Computational Economics: Problem Set 5

Yangming Bao, ID: 5601239
Cheung Ying Lun, ID: 5441897

Problem a

Solution see `Q1.m`.

Problem b

1. Substituting the budget constraint into the objective function,

$$\max_{\{\alpha_i\}_{i=1}^n} \mathbb{E}\left[\left\{\frac{1}{1-\gamma}\left[(1+r^f+\sum_{i=1}^n \alpha_i(r_i-r^f)\right]W_0 - W_{\min}\right\}^{1-\gamma}\right]. \tag{1}$$

   The FOC wrt α_i would be

$$\mathbb{E}\left[\left((1+r^f+\sum_{i=1}^n \alpha_i(r_i-r^f))W_0 - W_{\min}\right)^{-\gamma} W_0(r_i-r^f)\right] \stackrel{!}{=} 0. \tag{2}$$

2. Assume the limiting case that $\gamma = 1$ (log utility), the optimal portfolio holdings are reported in the following table.

	$W_{\min}$					
α	0	10	20	30	40	50
α_1	1.280	1.155	1.029	0.904	0.778	0.653
α_2	0.640	0.577	0.515	0.452	0.389	0.326

Portfolio holdings decrease when $W_{\min}$ increases. This is because when the minimum wealth level one would like to achieve increases, one cannot take so much risk, since a bad shock would be very costly in terms of utility. One becomes more risk averse in some sense.

3. The optimal portfolio holdings with constraints are reported in the following table.

	$W_{\min}$					
α	0	10	20	30	40	50
α_1	1.000	1.000	1.000	0.904	0.778	0.653
α_2	0.720	0.622	0.523	0.452	0.389	0.326

The constraint is binding when $W_{\min} \leq 20$. The agent thus holds more the second asset than in the unconstrained case to take more risk when the constraint is binding.

Problem c

1. $\mathbb{E}[y_1] = 1.02$, $\mathbb{E}[y_2] = 1.0317$.

2. For $\gamma = 1.5$,

$$\mathbb{E}\left[\frac{y_1^{1-\gamma}-1}{1-\gamma}\right] = 0.0197 > -0.0158 = \mathbb{E}\left[\frac{y_2^{1-\gamma}-1}{1-\gamma}\right]. \tag{3}$$

Thus the agent would choose project 1.

3. Solving

$$\mathbb{E}\left[\frac{y_1^{1-\gamma}-1}{1-\gamma}\right] = \mathbb{E}\left[\frac{y_2^{1-\gamma}-1}{1-\gamma}\right] \tag{4}$$

for γ yields $\widehat{\gamma} = 0.3663$.

Question a

Q1.m

```
% Q1
cepath='/Users/baoyangming/Dropbox/gsefm/2015SoSe/Computational Economics/
    Applied Computational Economics and Finance/compecon/';
path([cepath 'cetools;' cepath 'cedemos'],path);
clc;clear

%% functions
f1 = @(x)x.^4;
f2 = @(x)x.^6;
f3 = @(x)1./(1+x.^2);

%% Monte carlo

montec11 = montec(f1,100);
montec12 = montec(f1,1000);
montec13 = montec(f1,10000);
montec14 = montec(f1,50000);

montec21 = montec(f2,100);
montec22 = montec(f2,1000);
montec23 = montec(f2,10000);
montec24 = montec(f2,50000);

montec31 = montec(f3,100);
montec32 = montec(f3,1000);
montec33 = montec(f3,10000);
montec34 = montec(f3,50000);

%% Gaussian Quadrature
n = [2 3 4 5 7];
q1 = zeros(5,1);
for ii=1:5
    q1(ii) = gaussianq(f1,n(ii));
end

q2 = zeros(5,1);
for ii=1:5
```

```
    q2(ii) = gaussianq(f2,n(ii));
end

q3 = zeros(5,1);
for ii=1:5
    q3(ii) = gaussianq(f3,n(ii));
end
```

montec.m

```
function intemc = montec(f,n)
% Monte carlo Integration

sum = 0;
x = randn(n,1);

for ii=1:n
    fval = feval(f,x(ii));
    sum = fval+sum;
end


intemc = sum/n;
```

gaussianq.m

```
function q = gaussianq(f,n)

mu = 0;
var = 1;
[x,w] = qnwnorm(n,mu,var);
fval = feval(f,x);

if size(fval,2)>1
    q = fval*w;
else
    q = fval'*w;
end
```

