The Cooper Union Department of Electrical Engineering
Prof. Fred L. Fontaine
ECE478 Financial Signal Processing
Problem Set I: Portfolio Analysis
September 20, 2018

Data

In terms of time scale, we are interested in daily values for return, interest rate, etc. Data may be on different time scales (e.g., daily, monthly, etc.). You may need to do some conversion to put all values on the equivalent annualized basis. For example, you can't take say daily S&P 500 returns and compare them to annual LIBOR. Normalize all time scales for your data. Note that daily data is not uniformly sampled in time (e.g., weekends are omitted), but for our purposes you can ignore that issue.
By default, we will use the Farma and French benchmark dataset, specifically FF48. Data from a large number of stocks is separated into 48 sectors (e.g., agriculture, soda), and in each sector a composite portfolio return is computed. In particular, we will use the *daily, equally weighted returns* for Jan 1, 2000 through Dec 31, 2016. To make the statements of the assignments more clear, I am going to call each of these a security (i.e., the portfolio in each sector is bundled as a single security which we can imagine we can trade with the usual assumptions- liquidity, divisibility, etc.). Basically, you will have 17 separate data sets (one for each year 2000 through 2016).
I have sent the class a link to the data. The origin of this data can be found at the following website:
http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_ library.html
You will need to create "wrapper code" to import this data. Please note the special indicator in the data tables that is used for "missing data". Your program should handle this data properly. For our purposes a simple approach can be used. For example, take the immediately preceding and immediate following values and just average the two. Any reasonable approach is acceptable, but should be documented. If there is a large chunk of data missing, it may make sense to omit that security (i.e., work with fewer than 48). Once this preprocessing is done, store the cleaned up data for each year in a separate data file.
As a so-called market portfolio, we will use S&P500 daily returns in the same time period.
For interest rate, we will use the USD LIBOR. In a simple case (when it is assumed constant), you can just compute the average over the time period of concern. Yes there are different types of LIBOR (e.g., 3 month). Pick something reasonable, but remember to convert it to an effective daily rate.
Again, create nice data files to store these quantities, for separate years.
The rest of your code should taken a year, load the corresponding data, and process it.
There are several types of portfolios we will consider: in general, we will have a portfolio of 48 securities (possibly fewer- see above if large chunks of data are missing), defined by an appropriate weight vector; If we have a portfolio of M securities, the "naive" portfolio assumes equal weights $w_i = 1/M$ for each; the S&P 500 to be taken as the market portfolio; the LIBOR representing the risk-free asset. In terms of notation used below, recall MVP stands for the minimum variance portfolio (obtained from just the risky securities).

As you process the data you may not get results that match the theory. For example, we may not have the risk-free return R and the μ_{MVP} satisfy $R < \mu_{MVP}$; the S&P 500 may not be a good representative of the theoretical market portfolio, even if one were to exist. Keep in mind the time period involved had some unusual periods- the dot-com crash in 2000, and financial crisis in 2008, for example. It would be interesting to see, for example, if results for those years seem less "reasonable" than for so-called "normal" years.

Problems

1. **Basic Markovitz Portfolio Analysis:**

   We have data over 17 years (2000 through 2016). Repeat this experiment over a representative collection of years (no, you don't have to run your code 17 times!). We will treat each year as a separate, independent experiment. I expect to see commentary, regarding agreement with theory, consistency of results from year to year, and noting aberrations (perhaps?) in certain years.

   (a) Compute the sample mean and sample covariance matrix of returns for the 48 securities.

   (b) Compute (σ, μ) for the naive portfolio, the market portfolio, and the MVP. Randomly select a pair of securities and draw the feasible portfolio curve in the (σ, μ) plane assuming *no short selling*. Repeat this 5 times, to generate various curves. Superimpose all this in a plot.

   (c) Compute the efficient frontier and graph it. Highlight the portion of the graph that corresponds to no short selling. Draw a line from $(0, R)$ to (σ_M, μ_M) of the market portfolio. Determine the range of returns that correspond to no short selling.

   (d) Pick three points on the efficient frontier that are NOT the market portfolio, or MVP. If you sort them say $\mu_1 < \mu_2 < \mu_3$, confirm that $\sigma_1 < \sigma_2 < \sigma_3$, and that $\vec{w}_2$ can be obtained as a convex combination of $\vec{w}_1, \vec{w}_3$ (that means we can take a combination of portfolios 1 and 3 without short selling to obtain portfolio 2: it is ok if $\vec{w}_1, \vec{w}_3$ themselves involve short selling, I mean the combination of these composite portfolios does not involve short selling these two portfolios).

   (e) Is $R < \mu_{MVP}$?

   (f) Find the equation of the Capital Market Line.

   (g) Find the β for the MVP, the three portfolios you took on the efficient frontier, and the naive portfolio. Specifically, compute the covariance between the return of each of these and the market portfolio (i.e., S&P 500). Also recall $\beta = 0$ for the market portfolio. Graph μ versus β and see how close these points lie on a line.

