Portfolio Optimization

In this notebook, we will first solve the problem discussed in this lecture note. Then using the same data, we will solve three different optimization problems commonly encountered in Portfolio optimization.

1. Exact problem discussed in the lecture note
2. Maximizing mean return
3. Minimizing risk
4. Simultaneously maximizing mean and minimizing risk

We will use CVX library (http://cvxr.com/cvx/) to solve contrained quadratic optimization problems.

1. Problem discussed in the lecture note

The dataset used in this notebook can be found in the lecture notes of Prof. Shabbir Ahmed. The lecture note can be found at this link: https://www2.isye.gatech.edu/~sahmed/isye6669/notes/portfolio

Read the data.

```
IBM = [93.043, 84.585, 111.453, 99.525, 95.819,...
       114.708, 111.515, 113.211, 104.942, 99.827,...
       91.607, 107.937, 115.590];
WMT = [51.826, 52.823, 56.477, 49.805, 50.287,...
       51.521, 51.531, 48.664, 55.744, 47.916,...
       49.438, 51.336, 55.081];
SEHI = [1.063, 0.938, 1.000, 0.938, 1.438,...
        1.700, 2.540, 2.390, 3.120, 2.980,...
        1.900, 1.750, 1.800];

data = [IBM', WMT', SEHI'];
table(IBM', WMT', SEHI', 'VariableName', {'IBM','WMT', 'SEHI'})
```

ans = 13×3 table

	IBM	WMT	SEHI
1	93.0430	51.8260	1.0630
2	84.5850	52.8230	0.9380
3	111.4530	56.4770	1.0000
4	99.5250	49.8050	0.9380
5	95.8190	50.2870	1.4380
6	114.7080	51.5210	1.7000
7	111.5150	51.5310	2.5400
8	113.2110	48.6640	2.3900
9	104.9420	55.7440	3.1200
10	99.8270	47.9160	2.9800

⋮

```
size(data)
```

```
ans = 1×2
    13     3
```

Calculate change in stock price

Change in stock price is calculated by subtracting a given day's stock price from following day's stock price.

```
change_in_stock_price = diff(data);
array2table(change_in_stock_price, 'VariableName', {'IBM','WMT','SEHI'})
```

ans = 12×3 table

	IBM	WMT	SEHI
1	-8.4580	0.9970	-0.1250
2	26.8680	3.6540	0.0620
3	-11.9280	-6.6720	-0.0620
4	-3.7060	0.4820	0.5000
5	18.8890	1.2340	0.2620
6	-3.1930	0.0100	0.8400
7	1.6960	-2.8670	-0.1500
8	-8.2690	7.0800	0.7300
9	-5.1150	-7.8280	-0.1400
10	-8.2200	1.5220	-1.0800

⋮

Calculate rate of change of stock price

```
rate_of_return = change_in_stock_price./data(1:end-1,:);
array2table(rate_of_return, 'VariableName', {'IBM', 'WMT', 'SEHI'})
```

ans = 12×3 table

	IBM	WMT	SEHI
1	-0.0909	0.0192	-0.1176
2	0.3176	0.0692	0.0661
3	-0.1070	-0.1181	-0.0620
4	-0.0372	0.0097	0.5330
5	0.1971	0.0245	0.1822
6	-0.0278	0.0002	0.4941
7	0.0152	-0.0556	-0.0591
8	-0.0730	0.1455	0.3054

	IBM	WMT	SEHI
9	-0.0487	-0.1404	-0.0449
10	-0.0823	0.0318	-0.3624

⋮

Sample covariance matrix

```
C = cov(rate_of_return);
C
```

```
C = 3×3
    0.0186    0.0036    0.0013
    0.0036    0.0064    0.0049
    0.0013    0.0049    0.0687
```

Covariance matrix given in the paper is slightly different from this result.

