**Assignment2**

Name: Meng Nan

UID: 3030036376

(i) Show the

(ii) Show the

(iii) Show the optimum weight vector

The derivation of the L is:

**(1)**

**So:**

For the equation (1), we can get:

**(2)**

We multiply on the both side of the equation (2):

**(3)**

We multiply on the both side of the equation (2):

Then we can get:

**(4)**

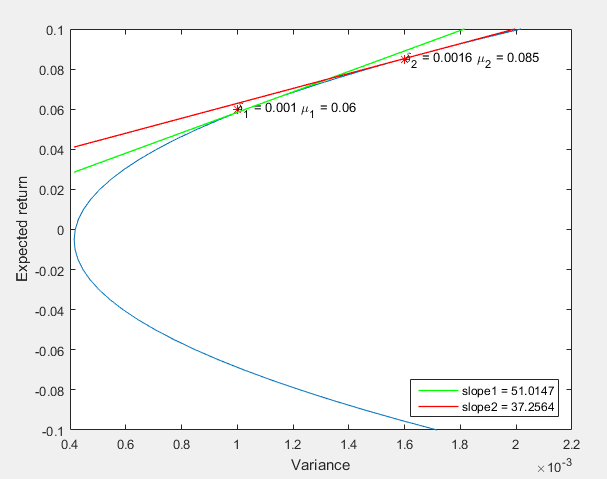
From the equation (3) and (4), we can easily get the result:

:

:

**Part 2**

(i) The graph is shown as follows:



From the graph, we can easily know that:

The best expected return when the standard deviation of the portfolio is

The best expected return when the standard deviation of the portfolio is

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

**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');