Mohammadreza Arani

Convex Optimization

810100511

```
% Project 6:
```

In this project, the problem is to maximize diversification in a portfolio so that we have maximum diversity in equity at a certain portfolio. ==> Meaning we have to choose maximum number of shares in a market!

The Diversity function here is defined and given as:

$$D(x) = \frac{\sigma^T x}{(x^T \Sigma x)^{1/2}}$$

with some given constraints the problem becomes:

$$\begin{array}{ll} \text{maximize} & D(x) \\ \text{subject to} & \mathbf{1}^T x = 1, \quad 0 \preceq x \preceq M, \end{array}$$

Maximum value of a share is equal to M.

Summation of shares shall be unique value.

We always seek for ultimate score so we want to get the full credit! ==>

we solve this problem with a convex optimization format not a quasiconvex which can be solved via bi-section method!

In solving this question a feature helps the most and it is given in the hint as:

"Note also that $D(tx) = D(x)$ for any $t > 0$. "

So, the change of variable with this hint can be:

$x = \frac{y}{\tau}; \Rightarrow D(x) = D(\tau\, x) = D(y) \Rightarrow$

$\frac{(\sigma^T x)}{(x^T \Sigma\, x)^{\frac{1}{2}}} = \frac{\left(\sigma^T \frac{y}{\tau}\right)}{\frac{1}{\tau}(y^T \Sigma\, y)^{\frac{1}{2}}} \Rightarrow$ choose y and τ so that $\sigma^T y = 1$;

and the constraint $1^T x = 1$ becomes :

$\sum_i y_i = \tau \Rightarrow$

our objective will look like :

maximize $\frac{1}{(y^T \Sigma\, y)^{\frac{1}{2}}}$ which is equivalent to :

maximize $\frac{1}{(y^T \Sigma\, y\,)}$ which is equivalent to :

minimize $(y^T \Sigma\, y)$;

S.t.

$\sum_i y_i = \tau$;

$\tau > 0$;

$0 \leq y \leq \tau\, M$;

```
% As we saw, this will be a Convex Optimization problem!
% It can be solved using CVX_MATLAB!
% So we get the full credit out of this project :)
```

Part (b)

```
% Load Data:
clear; clc; close all;
load('max_divers_data.mat'); % Given M(20*1)  ,  n =20; sigma(20*1) and Sigma(20*20)
n = double(n);
M = double(M);
```

```
% Solve:

cvx_begin
    variables y(n,1) Tau(1)
        minimize(quad_form(y,Sigma))
    % Constraints:
        subject to
        sum(y)    ==     Tau         ;
        Tau       >=      0          ;
        sigma'*y  ==      1          ;
        y         >=      0          ;
```

```
        y          <=      M          ;
cvx_end
```

```
Calling SDPT3 4.0: 64 variables, 21 equality constraints
   For improved efficiency, SDPT3 is solving the dual problem.
------------------------------------------------------------

 num. of constraints = 21
 dim. of socp   var  = 22,   num. of socp blk  =  1
 dim. of linear var  = 41
 dim. of free   var  =  1 *** convert ublk to lblk
*******************************************************************
   SDPT3: Infeasible path-following algorithms
*******************************************************************
 version  predcorr  gam  expon  scale_data
    NT      1      0.000   1        0
it pstep dstep pinfeas dinfeas  gap      prim-obj      dual-obj    cputime
-------------------------------------------------------------------
 0|0.000|0.000|2.9e+01|4.1e+01|6.7e+05| 5.860071e+03  0.000000e+00| 0:0:00| chol  1  1
 1|1.000|0.973|8.8e-07|1.5e+00|3.4e+04| 5.861406e+03 -2.569144e+02| 0:0:00| chol  1  1
 2|1.000|0.970|5.6e-07|2.8e-01|6.4e+03| 2.471626e+03 -1.603800e+02| 0:0:00| chol  1  1
 3|0.951|0.994|1.2e-07|1.3e-01|6.4e+02| 1.210284e+02 -1.135202e+02| 0:0:00| chol  1  1
 4|0.985|1.000|1.1e-07|3.7e-02|2.5e+02| 5.097040e+01 -1.150978e+02| 0:0:00| chol  1  1
 5|0.941|0.923|8.3e-09|1.3e-02|2.6e+01|-9.629621e+01 -1.133925e+02| 0:0:00| chol  1  1
 6|0.862|0.881|1.5e-09|2.6e-03|9.0e+00|-1.057403e+02 -1.139525e+02| 0:0:00| chol  1  1
 7|0.969|0.661|2.2e-10|9.5e-04|6.7e-01|-1.133321e+02 -1.139322e+02| 0:0:00| chol  1  1
 8|0.933|0.617|2.3e-10|3.7e-04|1.3e-01|-1.138160e+02 -1.139269e+02| 0:0:00| chol  1  1
 9|1.000|0.335|1.1e-10|2.5e-04|5.4e-02|-1.138851e+02 -1.139256e+02| 0:0:00| chol  1  1
10|0.956|0.547|9.2e-11|1.1e-04|1.8e-02|-1.139121e+02 -1.139245e+02| 0:0:00| chol  1  1
11|1.000|0.661|1.8e-11|3.8e-05|3.5e-03|-1.139231e+02 -1.139247e+02| 0:0:00| chol  1  1
12|0.818|0.782|4.8e-12|1.5e-05|7.7e-04|-1.139249e+02 -1.139252e+02| 0:0:00| chol  1  1
13|0.977|0.879|7.2e-13|3.0e-06|6.9e-05|-1.139253e+02 -1.139254e+02| 0:0:00| chol  1  1
14|0.954|0.894|3.4e-13|2.7e-07|7.0e-06|-1.139254e+02 -1.139254e+02| 0:0:00| chol  1  1
15|0.982|0.975|2.9e-13|2.6e-08|6.9e-07|-1.139254e+02 -1.139254e+02| 0:0:00| chol  1  1
16|1.000|0.985|5.8e-13|2.6e-09|2.6e-08|-1.139254e+02 -1.139254e+02| 0:0:00|
  stop: max(relative gap, infeasibilities) < 1.49e-08
-------------------------------------------------------------------
 number of iterations   = 16
 primal objective value = -1.13925393e+02
 dual   objective value = -1.13925393e+02
 gap := trace(XZ)       = 2.55e-08
 relative gap           = 1.12e-10
 actual relative gap    = 1.01e-10
 rel. primal infeas (scaled problem)   = 5.79e-13
 rel. dual     "        "       "      = 2.56e-09
 rel. primal infeas (unscaled problem) = 0.00e+00
 rel. dual     "        "       "      = 0.00e+00
 norm(X), norm(y), norm(Z) = 1.6e+02, 5.1e-01, 1.1e+00
 norm(A), norm(b), norm(C) = 6.9e+01, 2.3e+02, 2.9e+00
 Total CPU time (secs)  = 0.20
 CPU time per iteration = 0.01
 termination code       =  0
 DIMACS: 5.8e-13  0.0e+00  3.8e-09  0.0e+00  1.0e-10  1.1e-10
-------------------------------------------------------------------

------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +0.0267303
```