Mean return for each stock

```
means = mean(rate_of_return, 1);
means
```

```
means = 1×3
    0.0260    0.0081    0.0737
```

Optimization problem of the lecture note

Notation:

C: Covariance matrix

$e = [1, 1, 1]^T$

Problem:

$$\min x^T C x$$

s.t.

$$e^T x \leq 1000$$

$$\bar{r}^T x \geq 50$$

$$x \geq 0$$

The solver solves problems of the following kind:

$$\min x^T C x + q^T x$$

s.t.

$$Gx \leq h$$

$$Ax = b$$

So we convert our constraints to a form understandable by the solver. We don't have any equality constraints. We modify all inequality constraints to less than equal to type. The matrix G takes following form.

$$\begin{pmatrix} 1 & 1 & 1 \\ -0.026 & -0.008 & -0.073 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \leq \begin{pmatrix} 1000 \\ -50 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

```
G = [ones(1,3);-means; -eye(3)];
G
```

```
G = 5×3
    1.0000    1.0000    1.0000
   -0.0260   -0.0081   -0.0737
   -1.0000         0         0
         0   -1.0000         0
         0         0   -1.0000
```

```
h = [1000; -50; zeros(3,1)];
h
```

```
h = 5×1
        1000
         -50
           0
           0
           0
```

```
n = 3;
cvx_begin quiet
    variable x_prob1(n)
    minimize (x_prob1'*C*x_prob1)
    subject to
        G*x_prob1 <= h
cvx_end
fprintf("Variance is: %f\n", cvx_optval)
```

```
Variance is: 22634.418441
```

```
disp("Solution (x) is :")
```

```
Solution (x) is :
```

```
x_prob1
```

```
x_prob1 = 3×1
  497.0455
    0.0000
  502.9545
```

Obtained variance is higher than that of the paper. This is because the covariance matrix given in the paper is slightly different that sample covariance matrix.

2. Maximizing mean return

m : mean

$e = [1, 1, 1]^T$

Problem:

$$\max x^T m$$

s.t.

$$e^T x = 1$$

$$x^T C x \leq \sigma_0^2$$

$$x \geq 0$$

We have taken $\sigma_0^2 = 130$.

```
e = ones(3,1);
cvx_begin quiet
    variable x_prob2(n)
    maximize (x_prob2'*means')
    subject to
        e'*x_prob2 == 1
        x_prob2'*C*x_prob2 <= 130
        -eye(3)*x_prob2 <= zeros(3,1)
cvx_end
fprintf("Mean return is: %f\n", cvx_optval)
```

```
Mean return is: 0.073716
```

```
disp("Solution (x) is:")
```

```
Solution (x) is:
```

```
x_prob2
```

```
x_prob2 = 3×1
    0.0000
    0.0000
    1.0000
```

3. Minimizing risk

$e = [1, 1, 1]^T$

m: mean

Problem:

$$\min x^T C x$$

s.t.

$$e^T x = 1$$

$$x^T m = m_0$$

$$x \geq 0$$

We have taken $m_0 = 0.05$. We have taken this peculiar value because solution doesn't exist for all values of m_0. For $m_0 = 1, 5, 10, 50, etc.$, no solution exists.

```
m_0 = 0.05;
cvx_begin quiet
    variable x_prob3(n)
    minimize (x_prob3'*C*x_prob3)
    subject to
        e'*x_prob3 == 1
        x_prob3'*means' == m_0
        -eye(3)*x_prob3 <= zeros(3,1)
cvx_end
fprintf("Minimum variance is: %f\n", cvx_optval)
```

```
Minimum variance is: 0.022634
```

```
disp("Solution (x) is:")
```

```
Solution (x) is:
```

```
x_prob3
```

```
x_prob3 = 3×1
    0.4970
    0.0000
    0.5030
```

4. Simultaneously maximizing mean return and minimizing risk

We have taken $\lambda = 5$. λ value can be changed to control risk and obtain different results.

$e = [1, 1, 1]^T$

Problem:

$$\max(x^T m - \lambda x^T C x)$$

s.t.

$$e^T x = 1$$

$$x \geq 0$$

```
lamda = 5;
cvx_begin quiet
    variable x_prob4(n)
    maximize (x_prob4'*means' - lamda*x_prob4'*C*x_prob4)
```

```
    subject to
        e'*x_prob4 == 1
        -eye(3)*x_prob4 <= zeros(3,1)
cvx_end
fprintf("Maximum value is: %f\n", cvx_optval)
```

```
Maximum value is: -0.012816
```

```
disp("Solution (x) is: ")
```

