, we can apply an trace operator to both sides, and noting that $R^{-1}R^{-T} = \Sigma^{-1}$, we have that:

$$\begin{aligned}
-\operatorname{tr}\left(x^T\left(R^{-1}\delta R\Sigma^{-1} + \Sigma^{-1}\delta R^T R^{-T}\right)x\right) &\approx \operatorname{tr}\left(\left(\frac{\partial(x^T R^{-1}R^{-T})x}{\partial R}\right)^T \delta R\right) \\
-\operatorname{tr}\left(x^T R^{-1}\delta R\Sigma^{-1}x\right) - \operatorname{tr}\left(x^T\Sigma^{-1}\delta R^T R^{-T}x\right) &\approx \\
-\operatorname{tr}\left(x^T R^{-1}\delta R\Sigma^{-1}x\right) - \operatorname{tr}\left(x^T R^{-1}\delta R\Sigma^{-T}x\right) &\approx \\
-2\operatorname{tr}\left(x^T R^{-1}\delta R\Sigma^{-1}x\right) &\approx \\
-2\operatorname{tr}\left(\Sigma^{-1}xx^T R^{-1}\delta R\right) &\approx \operatorname{tr}\left(\left(\frac{\partial(x^T R^{-1}R^{-T})x}{\partial R}\right)^T \delta R\right)
\end{aligned}$$

Therefore we can see that

$$\frac{\partial}{\partial R}x^T(R^{-1}R^{-T})x \approx -2\Sigma^{-1}xx^T R^{-1} \tag{14}$$

Therefore we have that:

$$\frac{\partial}{\partial R}e^{\frac{1}{2}x^T\Sigma^{-1}x} \approx -e^{\frac{1}{2}x^T\Sigma^{-1}x}\Sigma^{-1}xx^T R^{-1} \tag{15}$$

Therefore we have that

$$\frac{\partial}{\partial R}K_\Sigma(x) \approx -K_\Sigma(x)R^{-T} - K_\Sigma(x)\Sigma^{-1}xx^T R^{-1} \tag{16}$$

And thus we have:

$$\frac{\partial}{\partial R}\hat{f}(x) \approx -\sum_{i=1}^{n} K_\theta(x - x_i')\left(R^{-T} - \Sigma^{-1}(x - x_i')(x - x_i')^T R^{-1}\right) \tag{17}$$

Therefore we have that:

$$\frac{\partial}{\partial R}\mathcal{L}(x_1,\ldots,x_k) \approx \sum_{i=1}^{k}\frac{1}{\hat{f}(x_i)}\sum_{j=1}^{n} K_\theta(x_i - x_j')\left(R^{-T} - \Sigma^{-1}(x_i - x_j')(x_i - x_j')^T R^{-1}\right) \tag{18}$$

From this we can just optimize R using stochastic gradient descent, or other more advanced methods such as ADAM. For this report we just used stochastic gradient descent with $\eta = 0.01$ and $T = 10$ epochs with minibatching the data into 20 batches. We separated the train data into 80% train and 20% validation data. We then used the validation data as the test values we used to calculate the negative log likelihood.

3.1.2 Obtaining a More Accurate Sharpe Ratio

As we noted before what we want to optimize is the sharpe ratio: What we want to optimize is the sharpe ratio, which is given by:

$$\frac{\mathbb{E}[R_p]}{\sigma_p} \tag{19}$$

If we have that the weights for each of the m stocks is given by a vector w, then we can write the expected return as:

$$\mathbb{E}[R_p] = \mathbb{E}[w^T x] \tag{20}$$

Where x is the vector of daily returns for each of the m stocks. We have that this is given by:

$$\mathbb{E}[w^T x] = \int_{\mathbb{R}^m} w^T x\hat{f}_h(x)dx \tag{21}$$

$$= \frac{1}{n}\sum_{i=1}^{k}\int_{\mathbb{R}^m} w^T xK_\Sigma(x - x_i')dx \tag{22}$$

Now we can use the fact that the kernel function is symmetric to get:

$$\int_{\mathbb{R}^m} xK_\Sigma(x - x'_i)dx = \int_{\mathbb{R}^m} (x'_i - u)K_\Sigma(u)du$$

