ELEC5470 - Convex Optimization, Fall 2018-19

Homework Set #2

Name: `Tingyuan LIANG` **Major:** `MPhil in ECE` **E-mail:** `tliang@ust.hk`

I. CONVEX OPTIMIZATION WITH MATLAB

1) **Solution to Linear Regression:**

a)

$$minimize_{a,b} \sum \phi(r_i) \tag{1}$$

subject to:

$$r_i = y_i - (ax_i + b) \tag{2}$$

where the function $\phi(\dot{})$ could be:

$$\phi_1(r) = r^2 \tag{3}$$

$$\phi_2(r) = |r| \tag{4}$$

$$\phi_3(r) = log(1 + r^2) \tag{5}$$

$r_i = y_i - (ax_i + b)$ is a linear expression
$\phi_1(r)$ and $\phi_2(r)$ are convex functions (They are actually NORM) but $\phi_3(r)$ is not, since $\phi_3''(r) = \frac{2(1-r^2)}{(1+r^2)^2}$, which is not constantly greater than 0.

Therefore, the problem is convex when the penalty function is $\phi_1(r)$ or $\phi_2(r)$. In contrast, the problem is not convex when the penalty function is $\phi_3(r)$.

b) The procedure for sub-problem (b)(c) is described by the MATLAB source code, which is shown below. According to the experimental result, to conclude, $\phi_1(r)$ is suitable for the samples without outliers and large variance. In contrast, $\phi_2(r)$ has better performance when there are outliers and large variance. Related results are shown in the Figures 1, 2 and 3 below:

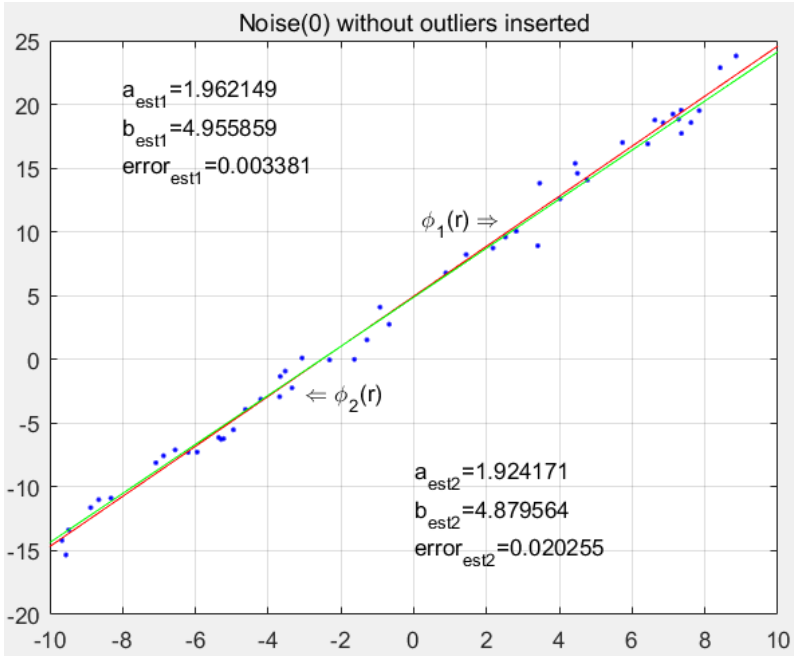


Figure 1. noise = randn(50,1)