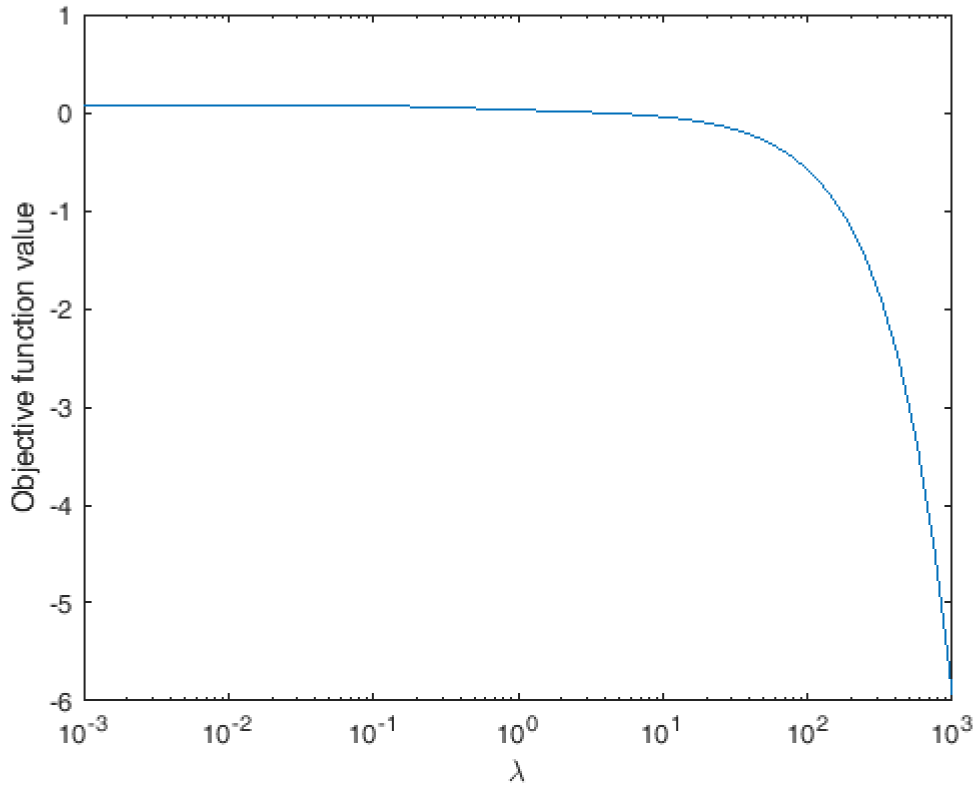
```
Solution (x) is:
```

```
x_prob4
```

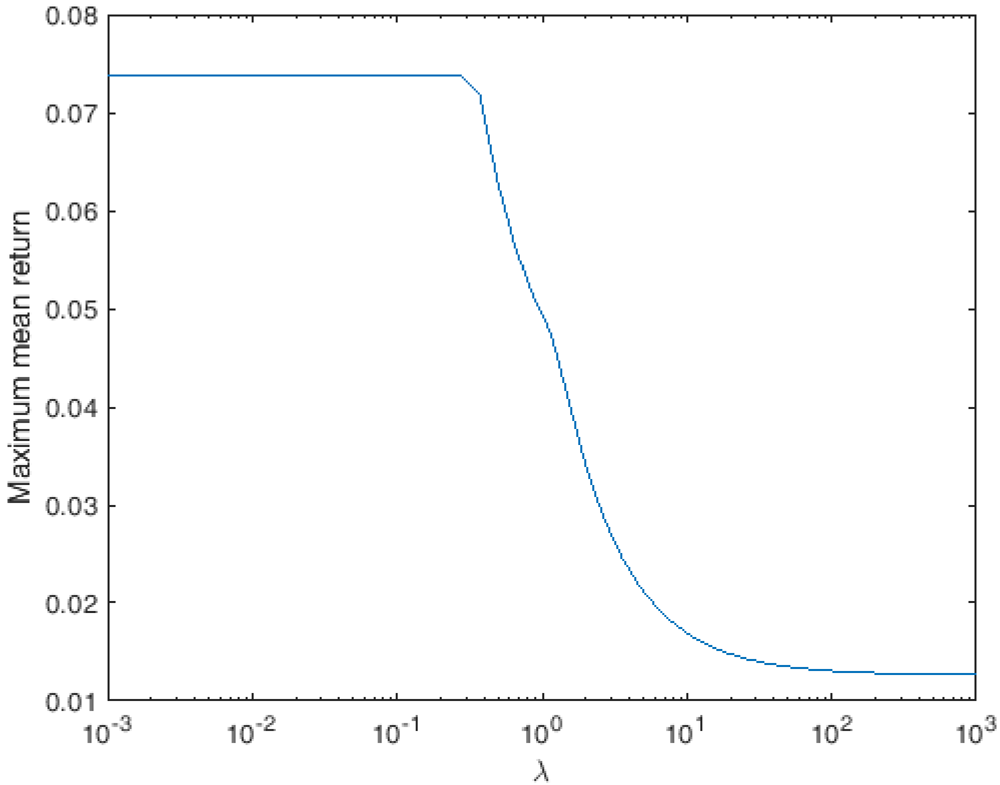
```
x_prob4 = 3×1
    0.2641
    0.6088
    0.1271
```