Where $u = x - x'_i$, because $K_\Sigma(u)$ is symmetric (even) about $u = 0$. we have that $\int_{\mathbb{R}^m} uK_\Sigma(u)du = 0$, thus we have that

$$\int_{\mathbb{R}^m} xK_\Sigma(x - x'_i)dx = \int_{\mathbb{R}^{10}} x'_i K_\Sigma(u)du \tag{23}$$

$$= x'_i \tag{24}$$

Therefore we have that:

$$\mathbb{E}[w^T x] = \frac{1}{n}\sum_{i=1}^{n} w^T x'_i \tag{25}$$

Now we need to find the variance of the portfolio, which is given by:

$$\sigma_p^2 = \mathbb{E}[(w^T x)^2] - \mathbb{E}[w^T x]^2 \tag{26}$$

From equation (25) we have that, $\mathbb{E}[w^T x]^2 = \left(\frac{1}{n}\sum_{i=1}^{n} w^T x'_i\right)^2$. We have that $\mathbb{E}[(w^T x)^2]$ is given by:

$$\mathbb{E}[(w^T x)^2] = \int_{\mathbb{R}^m} (w^T x)^2 \hat{f}_h(x)dx \tag{27}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\int_{\mathbb{R}^m} (w^T x)^2 K_\Sigma(x - x'_i)dx \tag{28}$$

We have that because $w^T x$ is a scalar, we have that $(w^T x)^T = x^T w = w^T x$, therefore we have that:

$$\mathbb{E}[(w^T x)^2] = \frac{1}{n}\sum_{i=1}^{n}\int_{\mathbb{R}^m} w^T x x^T w K_\Sigma(x - x'_i)dx \tag{29}$$

$$= \frac{1}{n}\sum_{i=1}^{n} w^T \left(\int_{\mathbb{R}^m} x x^T K_\Sigma(x - x'_i)dx\right) w \tag{30}$$

$$= w^T \left(\frac{1}{n}\sum_{i=1}^{n}\int_{\mathbb{R}^m} x x^T K_\Sigma(x - x'_i)dx\right) w \tag{31}$$

For a gaussian kerenel with covariance matrix Σ, we have that:

$$\int_{\mathbb{R}^{10}} x x^T K_\Sigma(x - x'_i)dx = x'_i x'^T_i + \Sigma \tag{32}$$

Therefore we have that equation (31) becomes

$$\mathbb{E}[(w^T x)^2] = w^T \left(\frac{1}{n}\sum_{i=1}^{n} x'_i x'^T_i + \Sigma\right) w \tag{33}$$

$$= w^T \left(\Sigma + \frac{1}{n}\sum_{i=1}^{n} x'_i x'^T_i\right) w \tag{34}$$

$$= w^T \Sigma w + w^T \left(\frac{1}{n}\sum_{i=1}^{n} x'_i x'^T_i\right) w \tag{35}$$

Therefore we have that equation (26) becomes:

$$\sigma_p^2 = w^T \Sigma w + w^T \left(\frac{1}{n}\sum_{i=1}^{n} x'_i x'^T_i\right) w - \left(\frac{1}{n}\sum_{i=1}^{n} w^T x'_i\right)^2 \tag{36}$$

We have: $\left(\frac{1}{n}\sum_{i=1}^{n} w^T x'_i\right) = w^T \left(\frac{1}{n}\sum_{i=1}^{n} x'_i\right) = \left(\frac{1}{n}\sum_{i=1}^{n} x'_i\right)^T w$, therefore we have that:

$$\sigma_p^2 = w^T \left(\Sigma + \frac{1}{n}\sum_{i=1}^{n} x'_i x'^T_i - \left(\frac{1}{n}\sum_{i=1}^{n} x'_i\right)\left(\frac{1}{n}\sum_{i=1}^{n} x'_i\right)^T\right) w \tag{37}$$

Thus we can see that with Kernel Density Estimation we have effectively created a pseduo-covariance matrix

$$\hat{\Sigma} = \Sigma + \frac{1}{n}\sum_{i=1}^{n} x'_i x'^T_i - \left(\frac{1}{n}\sum_{i=1}^{n} x'_i\right)\left(\frac{1}{n}\sum_{i=1}^{n} x'_i\right)^T \tag{38}$$

Therefore we can use this instead of the sample covariance matrix in the fractional programming problem to find the optimal portfolio weights that maximize the Sharpe ratio.