Problem is SOlved!

```
disp("Optimal Value of Tau: "+num2str(Tau));
```

```
Optimal Value of Tau: 0.28542
```

```
disp("Optimal Values of y :");
```

```
Optimal Values of y :
```

```
disp(y'/Tau);
```

```
  Columns 1 through 11

    0.0260    0.0000    0.0235    0.0219    0.0000    0.0838    0.0445    0.0196    0.0176    0.2296    0.1075

  Columns 12 through 20

    0.0129    0.1483    0.0550    0.0105    0.0338    0.0990    0.0082    0.0334    0.0247
```

```
Optimal_cvx_y = cvx_optval;
disp(cvx_optval);
```

```
    0.0267
```

```
% Long-only portfolio:
cvx_begin
    variables x_long(n,1)
        minimize(quad_form(x_long,Sigma))
    % Constraints:
        subject to
        sum(x_long)    ==      1           ;
        x_long         >=      0           ;
        x_long         <=      M           ;
cvx_end
```

```
Calling SDPT3 4.0: 63 variables, 21 equality constraints
   For improved efficiency, SDPT3 is solving the dual problem.
------------------------------------------------------------

 num. of constraints = 21
 dim. of socp   var  = 22,   num. of socp blk  =  1
 dim. of linear var  = 40
 dim. of free   var  =  1 *** convert ublk to lblk
*******************************************************************
   SDPT3: Infeasible path-following algorithms
*******************************************************************
 version  predcorr  gam  expon  scale_data
    NT      1      0.000   1        0
it pstep dstep pinfeas dinfeas  gap      prim-obj      dual-obj    cputime
-------------------------------------------------------------------
 0|0.000|0.000|9.5e-01|2.1e+01|5.8e+05| 5.788165e+03  0.000000e+00| 0:0:00| chol  1  1
 1|1.000|1.000|3.4e-07|5.0e-01|1.8e+04| 5.396913e+03 -2.156547e+02| 0:0:00| chol  1  1
 2|1.000|1.000|1.3e-07|2.5e-01|1.7e+03| 6.021244e+02 -1.086685e+02| 0:0:00| chol  1  1
 3|1.000|1.000|1.4e-08|7.5e-02|1.6e+02|-8.456995e+00 -1.120715e+02| 0:0:00| chol  1  1
 4|0.962|0.934|1.5e-08|1.2e-02|1.7e+01|-9.941549e+01 -1.146531e+02| 0:0:00| chol  1  1
 5|1.000|0.058|1.8e-08|1.2e-02|9.1e+00|-1.072503e+02 -1.146429e+02| 0:0:00| chol  1  1
 6|0.635|0.835|6.3e-09|2.0e-03|5.0e+00|-1.095941e+02 -1.144073e+02| 0:0:00| chol  1  1
 7|1.000|0.415|7.4e-10|1.2e-03|2.1e+00|-1.122817e+02 -1.143158e+02| 0:0:00| chol  1  1
```

```
 8|0.960|0.448|4.7e-10|6.4e-04|8.8e-01|-1.133755e+02 -1.142116e+02| 0:0:00| chol  1  1
 9|1.000|0.413|1.7e-10|3.8e-04|4.3e-01|-1.137310e+02 -1.141351e+02| 0:0:00| chol  1  1
10|1.000|0.514|5.5e-11|1.8e-04|1.5e-01|-1.139451e+02 -1.140832e+02| 0:0:00| chol  1  1
11|1.000|0.468|2.8e-11|9.8e-05|6.9e-02|-1.139940e+02 -1.140573e+02| 0:0:00| chol  1  1
12|1.000|0.625|8.9e-12|3.7e-05|1.8e-02|-1.140233e+02 -1.140397e+02| 0:0:00| chol  1  1
13|1.000|0.743|3.3e-12|9.4e-06|3.8e-03|-1.140292e+02 -1.140325e+02| 0:0:00| chol  1  1
14|0.895|0.787|4.2e-13|1.4e-05|7.3e-04|-1.140303e+02 -1.140309e+02| 0:0:00| chol  1  1
15|0.989|0.913|6.1e-14|2.7e-06|5.5e-05|-1.140304e+02 -1.140305e+02| 0:0:00| chol  1  1
16|0.996|0.880|2.3e-14|2.1e-07|1.4e-05|-1.140305e+02 -1.140305e+02| 0:0:00| chol  1  1
17|1.000|0.977|1.6e-13|5.1e-08|8.6e-07|-1.140305e+02 -1.140305e+02| 0:0:00| chol  1  1
18|1.000|0.987|3.4e-12|3.2e-09|2.3e-08|-1.140305e+02 -1.140305e+02| 0:0:00|
  stop: max(relative gap, infeasibilities) < 1.49e-08
-------------------------------------------------------------------
 number of iterations   = 18
 primal objective value = -1.14030461e+02
 dual   objective value = -1.14030461e+02
 gap := trace(XZ)       = 2.29e-08
 relative gap           = 9.99e-11
 actual relative gap    = 8.66e-11
 rel. primal infeas (scaled problem)   = 3.37e-12
 rel. dual     "        "       "      = 3.23e-09
 rel. primal infeas (unscaled problem) = 0.00e+00
 rel. dual     "        "       "      = 0.00e+00
 norm(X), norm(y), norm(Z) = 1.6e+02, 6.3e-01, 1.1e+00
 norm(A), norm(b), norm(C) = 1.1e+01, 2.3e+02, 2.9e+00
 Total CPU time (secs)  = 0.16
 CPU time per iteration = 0.01
 termination code       =  0
 DIMACS: 3.4e-12  0.0e+00  4.8e-09  0.0e+00  8.7e-11  1.0e-10
-------------------------------------------------------------------

------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +0.131798
```

```
long_only_cvx_opt_val  = cvx_optval;
long_only_x = x_long;
disp("Optimal Value: "+num2str(cvx_optval))
```

```
Optimal Value: 0.1318
```

```


disp(long_only_x')
```

```
  Columns 1 through 11

    0.0219    0.0000    0.0042    0.0035    0.1053    0.0122    0.0137    0.0129    0.1671    0.1535    0.0412

  Columns 12 through 20

    0.0039    0.2000    0.0201    0.0000    0.2000    0.0180    0.0000    0.0069    0.0155
```