Solving for different values of λ

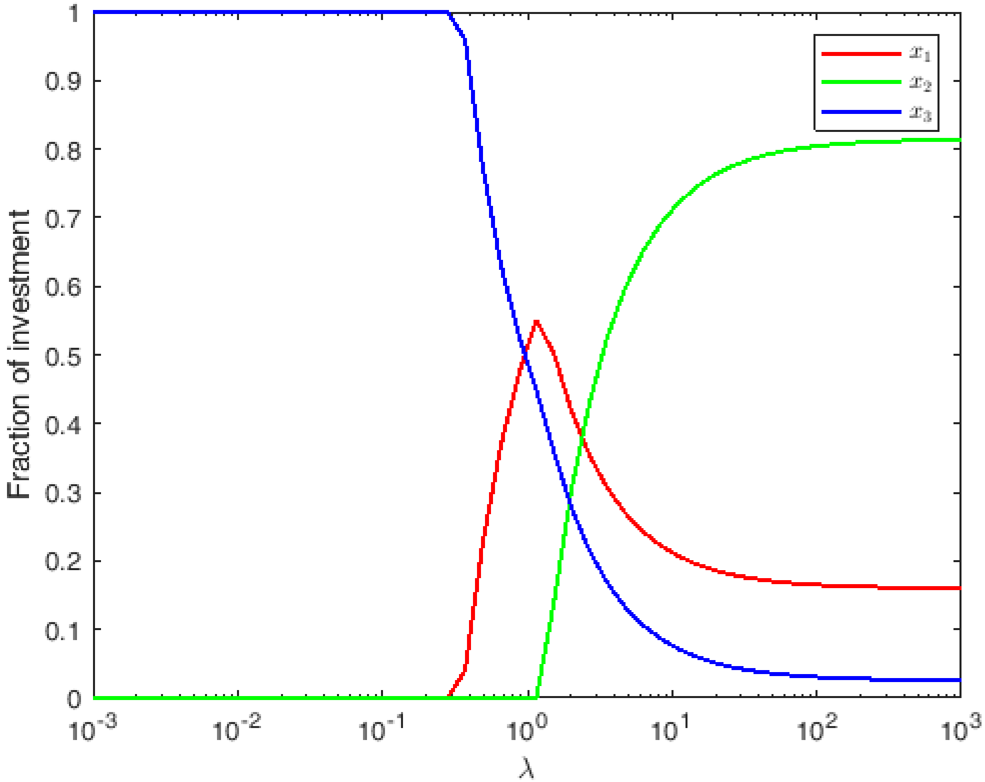
```
values = nan(50,1);
x_values = nan(50,3);
lambda = logspace(-3,3,50);
i = 1;
for lam = lambda
    cvx_begin quiet
        variable x_new(n)
        maximize (x_new'*means' - lam*x_new'*C*x_new)
        subject to
            e'*x_new == 1
            -eye(3)*x_new <= zeros(3,1)
    cvx_end
    values(i) = cvx_optval;
    x_values(i,:) = x_new;
    i = i + 1;
end
% Plot objective function value
semilogx(lambda, values)
xlabel("\lambda")
ylabel("Objective function value")
```




```
% Calculate mean return and plot it.
mean_returns = x_values*means';
figure
semilogx(lambda, mean_returns)
xlabel("\lambda")
ylabel("Maximum mean return")
```




```
% Plot of fraction of investment
figure
semilogx(lambda, x_values(:,1),'-r', 'linewidth',1.5);hold on
semilogx(lambda, x_values(:,2),'-g', 'linewidth',1.5);
semilogx(lambda, x_values(:,3),'-b', 'linewidth',1.5); hold off
xlabel("\lambda")
ylabel("Fraction of investment")
legend("$x_1$", "$x_2$","$x_3$", 'Interpreter', 'latex')
```



Fraction of investment
λ
x_1
x_2
x_3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}