Homework #2

ELEC/IEDA3180 - Data Driven Portfolio Optimization Spring 2022/2023

Prof. Daniel Palomar, TAs: Amir Javaheri, Yifan Yu

2023-03-02

The HKUST Academic Honor Code applies. This assignment is to be done individually. Cheating **won't** be tolerated. Total marks: 110 points.

Problem 1 (60 points)

Suppose we have N stocks, for which, the log-returns at time t is denoted by $\mathbf{x}_t \in \mathbb{R}^N$. Consider the following linear factor model

$$\mathbf{x}_t = \boldsymbol{\alpha} + \boldsymbol{\beta}\mathbf{f}_t + \boldsymbol{\epsilon}_t, \;\; t = 1, 2, \ldots, T$$

where $\mathbf{f}_t \in \mathbb{R}^2$ denotes the values of the factors at time t, and $\boldsymbol{\alpha} \in \mathbb{R}^N$ and $\boldsymbol{\beta} \in \mathbb{R}^{N\times 2}$ are constant parameters.

In this excercise, we use the log-returns of 10 popular stocks as $\mathbf{x}_t$ starting from `2019-01-01` to `2022-12-31`. We also use the S&P500 index and the Bitcoin price within the same period, for the factors $\mathbf{f}_t$. The price data may be queried using the following code:

```
from pandas_datareader import data as pdr
import yfinance as yf

yf.pdr_override()

stocks = ['TSLA', 'AMZN', 'EBAY', 'AAPL', 'MSFT', 'META', 'GO
          'NFLX', 'IBM', 'NVDA']

indices = ['^GSPC', 'BTC-USD']

prices = pdr.get_data_yahoo(stocks, start="2019-01-01", end="
factor = pdr.get_data_yahoo(indices, start="2019-01-01", end=

print(factor['Adj Close'].head())
```

(a) Explain about the meaning of $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\mathbf{f}_t$, and $\boldsymbol{\epsilon}_t$.

(b) Find $\boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}^*$ by solving the following least-squares problem using an optional solver in python (e.g., CVXPY):

$$\{\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*\} = \underset{\boldsymbol{\alpha},\boldsymbol{\beta}}{\operatorname{argmin}} \quad \sum_{t=1}^{T} \|\mathbf{x}_t - \boldsymbol{\alpha} - \boldsymbol{\beta}\mathbf{f}_t\|^2$$

Note: You may need to interpolate the data first to fill in the NaN entries.

(c) Find $\mathbf{X}$ and $\mathbf{F}$, such that the above problem can be restated as follows

$$\boldsymbol{\Gamma}^* = \underset{\boldsymbol{\Gamma}}{\operatorname{argmin}} \|\mathbf{X} - \boldsymbol{\Gamma}\mathbf{F}\|_F^2$$

where $\boldsymbol{\Gamma} = [\boldsymbol{\alpha} \ \boldsymbol{\beta}] \in \mathbb{R}^{N\times 3}$.

(d) Find the optimal values $\boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}^*$ using the closed-form solution to the problem above (given in the `Py-session_iid_modeling` slides) and compare with the previous part.

(e) Draw a 3D scatter plot of points where the x-axis is $\boldsymbol{\beta}_1$ (first column of $\boldsymbol{\beta}$), the y-axis is $\boldsymbol{\beta}_2$ (second column of $\boldsymbol{\beta}$) and the z-axis is $\boldsymbol{\alpha}$.

Hint: You may use matplotlib for this purpose. You can also use the function `ax.text()` to add labels to the points (stocks).

(f) Based on the previous plot and the meaning of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, which stocks are less risky to invest on and why?

(g) (Bonus) Consider the statistical factor model where the factors $\mathbf{f}_t$ are not available. Assume the number of factors $K = 2$. Write a piece of code to estimate the factors $\mathbf{f}_t$ and the values of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Compare these parameters with those in part (b). Which method do you think works better and why?

Hint: Implement the algorithm given in the `Py-session_iid_modeling` lecture slides under the heading `Statistical factor model`.

Problem 2 (50 points)

The Sharpe ratio of a portfolio

$\mathbf{w} \in \mathbb{R}^N$ is defined as the ratio of its expected return to its volatility.

$$\mathsf{S} = \frac{\mathbf{w}^\top \boldsymbol{\mu}}{\sqrt{\mathbf{w}^\top \boldsymbol{\Sigma} \mathbf{w}}}$$

Consider the portfolio optimization method in Problem 1 in your previous homework and choose $\beta = 0.05$. We refer to this as the "designed portfolio".

(a) Consider the log-returns data in **Problem 1** for three stocks including 'AAPL', 'MSFT' and 'GOOGL'. Divide this data into two parts, one from `"2019-01-01"` to `"2021-12-31"`, which we call the training data and the other from `"2022-01-01"` to `"2022-12-31"`, which we refer to as the "test" data. Using the training set, compute the Sharpe ratio for the designed portfolio as a function of ρ_1 and ρ_2 assuming

$$\boldsymbol{\Sigma} = \rho_1 \hat{\boldsymbol{\Sigma}} + (1 - \rho_1)\, \mathsf{Diag}(\hat{\boldsymbol{\Sigma}})$$

$$\boldsymbol{\mu} = \rho_2 \hat{\boldsymbol{\mu}} + (1 - \rho_2) \frac{\mathbf{1}_N{}^\top \hat{\boldsymbol{\mu}}}{N} \mathbf{1}_N$$

where $\hat{\boldsymbol{\Sigma}}$ is the sample covariance matrix and $\hat{\boldsymbol{\mu}}$ is the sample mean.

(b) Use all possible pairs for ρ_1 and ρ_2 in $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$ and plot a heatmap of the Sharpe ratios. Find the pair (ρ_1, ρ_2) that gives the highest

Sharpe ratio (on the training data). (You may also plot this as an image via imshow).

(c) With the pair found in part (b), compute the Sharpe ratio for the designed portfolio on the test data.

(d) Compare the value of the test Sharpe ratio to the case where there is no shrinkage estimator used (For $(\rho_1, \rho_2) = (1, 1)$).

Note #1: make sure to write your code in a modular, readable way. Code organization will be taken into account for the grading. Use meaningful names for variables, create functions to organize your code as much as possible. In case of doubt on coding best practices, take a look at the Google's style guide. Note that you don't have to strictly follow that style, you can develop your own, but make sure it is understandable and you use it consistently.

Note #2: submit your code via canvas in zipped file (`.zip`) containing a Jupyter notebook (`.ipynb`) and also its exported version in the HTML (`.html`) format.