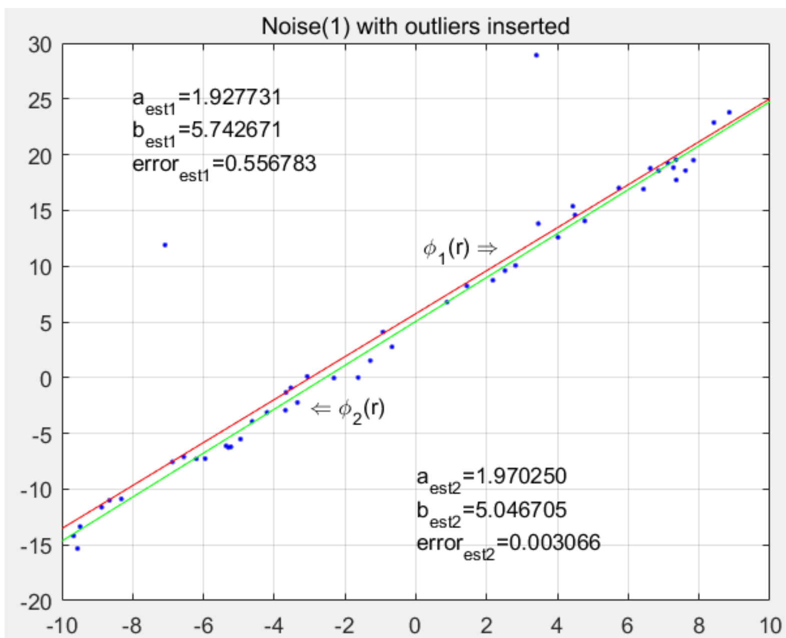


Figure 2. noise = randn(50,1) + 20*[zeros(19,1);1;zeros(19,1);1;zeros(10,1)];

.

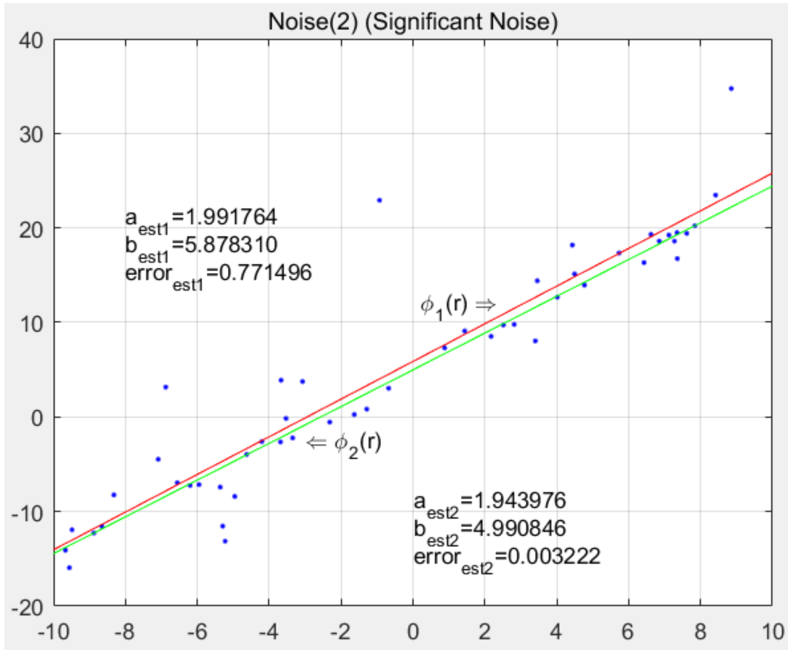


Figure 3. noise = randn(50,1) + 20*[zeros(19,1);1;zeros(19,1);1;zeros(10,1)];

```
clear all;
clf;
randn('seed',2);
rand('seed',2);
x = -10 + 20*rand(50,1);
a_true = 2; b_true = 5;
y =a_true * x + b_true;
%noise = randn(50,1) + 20*[zeros(19,1);1;zeros(19,1);1;zeros(10,1)];
%noise = randn(50,1);
noise = trnd(1,[50,1]);

y = y + noise;
plot(x,y,'b.')

cvx_begin
    variable a_est1(1)
    variable b_est1(1)
    minimize( norm( y-a_est1 * x - b_est1, 2 ) )
cvx_end
x_est1 = (-10:10);
y_est1 = a_est1 * x_est1 + b_est1;
hold on;
plot(x_est1,y_est1,'r-');
txt2 = ' \phi_1(r) \Rightarrow';
text(x_est1(14)-3,y_est1(14),txt2)
grid on


cvx_begin
    variable a_est2(1)
    variable b_est2(1)
    minimize( norm( y-a_est2 * x - b_est2, 1 ) )
cvx_end
x_est2 = (-10:10);
y_est2 = a_est2 * x_est2 + b_est2;
hold on;
plot(x_est2,y_est2,'g-');
txt1 = '\Leftarrow \phi_2(r)';
text(x_est2(7)+1,y_est2(7),txt1)
grid on

est_error1 = (a_est1-a_true)^2+(b_est1-b_true)^2;
est_error2 = (a_est2-a_true)^2+(b_est2-b_true)^2;

txt = sprintf('%s=%f','a_{est1}',a_est1);
text(-8,21,txt)
txt = sprintf('%s=%f','b_{est1}',b_est1);
text(-8,18,txt)
txt = sprintf('%s=%f','error_{est1}',est_error1);
```

```
text(-8,15,txt)
txt = sprintf('%s=%f','a_{est2}',a_est2);
text(0,-9,txt)
txt = sprintf('%s=%f','b_{est2}',b_est2);
text(0,-12,txt)
txt = sprintf('%s=%f','error_{est2}',est_error2);
text(0,-15,txt)
title('Noise(2) (Significant Noise)')
```

