Assignment2

Name: Meng Nan

UID: 3030036376

(i) Show the $E(y(n))$

$$E(y(n)) = E\left[\sum_{j=1}^{J} w_j x_j(n)\right] = \sum_{j=1}^{J} w_j E[x_j(n)] = \boldsymbol{w}^T E[\boldsymbol{x}(n)] = \boldsymbol{w}^T \boldsymbol{\mu}_x$$

(ii) Show the $var(y(n))$

$$var(y(n)) = E\left(\left(y(n) - E(y(n))\right)^2\right) = E\left(y^2(n) - 2y(n)E(y(n)) + E^2(y(n))\right)$$

$$= E((\boldsymbol{w}^T\boldsymbol{x}(n))^2 - 2(\boldsymbol{w}^T\boldsymbol{x}(n)\boldsymbol{\mu}_x\boldsymbol{w}) + (\boldsymbol{w}^T\boldsymbol{\mu}_x)^2)$$

$$= \boldsymbol{w}^T E((\boldsymbol{x}(n) - \boldsymbol{\mu}_x)^2)\boldsymbol{w} = \boldsymbol{w}^T E[\overline{\boldsymbol{x}}(n)\overline{\boldsymbol{x}}(n)^T]\boldsymbol{w} = \boldsymbol{w}^T \boldsymbol{C}_{xx}\boldsymbol{w}$$

(iii) Show the optimum weight vector $\boldsymbol{w}^*$

$$L(\boldsymbol{w}, \lambda_1, \lambda_2) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{C}_{xx}\boldsymbol{w} - \lambda_1(\boldsymbol{w}^T\boldsymbol{1}_J - 1) - \lambda_2(\boldsymbol{w}^T\boldsymbol{\mu}_x - t)$$

The derivation of the L is:

$$\frac{\partial L(\boldsymbol{w}, \lambda_1, \lambda_2)}{\partial \boldsymbol{w}} = \boldsymbol{w}^T\boldsymbol{C}_{xx} - \lambda_1 \boldsymbol{1}_J{}^T - \lambda_2 \boldsymbol{\mu}_x{}^T = 0$$

$$\boldsymbol{C}_{xx}\boldsymbol{w} = \lambda_1\boldsymbol{1}_J + \lambda_2\boldsymbol{\mu}_x \quad \textbf{(1)}$$

So:

$$\boldsymbol{w}^* = \boldsymbol{C}_{xx}^{-1}(\lambda_1\boldsymbol{1}_J + \lambda_2\boldsymbol{\mu}_x)$$

For the equation (1), we can get:

$$\boldsymbol{w} = \lambda_1\boldsymbol{C}_{xx}^{-1}\boldsymbol{1}_J + \lambda_2\boldsymbol{C}_{xx}^{-1}\boldsymbol{\mu}_x \quad \textbf{(2)}$$

We multiply $\boldsymbol{1}_J^T$ on the both side of the equation (2):

$$\boldsymbol{1}_J^T\boldsymbol{w} = \lambda_1\boldsymbol{1}_J^T\boldsymbol{C}_{xx}^{-1}\boldsymbol{1}_J + \lambda_2\boldsymbol{1}_J^T\boldsymbol{C}_{xx}^{-1}\boldsymbol{\mu}_x$$

$$1 = \lambda_1 a + \lambda_2 b \quad \textbf{(3)}$$

We multiply $\boldsymbol{\mu}_x^T$ on the both side of the equation (2):

$$\boldsymbol{\mu}_x^T\boldsymbol{w} = \lambda_1\boldsymbol{\mu}_x^T\boldsymbol{C}_{xx}^{-1}\boldsymbol{1}_J + \lambda_2\boldsymbol{\mu}_x^T\boldsymbol{C}_{xx}^{-1}\boldsymbol{\mu}_x$$

Then we can get:

$$t = \lambda_1 b + \lambda_2 c \quad \textbf{(4)}$$

From the equation (3) and (4), we can easily get the result:

$$\begin{cases} 1 = \lambda_1 a + \lambda_2 b & (3) \\ t = \lambda_1 b + \lambda_2 c & (4) \end{cases}$$