3.2 Spectral Clustering

Our experimentation found that Kernel Density Estimation would fail on high dimensional datasets such as the 400+ stocks we used in our experiments. Therefore we decided to adopt a strategy of dimensionality reduction. We first performed Kernel Density Estimation on a subset of stocks that were from the same "sector," and then generating an optimal portfolio of these stocks of the dataset. Then we applied Kernel Density Estimation to the dataset of optimal portfolios for each sector.

To identify the sectors we used spectral clustering on the correlation matrix, specifically we used the negative of the correlation matrix as the adjacency matrix, with the diagonal elements zeroed out. We chose to use this method because we wanted to identify clusters of stocks that were highly correlated with each other.

To determine the optimal number of clusters we used the eigengap heuristic as proposed by [5] and [4]. The eigengap is defined as the difference between the eigenvalue of the kth smallest eigenvalue and the $(k+1)$th smallest eigenvalue. Then the optimal number of clusters is the value of k that maximizes the eigengap. Because our "graph" created by the covariance matrix is fully connected, we had to limit the number of clusters to a minimum of 5 and a maximum of 50. We found that the optimal number of clusters was 17.

4 Results

4.1 Spectral Clustering

As we discussed previously, we used the eigengap, we have plotted the eigenvalues and the eigengap for the for the sample covariance matrix measured from the train set in Figure 2. We can see that the eigengap is maximized at 17.

We also have plotted out the sample correlation matrix, before clustering, and after clustering in Figure 3. We can see that the correlation matrix after clustering has a block diagonal structure, which is what we would expect from a clustering algorithm.

To investigate how the stocks are clusters, we report the stock ticker, and corresponding company name for a select number of clusters in Table 1 and Table 2. We can see that the clustering algorithm was able to identify clusters, with the stocks in cluster 7 being mostly financial companies, and the stocks in cluster 10 being mostly energy companies.

Ticker	Company
AMP	Ameriprise Financial, Inc.
BK	The Bank of New York Mellon Corporation
C	Citigroup Inc.
FITB	Fifth Third Bancorp
JPM	JPMorgan Chase & Co.
LNC	Lincoln National Corporation
MET	MetLife, Inc.
NTRS	Northern Trust Corporation
PNC	The PNC Financial Services Group, Inc.
PRU	Prudential Financial, Inc.
RJF	Raymond James Financial, Inc.
STT	State Street Corporation
HIG	The Hartford Financial Services Group, Inc.
TFC	Truist Financial Corporation
USB	U.S. Bancorp
WFC	Wells Fargo & Company

Table 1: The tickers in cluster 7

Ticker	Company
LNT	Alliant Energy Corporation
AEE	Ameren Corporation
AEP	American Electric Power Company, Inc.
ATO	Atmos Energy Corporation
CNP	CenterPoint Energy, Inc.
CMS	CMS Energy Corporation
D	Dominion Energy, Inc.
DTE	DTE Energy Company
EVRG	Evergy, Inc.
ES	Eversource Energy
NEE	NextEra Energy, Inc.
NI	NiSource Inc.
PNW	Pinnacle West Capital Corporation
PEG	Public Service Enterprise Group Incorporated
SRE	Sempra
XEL	Xcel Energy Inc.