   (h) Your analysis so far is based on averaged values of risk and return. Let us stay with these identified portfolios: the MVP, the market portfolio (i.e. S&P500), the naive portfolio, and the three you chose on the efficient frontier. Suppose you start with \$1 on January 1 for each of these. Graph their value in time for the year (superimpose the graphs), using the actual (not averaged) data.

2. **Sparse Portfolio Analysis:**

We are now interested in applying modifications to the standard Markovitz optimization problem. Conforming to the notation in the lecture notes, we now use ρ (not μ) to denote the expected return of a portfolio.

(a) In the above, you identified some special portfolios: the naive one, the three points on the efficient frontier, the MVP and the market portfolio. (Actually, you have multiple sets, one for each year you tested). This problem explores a simple, brute-force approach to sparsification. In other words, here I am proposing an "ad-hoc" approach. The real purpose of this is to explore the fact that the L^1-norm is taken as the sparsity cost function for practical purposes (i.e., so that we obtain convex optimization problems), although the actual sparsity measure is the L^0-norm. In reality, a portfolio vector with a maximal number of 0 entries would be desirable, and having a low L^1-norm doesn't necessarily help in that regard. Instead (as discussed in class), the L^1-norm cost (combined with the affine constraint $\sum w_i = 1$) actually results in a constraint on short-selling, which is also desirable (but is not what is meant by 'sparsity' in the literature). In any case, here we explore L^0-sparsity. Define the truncation operation g_k as $\vec{w}' = g_k(\vec{w})$ as follows:

- Sort the elements of $\vec{w}$ by descending value of $|w_i|$.
- Keep the k largest coefficients and 0 out the rest.
- At this point, if $\sum w_i' \leq 0$, we don't have a reasonable result, and reject k (let's define $\vec{w}'$ to be the vector of ∞ in that case). Otherwise, we normalize by a **positive** constant so the final vector $\vec{w}'$ satisfies $\sum w_i' = 1$. Note that even when this is done, we may have actually INCREASED $\|\vec{w}'\|_1$! However, we definitely have reduced $\|\vec{w}'\|_0$.

For example, if:

$$\vec{w} = [-0.3, 0.2, 1.1]^T$$

then

$$g_2(\vec{w}) = [-0.3, 0, 1.1]^T / 0.8 = [-3/8, 0, 11/8]^T$$

We have $\|\vec{w}\|_1 = 1.6$ and $\|\vec{w}'\|_1 = 1.75$. We also see the largest negative coefficient has been made bigger. So there is more short-selling, but there are fewer securities in the proposed portfolio (hence, more true 'sparsity'). A more careful study of what happened here is that one of the larger coefficients was negative. In any case, define the sparsification operation S_p, $p = 1, 2$ as:

$$\vec{w}' = S_p(\vec{w}) = g_k(\vec{w}) \quad \text{such that} \quad k = \arg\min \|\vec{w} - g_k(\vec{w})\|_p \leq 0.1 \|\vec{w}\|_p$$

For each of the portfolios in question, compute the S_1 and S_2 sparsified portfolios, the 0-norm for each (i.e., how many non-zero coefficients need to be retained), and the modified (σ, ρ) values. Compare the Sharpe ratios ρ/σ before and after this sparsification.

(b) We now consider sparse portfolio design using one of the 'sophisticated' approaches discussed in class, applied to the full set of 48 securities. Consider each year as a separate dataset. Let $\mathbf{R}$ be the $N \times M$ matrix of returns (row index is time, column index is security), and $\hat{\boldsymbol{\mu}}$ is the vector of sample mean returns. For example, if w is a weight vector for a portfolio, $\mathbf{R}w$ is return for the portfolio as a function of time. Your matrix $\mathbf{R}$ contains data for the 48 securities over one full year. Now compute the efficient sparse portfolio using the modified optimization problem:

$$w_{sparse} = \arg\min \left[\|\rho \mathbf{1}_N - \mathbf{R}w\|_2^2 + \tau \|w\|_1\right] \quad \text{s.t.} \quad w^T \hat{\mu} = \rho,\ w^T \mathbf{1}_M = 1$$

This will require you to find a suitable optimization program. Solve this, for fixed τ, for a number of ρ, and plot the (σ, ρ) points. You may need to play with parameters to get reasonable results (i.e., ρ, τ should not be too big or too small). Once you have reasonable results, try varying τ (using the same set of ρ values for each case) and superimpose the graphs, and observe sensitivity to τ. Test the sparsity for every identified $\vec{w}$ by finding $S_1(\vec{w})$, and obtain a scatter graph of $S_1(\vec{w})$ versus ρ and τ. **Note:** Depending on your code, the optimization program you use, and the computer platform you work on, solving this problem may take a long time. In that case, you don't need to run it for 17 years, but try it at least for a few. You might consider subsampling your data set: randomly select a subset of the rows of $\mathbf{R}$ to make the matrix smaller; if M is the number of securities (columns), you probably should have $N \geq 2M$, just as a common rule of thumb.

(c) Optional: Develop a neural net to solve the above optimization problem, then use it.