Plans for the Project

Do a rolling window PCA to obtain the first 3 eigenvalues and eigenvectors for each window. (3 is chosen arbitrarily). [Window size of 3 months?]  
*Something to think about: In my code, I create data frames such that every row is complete. It may be beneficial to consider rows where only a few assets are missing. These rows would appear at the top of the dataframe since assets do not drop out early in our data set. The only difference is the time at which they first appear. Maybe create a threshold for the maximum number of empty values.*

Concatenate the eigenvectors and eigenvalues so we obtain one large vector which encompasses all of this information. We have one such vector for every window.

Do a clustering to group these vectors into clusters. We believe that the eigenvalues and eigenvectors hold information describing the state of the market. The number of clusters that appear will be the number of different market states.  
*Things to think about: Eigenvalues and eigenvectors may be correlated, so it might not be necessary to consider all of this information. Moreover, it is possible that including this extra information could reduce the quality of the clustering. For example, by considering the eigenvectors we may end up increasing the distance between two points which should actually appear closer together in the clustering.*

(Try: Create definitions for each market state)

Once we have these clusters, we can try different portfolio allocations on the clusters, and see if some portfolio allocation performs the best on every member of a cluster. We can create a chart as follows:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Allocation 1 (1/N) | Allocation 2 (MinVar) | Allocation 3 (EqRisk) |
| Cluster 1 | X% | Y% | Z% |
| … | … | … | … |
| Cluster N | R% | S% | T% |

The percentages represent the percentage of members in the given cluster which performed best under the given portfolio allocation. (Each row sums to 100%). What do we mean by “performing best”? There are various ways to measure how good a portfolio is. One example is to check the Sharpe Ratio.

*Something to think about: How do we forecast the current/future market states? Maybe there is some expected time duration for each market state. Think about this.*

If we find a 100% in any cell, this is a strong indicator that the chosen allocation is the best allocation for the market state corresponding to the given cluster. However, it seems unlikely that a 100% will appear. Moreover, even if one does appear, this only indicates that this allocation strategy performs the best out of the selected strategies which are tested. It does not indicate in any way that the strategy is optimal.

How do we optimize a portfolio allocation?

First of all, what is a portfolio allocation? It is simply a vector in which each component represents the percentage of the portfolio that is allocated into the asset which is represented by that specific component.

How can we use this to optimize our portfolio allocation? It seems like a neural network is one way that we could go about doing this. Our input layer is a vector which corresponds to one given time window (the vector is the concatenated vector explained earlier). By going through the neural network, the last layer will have m nodes, where m is the number of assets considered. We would like to find an optimal portfolio for each cluster. Thus, the number of data points is equal to the number of points in a given cluster. For each cluster we must perform a separate portfolio optimization. The “cost” function can simply be taken to be the negative of the Sharpe Ratio, for example.

Once we are satisfied with our outputted portfolio allocation, we can back test it on the data points for the given cluster. Hopefully we will find that 100% of the data points in the cluster perform best under this new portfolio allocation.

Later on, we could try to find some financial explanation to help make sense of the outputted portfolio allocations. Hopefully there is some intuition behind these portfolios.