2) **Solution to Multiuser Transmit Beamforming:**

a) The problem (P1) is not convex due the complex vectors in SINR constraints.

To transform it into SOCP, we need to reformulate as below:

$$minimize_{\mathbf{w}_1,...,\mathbf{w}_K} \sum_{k=1}^{K} \|\mathbf{w}_k\|_2^2 \tag{6}$$

subject to:

$$\mathbf{h}_k^H \mathbf{w}_k \geq \sqrt{\gamma_k \sum_{i \neq k} \|\mathbf{h}_k^H \mathbf{w}_i\|_2^2 + \gamma_k \sigma^2} \tag{7}$$

$$Im(\mathbf{h}_k^H \mathbf{w}_k) = 0 \tag{8}$$

The reason why we can suppose $\mathbf{h}_k^H \mathbf{w}_k$ is real number is because for $\mathbf{h}_k^H \mathbf{w}_k$ having imaginary part, we can always change its phase to make it only consist of real part without changing its magnitude.

q.e.d

b) To form a convex feasibility problem, we can check whether a given set of SINR constraints can be met under the given transmit sum-power constraint P.

$$find \qquad \mathbf{w}_1, ..., \mathbf{w}_K \tag{9}$$

subject to:

$$\mathbf{h}_k^H \mathbf{w}_k \geq \sqrt{\gamma_k \sum_{i \neq k} \|\mathbf{h}_k^H \mathbf{w}_i\|_2^2 + \gamma_k \sigma^2} \tag{10}$$

$$Im(\mathbf{h}_k^H \mathbf{w}_k) = 0 \tag{11}$$

$$\sum_{k=1}^{K} \|\mathbf{w}_k\|_2^2 \leq P \tag{12}$$

q.e.d

c) **Solution to Phase Transition of Feasibility:**

The source code to solve these problem is referred to a project on GitHub:

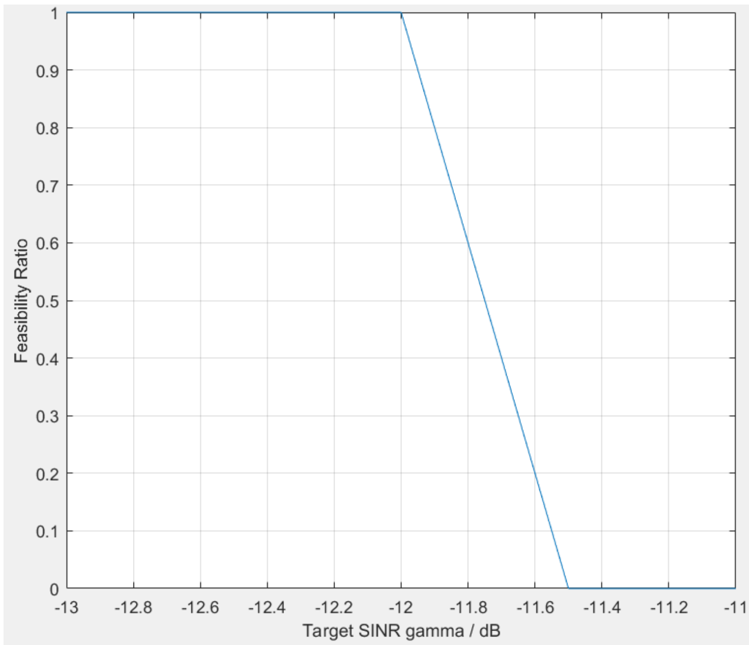
https://github.com/emilbjornson/optimal-beamforming/blob/master/functionFeasibilityProblem_cvx.m

Based on the reference, I modified it to adapt to the problems here. The modified source code is attached to this document.