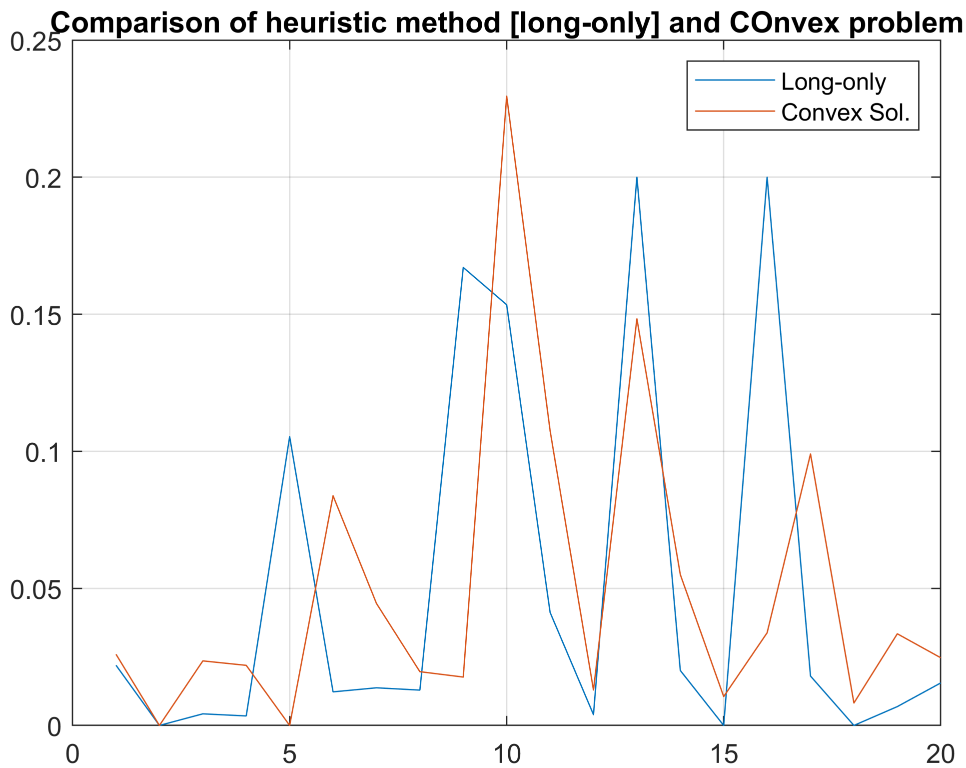
```
% we can see that Optimal Value of heuristic method is:
disp(Optimal_cvx_y/cvx_optval*100)
```

```
   20.2812
```

