Problem Shrinkage

A point raised by Ledoit and Wolf (2003, 2004) is that the covariance matrix $\sum$ is poorly estimated by its sample counterpart and can be greatly improved by their shrinkage estimators when p (number of assets) is not small in comparison with n (number of observations), which is often the case in portfolio management.

Question

A shrinkage method can be applied to covariance matrix estimation as well. Let $X_1, \dots, X_n$ be i.i.d. p-vector observations with unknown mean and covariance matrix Σ. Let's denote $X = [X_1, \dots, X_n] \in \mathbb{R}^{p\times n}$. The sample covariance matrix reads as

$$S = \frac{1}{n} X \left(I - \frac{1}{n} 1 1^\top \right) X^\top,$$

where 1 is the n-vector of all ones. We will assume that the systematic movement of X_i can be explained by a q-factor model:

$$X_i = \beta f_i + \varepsilon_i$$

where $\beta \in \mathbb{R}^{p\times q}$ are constant loadings, $\{f_i\}$ are q-factors and $\{\varepsilon_i\}$ are idiosyncratic errors with mean 0 and variance σ^2. Furthermore, we assume the factors are uncorrelated to each other with

variance 1. The factors and idiosyncratic errors are also uncorrelated. The covariance matrix can also be written as

$$\Sigma = \beta\beta^\top + \sigma^2 I$$

We can perform PCA on S and select the q principle components with largest eigenvalues as $\hat{\beta}$ (scaled by the square root of eigenvalues), with the corresponding factor estimators $X^\top \hat{\beta}$. σ^2 can be estimated by the sample variance of the residuals after projecting X to the space orthonormal to $\hat{\beta}$. Let's define

$$F = \hat{\beta}\hat{\beta}^\top + \hat{\sigma}^2 I$$

We define the shrinkage estimator of Σ as

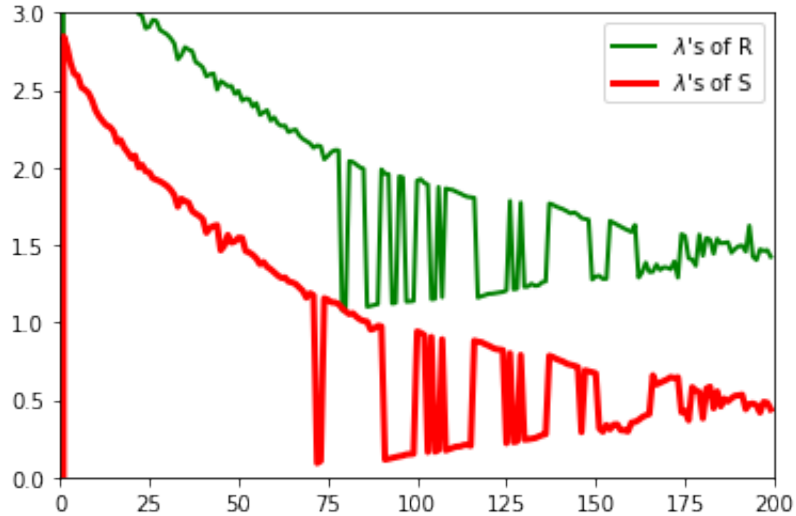
$$R(\alpha) = \alpha F + (1-\alpha) S$$

and choose the optimal α^* by minimizing the Frobenius norm $\|R(\alpha)\|_F$.

Simulate by sampling X_i from $\mathcal{N}(0, I)$, with $p = 200$ and $n = 400$. The number of factors $q = 3$. Plot the eigenvalue distribution of S and $R(\alpha^*)$. What can you tell from the plots? Why is the shrinkage a better estimator of the covariance matrix than the sample covariance matrix?

My solution

```
In [2]: import cvxpy as cp
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn.decomposition import PCA

# simulate X from normal dist. with size 200*400
X = np.random.normal(0,1,(200,400))

# number of factors is 3
pca_X_3 = PCA(3)
# reference for PCA: https://towardsdatascience.com/pca-using-python-s
cikit-learn-e653f8989e60
princomp_X_3 = pca_X_3.fit_transform(X)
principalDf = pd.DataFrame(data = princomp_X_3,
             columns = ['PrincipalComp 1', 'PrincipalComp 2', 'Princip
alComp 3'])

# reference for projection: https://stackoverflow.com/questions/178368
80/orthogonal-projection-with-numpy
Proj = np.dot(np.linalg.inv(X.T.dot(X)), X.T)
Proj = np.dot(X, Proj)
# Generate estimator with the sample variance of the residuals
# after projecting X to the space orthonormal to PCA loadings
sigma_square = np.var(X - Proj @ X)

# S
inner_term = np.eye(400) - (1/400) * np.dot(np.ones(400), np.ones(400)
.T)
S = (1/400) * np.dot(X, np.dot(inner_term, X.T))

# F
beta_hat = principalDf.to_numpy()
F = np.dot(beta_hat, beta_hat.T) + sigma_square * np.eye(200)

# Solve optimization problem
alpha = cp.Variable()
obj_fn = cp.Minimize(cp.norm(alpha * F + (1 - alpha) * S, "fro"))
problem = cp.Problem(obj_fn)
problem.solve()

S_eval = np.linalg.eigvals(S)
R_eval = np.linalg.eigvals(alpha.value * F + (1 - alpha.value) * S)

plt.plot(R_eval, label="$\lambda$'s of R", color='green',linewidth=2.0
)
plt.plot(S_eval, label="$\lambda$'s of S", color='red',linewidth=3.0)
plt.axis([0,200,0,3])
plt.legend()
plt.show()
```


3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
25
50
75
100
125
150
175
200
λ's of R
λ's of S

In [ ]: