NYU Courant MATH-GA 2708

Assignment4 Written Report

Yuanzhe Yang yy3460

Chien-Yueh Shih cs6380

Zedi Qiu zq2015

Problem 1

Readme:

In this prompt, we will build a pipeline to compute cointegration coefficient $\boldsymbol{\gamma}$ and perform relative analysis. We used the deque to compute and update the parameters efficiently without the need to loop through the dataset for regression analysis.

The key functionalities include:

a) Circular_queue: implement the deque data structure including operation like enqueue, dequeue with O(1) time complexity. In addition, the script will all the variables in each queue to help compute regression efficiently at each point in time.
b) Circular_queue_test: perform unit test for circular queue to make sure it operate data properly
c) Problem_a: find cointegrated trading pairs from a matrix of stock returns through performing Engle-Granger cointegration test.
d) Problem_b: compute and update the regression cointegration coefficient $\boldsymbol{\gamma}$ in an efficient way through queuing the variables in to the deque. The output will be the price series data, variables queued for computation, and the cointegration coefficient $\boldsymbol{\gamma}$.

The reason why we need a circular queue is that we need an effective way to rolling cointegration test while new tick comes in. Since we are facing a problem of large size data re-computation, we could hardly run regression module on the entire data array in the rolling progress. It would take much more time than we expected to do so, and may further lead us to miss the opportunity of trading.

1) Perform Granger-Engle Test.

The general Granger-Engle test includes two steps

Step 1: Run OLS Regression $Y_t = mx_t + b$, and compute the $\hat{z}_t = Y_t - \widehat{m}x_t - \hat{b}$

Step 2: Perform the Dickey-Fuller (DF) test, $\hat{z}_t = \gamma \hat{z}_{t-1} + \epsilon_t$

In problem a, we use module statsmodels to perform cointegration test. In our case, we set P-value to be 0.01. We get 618 pairs.

2) Run rolling cointegration using circular queue

In this part, we queued the variables to the deque and use it to obviate the need to re-compute the full regression from scratch

$$m = \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - \bar{x}^2}$$

$$b = \bar{y} - m\bar{x}$$

$$\bar{x} = \frac{1}{N}\sum_{i=t-N++1}^{t} x_i$$

We keep the variables in the deque and keep on appending in and popping out the data for cointegration coefficient calculation. In our case, we set the window size N to be 120.

$$\widehat{\gamma_t} = \frac{\sum_{i=t-N+2}^{t}(y_i - mx_i - b)(y_{i-1} - mx_{i-1} - b)}{\sum_{i=t-N+2}^{t}(y_{i-1} - mx_{i-1} - b)^2}$$

With future expansion of the formula,

$$\sum_{i=t-N+2}^{t}(y_{i-1}-mx_{i-1}-b)^2=\sum_{i=t-N+2}^{t}y_{i-1}^2+\hat{m}\sum_{i=t-N+2}^{t}x_{i-1}^2+2\hat{m}\hat{b}\sum_{i=t-N+2}^{t}x_{i-1}-2\hat{m}\sum_{i=t-N+2}^{t}x_{i-1}y_{i-1}-2\hat{b}\sum_{i=t-N+2}^{t}y_{i-1}+\hat{b}^2(N-1)$$

$$\sum_{i=t-N+2}^{t}(y_i-mx_i-b)(y_{i-1}-mx_{i-1}-b)=\sum_{i=t-N+2}^{t}y_iy_{i-1}-\hat{m}(\sum_{i=t-N+2}^{t}x_{i-1}y_i+\sum_{i=t-N+2}^{t}x_iy_{i-1})-\hat{b}(\sum_{i=t-N+2}^{t}y_i+\sum_{i=t-N+2}^{t}y_{i-1})+\hat{m}^2\sum_{i=t-N+2}^{t}x_ix_{i-1}+\hat{m}\hat{b}(\sum_{i=t-N+2}^{t}x_i+\sum_{i=t-N+2}^{t}x_{i-1})+\hat{b}(N-1)$$

Through this way, we don't need to loop over the whole dataset for parameter computing. We can put the data to deque and calculate the average values when the data fed in and reach the desired size. Notice that all the operations relative to our circular deque is O(1), since we store a local variable to track each sum of data/ mean of data as we enqueue and dequeue from the deque. In our case, our deque includes the mean of following nice values.

$x_i, y_i, x_iy_i, x_i^2, y_i^2, x_ix_{i-1}, y_iy_{i-1}, x_iy_{i-1}, x_{i-1}y_i$

For deque, we can add and remove the elements from both ends. The computation can be done in O(1).

We can use these nine mean values to compute the regression equation we mentioned above without going through the whole data. Only scaler multiplication is what really happening here, and we do not even need to compute variance and covariance between arrays, since we go one step further and represent the equation using computation only include mean values.

3) Result

For each cointegrated pairs that we detected from problem a, we are able to compute a matrix including original paired cumulative returns, mean value arrays in intermediate stage, and final gamma we got in a rolling setting. Namely we will get dataframe in shape 10000 * 12, which 10000 is the length of our original cumulative returns array and 12 represents all the attributes columns we want to show.

Since we run the regression in a rolling basis, we will not get those mean values and final gamma until the first period after our first-time window length.

Question 2.

a): Our objective function is $L = \frac{1}{2}x^T\Sigma x + \frac{1}{2}(x-x_0)^T\Lambda(x-x_0)$

$$\frac{\partial L}{\partial x} = \Sigma x + \Lambda(x-x_0) = 0.$$

we set the first order derivative to 0.

$$\Sigma x + \Lambda x = \Lambda x_0.$$

$$(\Sigma + \Lambda)x = \Lambda x_0$$

$$x^* = (\Sigma+\Lambda)^{-1}\Lambda x_0.$$

b): i we now have to solve the multiperiod version

$$\min_{x_1,\dots x_T} \sum_{t=1}^{T} \Big(\underbrace{\frac{1}{2}x_t^T\Sigma x_t}_{\text{variance}} + \underbrace{\frac{1}{2}(x_t-x_{t-1})^T\Lambda(x_t-x_{t-1})}_{TC}\Big)$$

since we want to minimize the objective function we are have a cost to go function.

First, we focus on last stage T, where our objective function becomes

$$\min_{x_T} \frac{1}{2}x_T^T\Sigma x_T + \frac{1}{2}(x_T-x_{T-1})^T\Lambda(x_T-x_{T-1})$$

using the same technique as previous question.

$$\frac{\partial L}{\partial x_T} = \Sigma x_T + \Lambda(x_T - x_{T-1}) = 0.$$

$$(\Sigma+\Lambda)x_T = \Lambda x_{T-1}$$

$$x_T^* = (\Sigma+\Lambda)^{-1}\Lambda x_{T-1}$$

We can represent our cost-to-go function J as well.

let G be $(\Sigma+\Lambda)^{-1}\Lambda$

G is a square, positive definite matrix.

$$J_T(x_{T-1}) = \min_{x_T^*} \frac{1}{2}x_T^T\Sigma x_T + \frac{1}{2}(x_T-x_{T-1})^T\Lambda(x_T-x_{T-1})$$

given we know x_{T-1}

$$= \frac{1}{2}(Gx_{T-1})^T\Sigma(Gx_{T-1}) + \frac{1}{2}(Gx_{T-1}-x_{T-1})^T\Lambda(Gx_{T-1}-x_{T-1})$$

$$= \frac{1}{2}x_{T-1}^TG^T\Sigma Gx_{T-1} + \frac{1}{2}[(G-I)x_{T-1}]^T\Lambda[(G-I)x_{T-1}]$$

$$= \frac{1}{2}x_{T-1}^TG^T\Sigma Gx_{T-1} + \frac{1}{2}x_{T-1}^T(G-I)^T\Lambda\underline{(G-I)x_{T-1}}$$

$$= \frac{1}{2}x_{T-1}^TG^T\Sigma Gx_{T-1} + \frac{1}{2}x_{T-1}^T(G^T-I^T)\Lambda Gx_{T-1} - \frac{1}{2}x_{T-1}^T(G^T-I^T)\Lambda x_{T-1}$$

$$G^T(\Sigma+\Lambda)G$$
$$= [(\Sigma+\Lambda)^{-1}\Lambda]^T(\Sigma+\Lambda)[(\Sigma+\Lambda)^{-1}\Lambda]$$
$$= \Lambda^T(\Sigma+\Lambda)^{-1}(\Sigma+\Lambda)(\Sigma+\Lambda)^{-1}\Lambda$$
$$= \Lambda^T(\Sigma+\Lambda)^{-1}\Lambda$$
$$= \Lambda^TG.$$
$$= G^T\Lambda.$$

$$= \frac{1}{2}x_{T-1}^TG^T\Sigma Gx_{T-1} + \frac{1}{2}x_{T-1}^TG^T\Lambda Gx_{T-1} - \underline{\frac{1}{2}x_{T-1}^T\Lambda Gx_{T-1}} - \frac{1}{2}x_{T-1}^TG^T\Lambda x_{T-1} + \underline{\frac{1}{2}x_{T-1}^T\Lambda x_{T-1}}$$

$$= \frac{1}{2}x_{T-1}^TG^T(\Sigma+\Lambda)Gx_{T-1} - \frac{1}{2}x_{T-1}^TG^T\Lambda x_{T-1} + \underline{\frac{1}{2}x_{T-1}^T\Lambda x_{T-1}} - \underline{\frac{1}{2}x_{T-1}^T\Lambda Gx_{T-1}}$$

$$= \cancel{\frac{1}{2}x_{T-1}^TG^T\Lambda x_{T-1}} - \cancel{\frac{1}{2}x_{T-1}^TG^T\Lambda x_{T-1}} + \frac{1}{2}x_{T-1}^T\Lambda x_{T-1} - \frac{1}{2}x_{T-1}^T\Lambda(\Sigma+\Lambda)^{-1}\Lambda x_{T-1}$$

$$= \frac{1}{2}x_{T-1}^TK_Tx_{T-1} \quad \text{where } K_T = \Lambda + \Lambda(\Sigma+\Lambda)^{-1}\Lambda$$

ii we want to minimize dynamic objective function.

$$\min_{x_1,\dots x_T} \sum_{t=1}^{T} \left(\frac{1}{2} x_t^T \Sigma x_t + \frac{1}{2}(x_t - x_{t-1})^T \Lambda (x_t - x_{t-1}) \right)$$

Using Bellman equation, our cost-to-go function can be represented by

$$J_t(x_{t-1}) = \min_{x_t} \left(\frac{1}{2} x_t^T \Sigma x_t + \frac{1}{2}(x_t - x_{t-1})^T \Lambda (x_t - x_{t-1}) + J_{t+1}(x_t) \right)$$ ← Bellman equation.

Assume we have an answer $x_t^* = L_t x_{t-1}$ and $J_t(x_{t-1}) = \frac{1}{2} x_{t-1}^T K_t x_{t-1}$.

WT prove by induction

we know that $J_T(x_{T-1}) = \frac{1}{2} x_{T-1}^T (\Lambda - \Lambda(\Sigma+\Lambda)^{-1}\Lambda) x_{T-1}$

$x_T^* = (\Sigma+\Lambda)^{-1}\Lambda x_{T-1}$

Try to find the pattern

$$J_{T-1}(x_{T-2}) = \max_{x_{T-1}} \left(\frac{1}{2} x_{T-1}^T \Sigma x_{T-1} + \frac{1}{2}(x_{T-1} - x_{T-2})^T \Lambda (x_{T-1} - x_{T-2}) + \frac{1}{2} x_{T-1}^T \underbrace{(\Lambda - \Lambda(\Sigma+\Lambda)^{-1}\Lambda)}_{K_T} x_{T-1} \right)$$

$$\frac{\partial L}{\partial x_{T-1}} = \Sigma x_{T-1} + \Lambda(x_{T-1} - x_{T-2}) + (\underbrace{\Lambda - \Lambda(\Sigma+\Lambda)^{-1}\Lambda}_{K_T}) x_{T-1} = 0.$$

$$[\Sigma + \Lambda + \underbrace{(\Lambda - \Lambda(\Sigma+\Lambda)^{-1}\Lambda)}_{K_T}] x_{T-1} = \Lambda x_{T-2}$$

$$[\Sigma + \Lambda + K_T] x_{T-1} = \Lambda x_{T-2}$$

$$x_{T-1}^* = [\Sigma + \Lambda + K_T]^{-1} \Lambda x_{T-2}$$

Now we plug x_{T-1}^* back to $J_{T-1}(x_{T-2})$

$$J_{T-1}(x_{T-2}) = \frac{1}{2} x_{T-1}^{*T} \Sigma x_{T-1}^* + \frac{1}{2}(x_{T-1}^* - x_{T-2})^T \Lambda (x_{T-1}^* - x_{T-2}) + \frac{1}{2} x_{T-1}^{*T} K_T x_{T-1}^*$$

$$= \frac{1}{2} x_{T-1}^{*T} (\Sigma+\Lambda) x_{T-1}^* - \frac{1}{2} x_{T-2}^T \Lambda x_{T-1}^* - \frac{1}{2} x_{T-1}^{*T} \Lambda x_{T-2} + \frac{1}{2} x_{T-2}^T \Lambda x_{T-2} + \frac{1}{2} x_{T-1}^{*T} K_T x_{T-1}^*$$

we know

$x_{T-1}^* = (\Sigma + \Lambda + K_T)^{-1} \Lambda x_{T-2}$

let $G = (\Sigma + \Lambda + K_T)^{-1} \Lambda$

$x_{T-1}^* = G x_{T-2}$

$x_{T-1}^{*T} (\Sigma + \Lambda + K_T) x_{T-1}^*$

$= x_{T-2}^T G^T \Lambda x_{T-2}$

combine

$$= \frac{1}{2} x_{T-1}^{*T} (\Sigma + \Lambda + K_T) x_{T-1}^* - \frac{1}{2} x_{T-2}^T \Lambda x_{T-1}^* - \frac{1}{2} x_{T-1}^{*T} \Lambda x_{T-2} + \frac{1}{2} x_{T-2}^T \Lambda x_{T-2}$$

$$= \frac{1}{2} \cancel{x_{T-2}^T G^T \Lambda x_{T-2}} - \frac{1}{2} x_{T-2}^T \Lambda G x_{T-2} - \frac{1}{2} \cancel{x_{T-2}^T G^T \Lambda x_{T-2}} + \frac{1}{2} x_{T-2}^T \Lambda x_{T-2}$$

$$= \frac{1}{2} x_{T-2}^T (\Lambda - \Lambda(\Sigma + \Lambda + K_T)^{-1}\Lambda) x_{T-2}.$$

we find the pattern where

$x_T = (\Sigma+\Lambda)^{-1}\Lambda x_{T-1}$ $\quad$ $K_T = \Lambda - \Lambda(\Sigma+\Lambda)^{-1}\Lambda$

$x_{T-1} = [\Sigma + \Lambda + K_T]^{-1}\Lambda x_{T-2}$ $\quad$ $K_{T-1} = \Lambda - \Lambda(\Sigma + \Lambda + K_T)^{-1}\Lambda.$

$$x_t^* = \underbrace{[\Sigma + \Lambda + K_{t+1}^*]^{-1}\Lambda}_{G} x_{t-1} \qquad K_t^* = \Lambda - \Lambda(\Sigma + \Lambda + K_{t+1}^*)^{-1}\Lambda$$

$$J_t(x_{t-1}) = \min_{x_t} \left(\frac{1}{2} x_t^T \Sigma x_t + \frac{1}{2}(x_t - x_{t-1})^T \Lambda (x_t - x_{t-1}) + J_{t+1}(x_t) \right).$$

by the pattern.

$$J_t(x_{t-1}) = \frac{1}{2} x_t^T (\Sigma+\Lambda) x_t - \frac{1}{2} x_{t-1}^T \Lambda x_t - \frac{1}{2} x_t^T \Lambda x_{t-1} + \frac{1}{2} x_{t-1}^T \Lambda x_{t-1} + \frac{1}{2} x_t^T K_{t+1}^* x_t$$

$$= \frac{1}{2} x_t^T(\Sigma+\Lambda+K_{t+1}^*)x_t^T - \frac{1}{2}x_{t-1}^T\Lambda x_t - \frac{1}{2}x_t^T\Lambda x_{t-1} + \frac{1}{2}x_{t-1}^T\Lambda x_{t-1}$$

$$= \frac{1}{2} x_{t-1}^T G^T \cancel{(\Sigma+\Lambda+K_{t+1}^*)G} x_{t-1} - \frac{1}{2}x_{t-1}^T\Lambda G x_{t-1} - \cancel{\frac{1}{2} x_{t-1}^T G^T\Lambda x_{t-1}} + \frac{1}{2}x_{t-1}^T\Lambda x_{t-1}$$

$$= \frac{1}{2} x_{t-1}^T \underbrace{[\Lambda - \Lambda(\Sigma+\Lambda+K_{t+1}^*)^{-1}\Lambda]}_{K_t^*} x_{t-1}$$

we finally prove that $x_t^* = [\Sigma+\Lambda+K_{t+1}^*]^{-1}\Lambda x_{t-1}$ $\qquad K_t^* = \Lambda - \Lambda(\Sigma+\Lambda+K_{t+1}^*)^{-1}\Lambda$

$L_t = [\Sigma+\Lambda+K_{t+1}]^{-1}\Lambda$ $\qquad K_t = \Lambda-\Lambda(\Sigma+\Lambda+K_{t+1})^{-1}\Lambda.$

given $K_T = \Lambda - \Lambda(\Sigma+\Lambda)^{-1}\Lambda.$

$J_t(x_{t-1}) = \frac{1}{2} x_{t-1}^T K_t x_{t-1}.$

c): Now we extend our problem to Infinite the horizon.

we shall use the same technique.

Bellman equation: $J_t(x_{t-1}) = \min_{x_t} \left(\frac{1}{2}x_t^T\Sigma x_t + \frac{1}{2}(x_t-x_{t-1})^T\Lambda(x_t-x_{t-1}) + \theta\cdot J_{t+1}(x_t)\right).$

Assume $x_t^* = L_t x_{t-1}$

$J_t(x_{t-1}) = \frac{1}{2} x_{t-1}^T K_t x_{t-1}$

$$\frac{\partial J}{\partial x_t} = \Sigma x_t + \Lambda(x_t - x_{t-1}) + \theta K_{t+1} x_t = 0.$$

$$\Rightarrow (\Sigma+\Lambda+\theta K_{t+1})x_t = \Lambda x_{t-1}$$

$$x_t^* = \underbrace{(\Sigma+\Lambda+\theta K_{t+1})^{-1}}_{G}\Lambda x_{t-1}.$$

$$\frac{1}{2} x_{t-1}^T K_t x_{t-1} = \frac{1}{2}x_t^{*T}\Sigma x_t + \frac{1}{2}(x_t^*-x_{t-1})^T\Lambda(x_t^*-x_{t-1}) + \frac{1}{2}\theta x_t^{*T}K_{t+1}x_t^*$$

$$= \frac{1}{2}x_t^{*T}(\Sigma+\Lambda+\theta K_{t+1})x_t^* - \frac{1}{2}x_{t-1}^T\Lambda x_t^* - \frac{1}{2}x_t^{*T}\Lambda x_{t-1} + \frac{1}{2}x_{t-1}^T\Lambda x_{t-1}$$

$$= \cancel{\frac{1}{2}x_{t-1}^T G^T\Lambda x_{t-1}} - \frac{1}{2}x_{t-1}^T\Lambda G x_{t-1} - \cancel{\frac{1}{2}x_{t-1}^T G^T\Lambda x_{t-1}} + \frac{1}{2}x_{t-1}^T\Lambda x_{t-1}$$

$$= \frac{1}{2}x_{t-1}^T(\Lambda - \Lambda G)x_{t-1}$$

$$= \frac{1}{2}x_{t-1}^T[\Lambda - \Lambda(\Sigma+\Lambda+\theta K_{t+1})^{-1}\Lambda]x_{t-1}$$

$$\Rightarrow K_t = \Lambda - \Lambda(\Sigma+\Lambda+\theta K_{t+1})^{-1}\Lambda.$$

Since in Infinite the horizon, we discount the future optimized cost/value by θ, similarly we have to take into discount when we update our x and K.

3. GP has a objective function look like this:

$$\max_{x_0,\dots,x_t} E_0\Big[\sum_t \Big[(1-p)^{t+1}\underbrace{(x_t^T r_{t+1} - \frac{\gamma}{2}x_t^T\Sigma x_t)}_{\text{MVO}} - \underbrace{\frac{(1-p)^t}{2}\Delta x_t^T \Lambda \Delta x_t}_{\text{Transaction cost}}\Big]\Big]$$

according to MPO scheme, we are essentially solving the following equation.

obtain the equivalent problem

$$\begin{array}{ll} \text{maximize} & \sum_{\tau=t+1}^{t+H}\Big(\hat{r}_{\tau|t}^T w_\tau - \gamma_\tau \psi_\tau(w_\tau) \\ & \qquad - \hat{\phi}_\tau^{\text{hold}}(w_\tau) - \hat{\phi}_\tau^{\text{trade}}(w_\tau - w_{\tau-1})\Big) \\ \text{subject to} & \mathbf{1}^T w_\tau = 1, \quad w_\tau - w_{\tau-1} \in \mathcal{Z}_\tau, \quad w_\tau \in \mathcal{W}_\tau, \\ & \tau = t+1,\dots,t+H, \end{array} \qquad (5.2)$$

The main statement of MPO is that at the t, we optimize over the future trajectory as part of planning exercise, just to be sure that we do not take any trade which will lead to a bad future position. However, we are only taking the first step trade.

So, The optimization function in GP can be solved by MPO as following

for $t = 0, \dots, \infty$:

$$\max_{x_t,\dots} \sum_{J=t+1}^{\infty} \Big((1-p)^{J-t}(\hat{r}_{J|t}^T x_{J-1} - \frac{\gamma}{2}x_{J-1}^T\Sigma x_{J-1}) - \frac{(1-p)^{J-t-1}}{2}(x_J - x_{J-1})^T\Lambda(x_J - x_{J-1})$$

b). (i) in order to constraint our portfolio being long only, we need adding constraint $Gx \le h$.

where $G = \underbrace{\begin{bmatrix} -1 & & \\ & \ddots & \\ & & -1 \end{bmatrix}}_{N} N \quad \begin{bmatrix} x^1 \\ \vdots \\ x^n \end{bmatrix} N \quad h \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} N.$

(ii) we would change our transaction cost to $\Big[\frac{1}{2}\gamma(x_J)^2 + \varepsilon|x_J - x_{J-1}|^T I + \tilde{\eta}(x_J - x_{J-1})^T(x_J - x_{J-1})\Big]$

where $\tilde{\eta} = \eta - \frac{1}{2}\gamma$

η is temporary impact parameter and γ is permanent impact parameter.