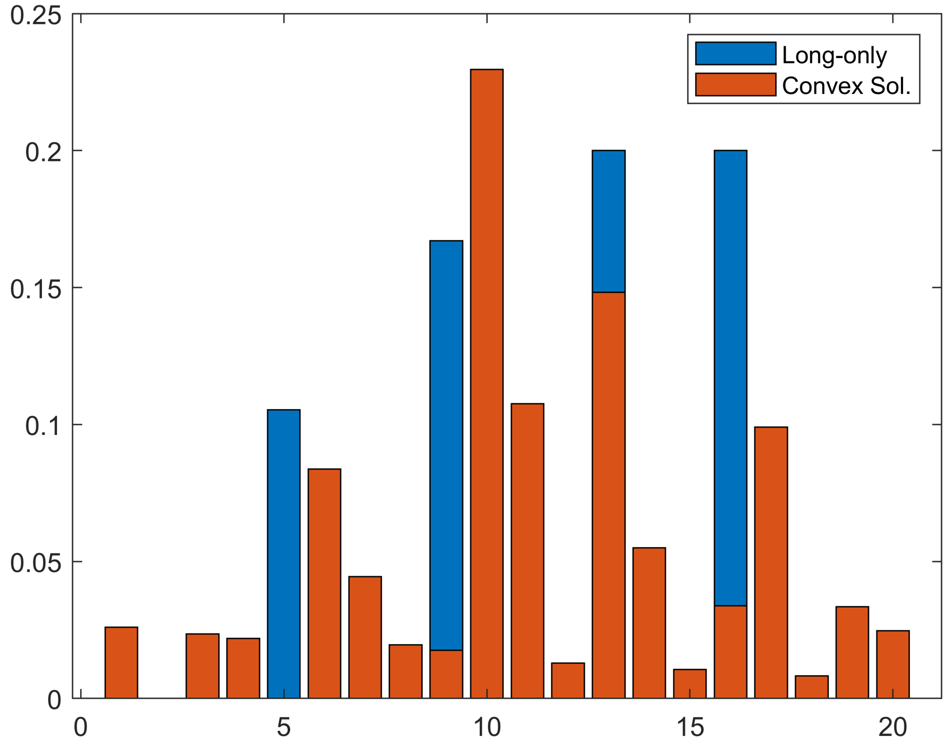
```
%percentage meaning we were 5 times better using Convex Optimization!
```

Figures

```
Fig1 = figure();
plot(x_long);
hold on
plot(y/Tau);
grid on
legend('Long-only','Convex Sol.');
title('Comparison of heuristic method [long-only] and COnvex problem');
```




```
% Using Bar plot:
Fig2 = figure();
bar(x_long);
hold on
bar(y/Tau);
legend('Long-only','Convex Sol.');
```




```
% we can see that the orange color which represents Convex solution of the
% problem is more Diverse than Long-only solution!
% Please Note that both Solutions satisfies the given conditions!
```