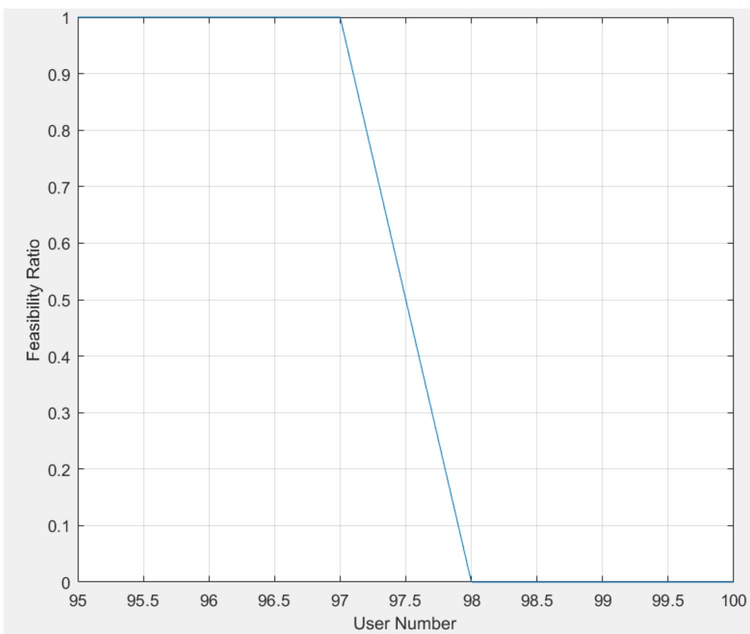
Before showing the figures and results, to conclude first, relaxed target SINR γ, less users and more antennas will make the problem more feasible, because the constraints are easier to be met.

The figures and results will be shown below, case by case.

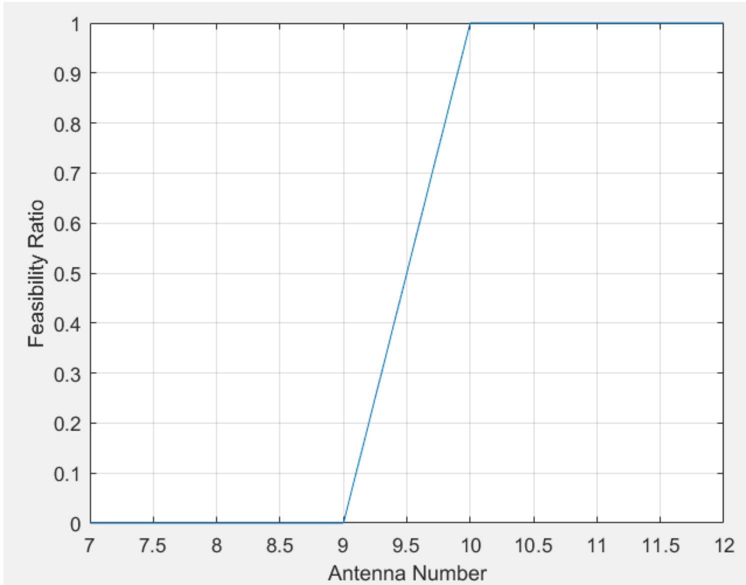
i) Fixed User and Antenna:



ii) Fixed Antenna and SINR:



iii) Fixed User and SINR:



All the evaluation is based on the following MATLAB function and different cases are actually to change the input of this function:

```
function [feasible,Wsolution] = HW_TEST2(H,gammavar)

% This is the source code for HW2 of ELEC 5470. It is refered to the
% following source code on GitHub.
% https://github.com/emilbjornson/optimal-beamforming/blob/master/functionFeasibilityProblem_cvx.m
% The related paper can be found on arXiv: https://arxiv.org/pdf/1404.0408.pdf


Kr = size(H,1); %Number of users
N = size(H,2); %Number of transmit antennas (in total)
D = repmat(eye(N),[1 1 Kr]);

%Solve the power minimization under QoS requirements problem using CVX
cvx_begin
cvx_quiet(true); % This suppresses screen output from the solver

variable W(N,Kr) complex; %Variable for N x Kr beamforming matrix
variable POWER %Scaling parameter for power constraints

minimize POWER %Minimize the power indirectly by scaling power constraints

subject to

%SINR constraints (Kr constraints)
for k = 1:Kr

    %Channels of the signal intended for user i when it reaches user k
    hkD = zeros(Kr,N);
    for i = 1:Kr
        hkD(i,:) = H(k,:)*D(:,:,i);
    end

    imag(hkD(k,:)*W(:,k)) == 0; %Useful link is assumed to be real-valued

    %SOCP formulation for the SINR constraint of user k
    real(hkD(k,:)*W(:,k)) >= sqrt(gammavar)*norm([1 hkD(k,:)*W(:,[1:k-1 k+1:Kr]) ]);
end

%Power constraints (L constraints) scaled by the variable betavar
norm(W,'fro') <= POWER;
POWER >= 0; %Power constraints must be positive

cvx_end


%Analyze result and prepare the output variables.
if isempty(strfind(cvx_status,'Solved')) %Both power minimization problem and feasibility problem are infeasible.
    feasible = false;
    Wsolution = [];
else %Both power minimization problem and feasibility problem are feasible.
    feasible = true;
```

```
    Wsolution = W;
    POWER
end
```

For each case, the procedure of input data generation is listed as below:

i) Fixed User and Antenna:

```
clear all;

K = 50; %user number
N = 3; %antenna number

gamma_dB_Seq = -13:0.5:-11
feasibility_ratio = -13:0.5:-11
gamma_index = 0

for gamma_dB=-13:0.5:-11
    gamma_dB
    gamma_index = gamma_index + 1;
    ok_num = 0;
    for test=1:20

        H = []; %initialize H matrix

        for i=1:K
            h = 1/sqrt(2*K)*mvnrnd(zeros(N,1),eye(N),1)'+1i/sqrt(2*K)*mvnrnd(zeros(N
                ,1),eye(N),1)';
            H = [H h];
        end

        H = H';

        gamma = db2mag(2*gamma_dB);

        [feasible,Wsolution] = HW_TEST2(H,gamma);
        ok_num = ok_num + feasible;
    end
    feasibility_ratio(gamma_index) = ok_num / 20.0
end
plot(gamma_dB_Seq,feasibility_ratio)
grid on
```

ii) Fixed Antenna and SINR:

```
clear all;

N = 3; %antenna number

K_seq = 95:1:100
feasibility_ratio = 95:1:100
K_index = 0
gamma_dB = -15; %SINR / dB

for K=95:1:100 %user number

    K
    K_index = K_index + 1;
    ok_num = 0;
    for test=1:20

        H = []; %initialize H matrix

        for i=1:K
            h = 1/sqrt(2*K)*mvnrnd(zeros(N,1),eye(N),1)'+1i/sqrt(2*K)*mvnrnd(zeros(N
                ,1),eye(N),1)';
            H = [H h];
        end

        H = H';

        gamma = db2mag(2*gamma_dB);

        [feasible,Wsolution] = HW_TEST2(H,gamma);
        ok_num = ok_num + feasible;
    end
    feasibility_ratio(K_index) = ok_num / 20.0
end
plot(K_seq,feasibility_ratio)
grid on
```

iii) Fixed User and SINR:

```
clear all;

K = 100; %antenna number

N_seq = 7:1:12
feasibility_ratio = 7:1:12
N_index = 0
gamma_dB = -10; %SINR / dB

for N=7:1:12 %user number

    N
    N_index = N_index + 1;
    ok_num = 0;
    for test=1:20

        H = []; %initialize H matrix

        for i=1:K
            h = 1/sqrt(2*K)*mvnrnd(zeros(N,1),eye(N),1)'+1i/sqrt(2*K)*mvnrnd(zeros(N
                ,1),eye(N),1)';
            H = [H h];
        end

        H = H';

        gamma = db2mag(2*gamma_dB);

        [feasible,Wsolution] = HW_TEST2(H,gamma);
        ok_num = ok_num + feasible;
    end
    feasibility_ratio(N_index) = ok_num / 20.0
end
plot(N_seq,feasibility_ratio)
grid on
```