(c) Yes because although we are solving a quadratic programming problem assuming both permanent and temporary impact are linear, we still can not have a closed solution to the objective function, since we are facing linear inequality constraints. Since, we are essentially facing a convex programming problem where our objective function and inequality constraint has function in convex form. So we do have a unique solution, although we might need to solve it numerically.

Problem 4:
In this problem, we are asked to try out a module named cvxportfolio to do a portfolio optimization.

Since the module is out-of-date and lack of maintenance, we fail to run the example code successfully on our end. However, we still try to interpret the code as following.

Single Period Optimization:
Step 1: loading variables
Sigmas: are defined by daily volatility, which is calculated by daily abs difference between open price and close price, note that we need to use sigma to simulate our transaction cost, which will be later implemented in our trading policy
Returns: are just common returns but filter out the bad stocks and bad dates
Volume: similar to sigmas, volumes are also variable when we simulate transaction cost

No matter we are in single period or multiperiod, we need return estimators of next period/ future periods to compute our optimization. So, we still need to generate return estimators.

We further decompose the risk/vol of our historical return matrix using factor models. We use the specific shrinkage method correspond to eigenvalues and eigenvectors.

The risk factor data includes 15 most significant eigenvectors and eigenvalues, where exposures represent eigenvectors, factor sigma represents eigenvalues, and idyo represent idiosyncratic risks c computed by rest eigenvectors

Step 2:
Our second step is to define our policy based on Lagrange multipliers. This is because we include two Lagrange multipliers in our optimization equation – risk aversion parameter and transaction cost parameter. We need to grid search a combination of these two parameters so that we end up get a good simulating result.

Multiple Period Optimization:
In our setting of MPO, instead of using current return estimate as we did in SPO, we want an array of return estimates of time tau starting from current time t towards to the end of holding period

We are essentially doing the same thing as we have done in SPO, what slightly changed is here we need a few more input parameters to our Optimizer to generate our policy. We need to pass in a return forecast and lookahead_period, which can be found in boyd paper. This is how we interpret the MPO cvxportfolio.

Problem 5

a) Modify Black-Litterman with T-costs

With general Black-Litterman framework, we have

$$\Pi = \boldsymbol{\mu} + \boldsymbol{\mathcal{V}} \textbf{ with } \boldsymbol{\mathcal{V}} \sim \mathbf{N}(\mathbf{0}, \boldsymbol{\tau\Sigma})$$

$$q = P\boldsymbol{\mu} + \boldsymbol{\varepsilon} \; \boldsymbol{with} \; \boldsymbol{N} \sim \mathbf{N}(\mathbf{0}, \boldsymbol{\Omega})$$

In our unit test in the MVO example, we make the assumption that

Tau = 0.02 to reflect our confidence to the market equivalent

Exepcted return from different asset class: [.12, .10, .07, .03]

$$\begin{aligned} \mu_{BL} &= (X'V^{-1}X)X'V^{-1}y \\ &= \Pi + \boldsymbol{\tau\Sigma} P'[P\boldsymbol{\tau\Sigma} P' + \Omega]^{-1}[q - P\Pi] \end{aligned}$$

$$\sum BL = (1 + \boldsymbol{\tau})\boldsymbol{\Sigma} - \boldsymbol{\tau}^2\boldsymbol{\Sigma}\, P'[P\boldsymbol{\tau\Sigma} P' + \Omega]^{-1}$$

Through these modified functions, we could calculate the expected returns and the covariance matrix for Black-litterman model.

b) Modify the code so it can handle the Almgren-style market impact.

In our code, we make the unit test with assumption that similar to the Almgen 2000. Both linear and permanent impact.

$$g(v) = \boldsymbol{\gamma v}$$

$$\boldsymbol{h}(\boldsymbol{v}) = \boldsymbol{\varepsilon}\,\mathbf{sgn(v)} + \boldsymbol{\eta v}$$

With MVO subject to the transaction cost we have

$$h^* = argmax_h E[h' - c(h, h0)] - \frac{\lambda}{2} var[h'r]$$