Contents

MIE377 (Winter 2024) - Laboratory 6

The purpose of this program is to solve a CVaR optimization problem. We will formulate the problem as a linear program. We will use historical scenarios to solve this problem.

TA: David Islip Instructor: Roy H. Kwon

```
clc
clear all
format short

% Program Start
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

1. DEFINE PARAMETERS

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Load the historical weekly price data for 50 assets (210 observations)
load('lab2data.mat')

% Calculate the asset and factor returns (factor models use returns, not
% prices)
rets = ( prices(2:end,:) ./ prices(1:end-1,:) ) - 1;
facRets = sp500price( 2:end , 1 ) ./ sp500price( 1:end - 1, 1 ) - 1;

% Number of assets and number of historical scenarios
[S, n] = size( rets );

% Define the confidence level
alpha = 0.95;

% Estimate the asset exp. returns by taking the geometric mean
mu = ( geomean(rets + 1) - 1 )';

% Set our target return by taking the geometric mean of the factor returns
R = geomean(facRets + 1) - 1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

2. Construct the appropriate matrices for optimization

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% We can model CVaR Optimization as a Linear Program.
%
%   min     gamma + (1 / [(1 - alpha) * S]) * sum( z_s )
%   s.t.    z_s   >= 0,                 for s = 1, ..., S
%           z_s   >= -r_s' x - gamma,   for s = 1, ..., S
%           1' x  =  1,
%           mu' x >= R
%
% Therefore, we will use MATLAB's 'linprog' in this example. In this
% section of the code we will construct our inequality constraint matrix
% 'A' and 'b' for
%
%   A x <= b
%
% This means we need to rearrange our constraint to have all the variables
% on the LHS of the inequality.

% Define the lower and upper bounds to our portfolio
lb = [-inf(n,1); zeros(S,1); -inf];
ub = [inf(n,1); inf(S,1); inf];
```

```
% Define the inequality constraint matrices A and b
A = [-rets, -eye(S), -ones(S,1); -mu', zeros(1,S), 0];
b = [zeros(S,1); -R];

% Define the equality constraint matrices A_eq and b_eq
Aeq = [ones(1,n), zeros(1,S), 0];
beq = 1;

% Define our objective linear cost function c
c = [zeros(n,1); 1/(1-alpha)/S*ones(S,1); 1];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

3. Find the optimal portfolio

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Set the linprog options to increase the solver tolerance
options = optimoptions('linprog','TolFun',1e-9);

% Use 'linprog' to find the optimal portfolio
y = linprog( c, A, b, Aeq, beq, lb, ub, options );

% Retrieve the optimal portfolio weights
x = y(1:n);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
Optimal solution found.
```

4. Plot the results

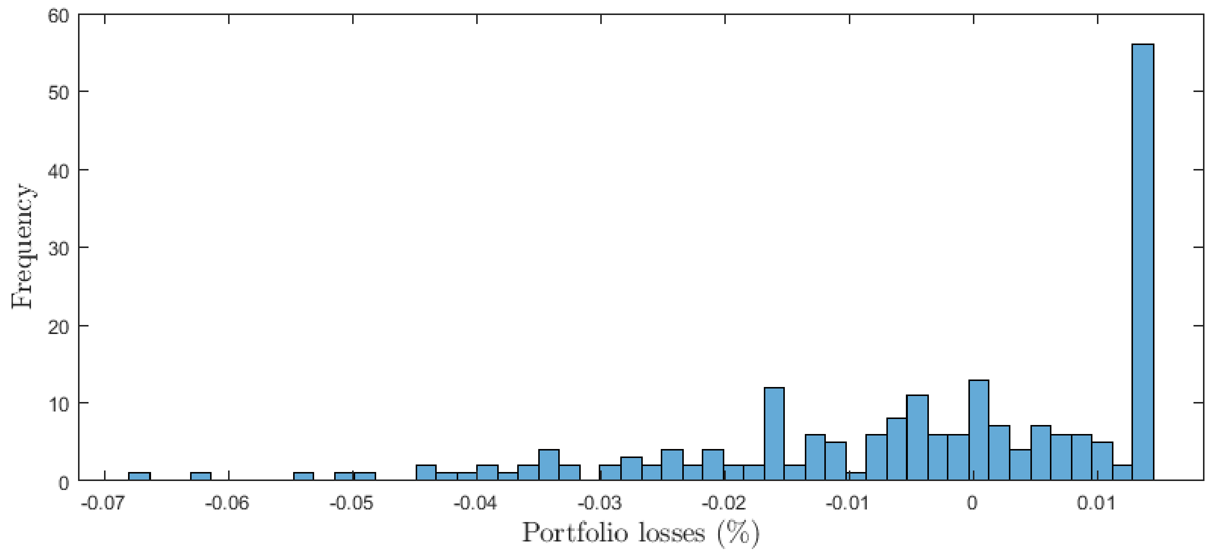
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Plot the historical loss distribution for this portfolio
loss = -rets * x;

fig1 = figure(1);
histogram(loss, 50);
xlabel('Portfolio losses ($\%$)','interpreter',...
    'latex','FontSize',14);
ylabel('Frequency','interpreter','latex','FontSize',14);

set(fig1,'Units','Inches', 'Position', [0 0 10, 4]);
pos2 = get(fig1,'Position');
set(fig1,'PaperPositionMode','Auto','PaperUnits','Inches',...
'PaperSize',[pos2(3), pos2(4)])

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Program End
```


Frequency
Portfolio losses (%)
0
10
20
30
40
50
60
-0.07
-0.06
-0.05
-0.04
-0.03
-0.02
-0.01
0
0.01