$a(4) - b(3)$:

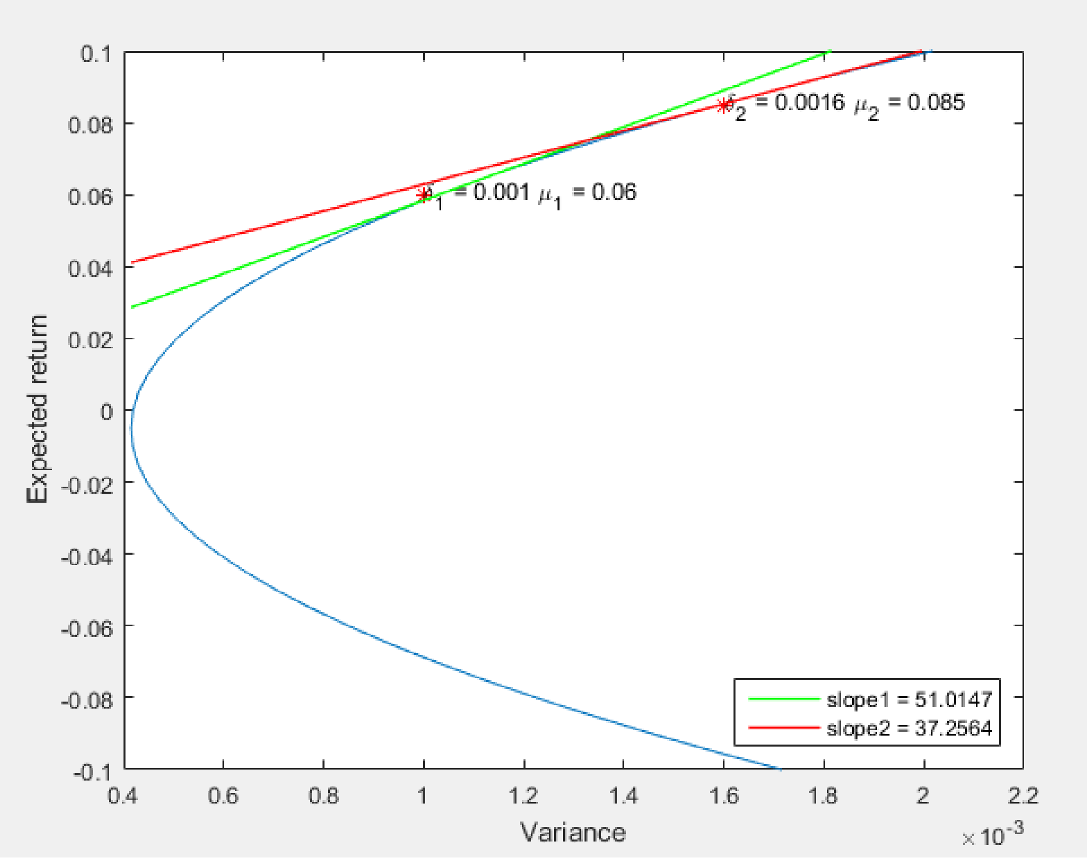
$$\lambda_2^* = \frac{at - b}{ac - b^2}$$

$c(3) - b(4)$:

$$\lambda_1^* = \frac{c - bt}{ac - b^2}$$

Part 2

(i) The graph is shown as follows:



From the graph, we can easily know that:

The best expected return $\mu_1 = 0.06$ when the standard deviation of the portfolio is $\sigma_1 = 10^{-3}$

The best expected return $\mu_2 = 0.085$ when the standard deviation of the portfolio is $\sigma_2 = 1.6 \times 10^{-3}$

(ii) From the graph, we can easily know that (μ_1, σ_1) gives a better portfolio, since the slope at that point is larger than the slope at the point (μ_2, σ_2)