Question 2

`Q2.m`

```
clear , clc
close all
addpath('/Users/YingLun/Documents/Dropbox/Academic/Postgraduate/GSEFM/PhD/year 2/summersemester/Computational Economics/Applied Computational Economics and Finance/compecon/CEtools');

%% Parameters
rf       = 0.02;

mu       = [0.04,0.06];
Sig1     = 0.1;
Sig2     = 0.2;
rho      = 0.5;
Sigma    = [Sig1^2,rho*Sig1*Sig2;rho*Sig1*Sig2,Sig2^2];

W0       = 100;
Wmin     = (0:10:50)';

gamma    = 1;

tole     = 1e-10;
told     = 1e-10;
maxiter  = 1e7;
cc       = [tole;told;maxiter];

%% Solving unconstrained problem
n        = [7,7];
[r,w]    = qnwnorm(n,mu,Sigma);
al_hat   = zeros(2,length(Wmin));
ini_al   = [0;0];
for ii=1:length(Wmin)
    fun            = @(alpha)ExpR(r,rf,W0,Wmin(ii),alpha,gamma,w);
    al_hat(:,ii)   = broyden(fun,ini_al,cc);
    ini_al         = al_hat(:,ii);
end

%% Solving constrained problem
```

```
amin       = 0;
amax       = 1;
% optset('ncpsolve','type','minmax')
% optset('ncpsolve','maxit',100)
% optset('ncpsolve','showiters',false)
al_hat2  = zeros(2,length(Wmin));
ini_al   = [0.5;0.5];
for ii=1:length(Wmin)
    fun              = @(alpha)ExpR(r,rf,W0,Wmin(ii),alpha,gamma,w);
    al_hat2(:,ii)    = ncpsolve(fun,amin,amax,ini_al);
    ini_al           = al_hat2(:,ii);
end
```

ExpR.m

```
function [ExpOut,fjac] = ExpR(r,rf,W0,Wmin,alpha,gamma,w)
%This function computes the expectation given parameters.
%   INPUT:
%         r: Txn matrix of returns
%        rf: scalar of risk-free rate
%        W0: scalar of initial wealth
%      Wmin: scalar of minimum wealth
%     alpha: nx1 vector of portfolio weights
%     gamma: scalar of relative risk aversion coefficient
%         w: Tx1 vector of probabilities
%
%   OUTPUT:
%    ExpOut: nx1 vector of expectation
%      fjac: nxn matrix of Jacobian

n         = size(r,2);
ExpOut    = zeros(n,1);
for ii=1:n
    ExpOut(ii)       = w'*FOC(r,rf,W0,Wmin,alpha,gamma,ii);
end

% if need fjac
if nargout>1
    I          = eye(n);
    fjac       = zeros(n);
    for ii=1:n
        if -alpha(ii)<ExpOut(ii) && ExpOut(ii)<1-alpha(ii)
```

```
            for jj=1:n
                fjac(ii,jj) = w'*SOC(r,rf,W0,Wmin,alpha,gamma,ii,jj);
            end
        else
            fjac(ii,:)   = -I(ii,:);
        end
    end
end
end
```

FOC.m

```
function output = FOC(r,rf,W0,Wmin,alpha,gamma,ii)
%This function computes the FOC values given parameters.
%   INPUT:
%        r: Txn matrix of returns
%       rf: scalar of risk-free rate
%       W0: scalar of initial wealth
%     Wmin: scalar of minimum wealth
%    alpha: nx1 vector of portfolio weights
%    gamma: scalar of relative risk aversion coefficient
%       ii: scalar of asset label
%
%   OUTPUT:
%   output: Tx1 vector of output

output   = (((1+rf+(r-rf)*alpha).*W0-Wmin).^(-gamma)).*(r(:,ii)-rf)*W0;

end
```

SOC.m

```
function output = SOC(r,rf,W0,Wmin,alpha,gamma,ii,jj)
%This function computes the FOC values given parameters.
%   INPUT:
%        r: Txn matrix of returns
%       rf: scalar of risk-free rate
%       W0: scalar of initial wealth
%     Wmin: scalar of minimum wealth
%    alpha: nx1 vector of portfolio weights
%    gamma: scalar of relative risk aversion coefficient
%    ii,jj: scalars of asset label
```

```
%
%    OUTPUT:
%    output: Tx1 vector of output

output  = -gamma.*(((1+rf+(r-rf)*alpha).*W0-Wmin).^(-gamma-1)).*(r(:,ii)-rf).*(r(:,jj)-rf)*W0^2;

end
```

Question 3

`Q3.m`

```
%Q3
clc;clear

%% 1
mu = 0;
sigma  =0.25;
n=100;

y1 = 1.02;
Ey1 = 1.02;

[z,w] = qnwnorm(n,0,1);
y2 = @(z)exp(mu+sigma*z);
y2fval = feval(y2,z);
Ey2 = w'*y2fval;

%% 2
gamma = 1.5;

E1 = (y1^(1-gamma)-1)/(1-gamma);

f2 = @(y2)(y2.^(1-gamma)-1)/(1-gamma);
fval = feval(f2,y2fval);
E2 = w'*fval;
disp('Since E1>E2, he will choose project 1.')

%% 3
```

```
eps         = 1e-5;
del         = 1e-5;
max_it      = 1e6;
ini_Jac     = 1;
ini_val     = 0.5;
stop_crit   = [eps,del,max_it];

s = @(gamma) udiff(gamma);
gamma = Inverse_Broyden_Method(s,ini_Jac,ini_val,stop_crit);
disp(['gamma is equal to ',num2str(gamma)]);
```

udiff.m

```
function s = udiff(gamma)
% this function is for Q3

y1 = 1.02;
E1 = (y1^(1-gamma)-1)/(1-gamma);

n = 100;
[z,w] = qnwnorm(n,0,1);
y2 = @(z)exp(0.25*z);
y2fval = feval(y2,z);
f2 = @(y2)(y2.^(1-gamma)-1)/(1-gamma);
fval = feval(f2,y2fval);
E2 = w'*fval;

s = E2-E1;
```