Table 2: The tickers in cluster 10

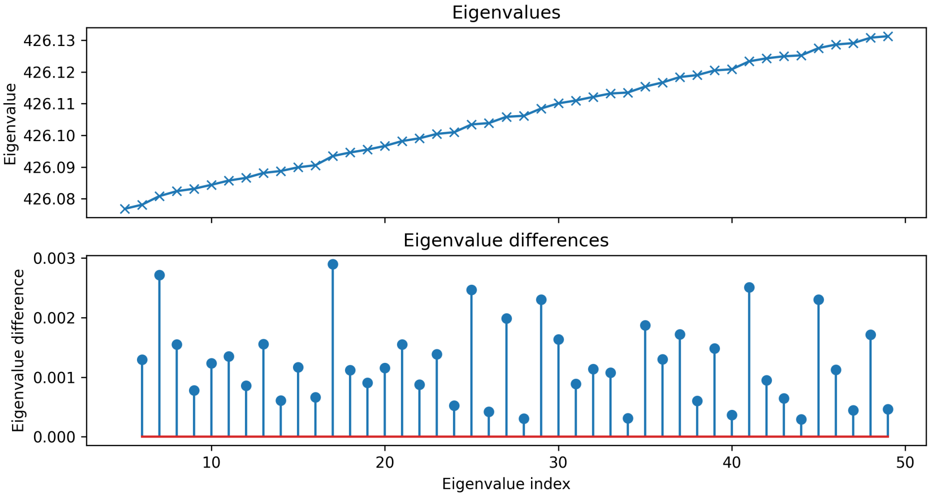

Figure 2: Eigenvalues and Eigengap

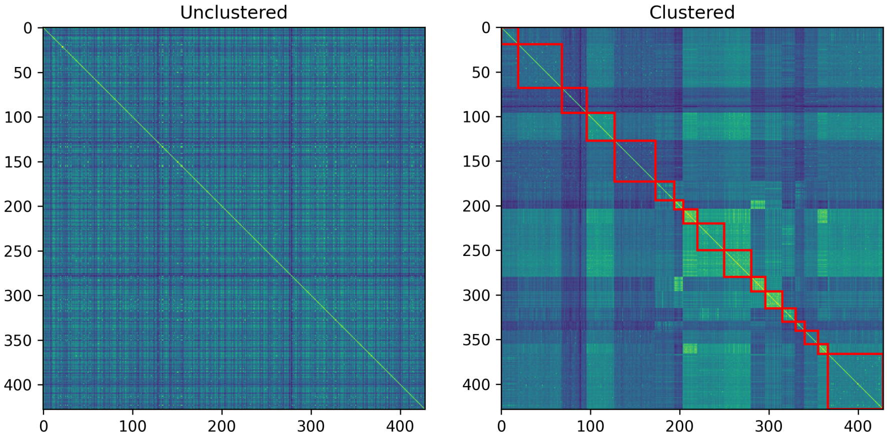

Figure 3: Correlation Matrix Before and After Clustering, with the Clustered blocks highlighted in red

4.2 Kernel Density Estimation

We investigated 3 methods of portfolio optimization, the first was the standard Markowitz portfolio optimization, the second was applying Kernel Density Estimation straight to the dataset, and the third was applying Kernel Density Estimation to each sector, and then applying Kernel Density Estimation to the dataset of optimal portfolios. In our table we denote this method as "Spectral Clustering + KDE."

The sharpe ratio each portfolio was able to achive is shown in Table 3. We can see that the our method of Spectral Clustering + KDE was able to achieve the highest sharpe ratio. We also reported out the average year over year (YoY) returns for each method in Table 4.

Method	Sharpe Ratio (train)	Sharpe Ratio (test)
S&P 500	0.804	0.709
Markowitz	2.266	1.199
KDE	1.201	0.920
Spectral Clustering + KDE	1.969	1.424

Table 3: Sharpe Ratios for each method

Method	YOY (train)	YOY (test)
S&P 500	11.352 %	9.947%
Markowitz	37.699%	15.969%
KDE	19.362%	12.743%
Spectral Clustering + KDE	25.246%	17.398%

Table 4: Sharpe Ratios for each method

As we can see the Spectral Clustering + KDE method was able to achieve the highest sharpe ratio, and the highest YoY returns on the test set. And was able to outperform the S&P 500 by 7.451% in terms of YoY returns in the test set.

5 Discussion

While we were able to achieve a higher sharpe ratio and YoY returns on the test set, we believe that there is additional performance that can be tuned out of our model. When we trained it, we noticed "ping-ponging" behavior, where the negative log likelihood would increase and decrease. Likewise we noticed that optimization algorithm was highly sensitive to the learning rate and initial conditions. We believe that this is due to the fact that we used a fairly simple optimization algorithm, and further work can be done with more sophisticated optimization algorithms such as stochastic gradient descent with momentum and a cosine annealing learning rate scheduler, or using Adam [2].

We also noticed that the algorithm would often produce a Σ that was not positive semidefinite or that was poorly numerically conditioned, leading to the production of NaN values by the kernel density estimation algorithm, which would cause the optimization algorithm to fail. Thus one direction of future work would be to make the algorithm more numerically stable. Another further direction would be to investigate other kernels that can have "fatter" tails than the Gaussian kernel, such as a multivariate t-distribution kernels.

Other directions also include developing a model that can handle non stationary distributions, the other drawback of MPT that we discussed in the introduction.

References

[1] Theo Diamandis. It's convex if you tilt your head a bit.

[2] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[3] Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[4] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001.

[5] Ulrike von Luxburg. A tutorial on spectral clustering, 2007.