Appendix

ass2.m

```
clear all;
close all;
addpath('ARMAX_GARCH_K_Toolbox');
% Reset random stream for reproducibility.
rng(0,'twister');
JAssets = 10; % desired number of assets
% Generate means of returns between -0.02 and 0.05.
a = -0.01; b = 0.05;
mean_return = a + (b-a).*rand(JAssets,1);
% Generate standard deviations of returns between 0.008 and 0.006.
a = 0.08; b = 0.06;
stdDev_return = a + (b-a).*rand(JAssets,1);
%%
Ntime=200;
%% X: Each row is a time-instant. Each Column is an asset.
X=zeros(Ntime,JAssets);
for j=1:JAssets
    X(:,j)=mean_return(j)+stdDev_return(j)*randn(Ntime,1);
end
%%
%% MF: mean forecast
%% VF: forecast of variance
MFpred=zeros(JAssets,1);
VFpred=zeros(JAssets);
for j=1:JAssets
    %% For each of the variables, fit the ARMA(1,1)-GARCH(1,1) model
    [parameters, stderrors, LLF, ht, resids, summary] =
garch(X(:,j),'GARCH', 'GAUSSIAN',1,1,0,1,1,0,[]);
    %% 1-step ahead Prediction of the mean and covariance of return
    [MFpred(j), VFpred(j,j), ~, ~] = garchfor2(X(:,j), resids, ht,
parameters, 'GARCH', 'GAUSSIAN',1,1,1,1,1);
end


a=ones(JAssets,1)'*(VFpred\ones(JAssets,1));
b=ones(JAssets,1)'*(VFpred\MFpred);
c=MFpred'*(VFpred\MFpred);

target=[-.1:5e-3:.1]';
risk=zeros(length(target),1);
for j=1:length(target)
   delta=a*c-b^2;
    lambda1=(c-b*target(j))/delta;
    lambda2=(a*target(j)-b)/delta;
    w=VFpred\(lambda1*ones(JAssets,1)+lambda2*MFpred);
    risk(j)=w'*VFpred*w;
end
plot(risk,target);
ylabel('Expected return')
xlabel('Variance')

%% Write your own code;
Delta = 4*10^-5;
delta1 = 10^-3;
index1 = find(abs(risk - delta1) < Delta);
mu_1 = max(target(index1));
```

```
delta2 = 1.6 * 10^-3;
index2 = find(abs(risk - delta2) < Delta);
mu_2 = max(target(index2));

hold on
plot([delta1,delta2],[mu_1,mu_2],'r*')
text(delta1,mu_1,['\delta_1 = ' num2str(delta1) ' \mu_1 = ' num2str(mu_1)])
text(delta2,mu_2,['\delta_2 = ' num2str(delta2) ' \mu_2 = ' num2str(mu_2)])


%% find the better one
index_1 = find(target == mu_1)
index_2 = find(target == mu_2)

Y1 = mu_1;
X1 = risk(index_1);
Y2 = mu_2;
X2 = risk(index_2);

Y1_1 = target(index_1 + 1);
X1_1 = risk(index_1 + 1);
Y2_1 = target(index_2 + 1);
X2_1 = risk(index_2 + 1);


slope_1 = ((Y1_1+Y1)/2 - Y1) / ((X1_1+X1)/2 - X1);
slope_2 = ((Y2_1+Y2)/2 - Y2) / ((X2_1+X2)/2 - X2);

RangeX = max(risk) - min(risk);
X1_start = min(risk);
Y1_start = Y1 + slope_1*(X1_start - X1);
X1_end = max(risk);
Y1_end = Y1 + slope_1*(X1_end - X1);

X2_start = min(risk);
Y2_start = Y2 + slope_2*(X2_start - X2);
X2_end = max(risk);
Y2_end = Y2 + slope_2*(X2_end - X2);

a = plot([X1_start,X1_end],[Y1_start,Y1_end],'g-','LineWidth',1);
b = plot([X2_start,X2_end],[Y2_start,Y2_end],'r-','LineWidth',1);
axis([0.4*10^-3, 2.2*10^-3, -0.1, 0.1])

L1 = ['slope1 = ' num2str(slope_1)];
L2 = ['slope2 = ' num2str(slope_2)];
legend([a,b],L1,L2,'Location','southeast');
```