Homework Assignment 1

Jacob Perricone, Ian Shaw, Weronika J. Święchowicz

9. (20pts) Computation Team Work: Consider the sensor localization problem on plane R^2 with one sensor and three anchors $a_1 = (1; 0)$, $a_2 = (-1; 0)$ and $a_3 = (0; 2)$. Suppose the Euclidean distances from the sensor to the three anchors are d_1, d_2 and d_3 respectively and known to us.Then, from the anchor and distance information, we can locate the second by finding $x \in R^2$ such that

$$\|x - a_i\|^2 = d_i^2, \ i = 1, 2, 3.$$

Do the following numerical experimentations:
Generate any sensor point in the convex hull of the three anchors, compute its distances to three anchors d_i, $i = 1, 2, 3$, respectively. Then solve the SOCP relaxation problem

$$\|x - a_i\|^2 \leq d_i^2, \ i = 1, 2, 3.$$

Did you find the correct location? What about if the sensor point was in the outside of the convex hull?

Table 1: Information Provided to the Solver for the SOCP relaxation problem with sensor located inside of the convex hull.

Elements	x-coordinate	y-coordinate
Anchor 1	1.0	0.0
Anchor 2	-1.0	0.0
Anchor 3	0.0	2.0
Sensor	0.131147960243	0.87137207825
Estimated Sensor	0.131147960236	0.87137207824

In addition, we provided the following set of distances between nodes (arranged 1, 2, and 3 as in Fig. 1) and the sensor (node 4 in Fig. 1): $1.2305, 1.4279, 1.1362$. The output of the computation was as follows; the norm of the difference between the true and estimated locations was 1.0157×10^{-11}. This clearly indicates that the estimated location is very accurate when the location of the sensor is inside of the convex hull created by the anchors.

`Matlab` code optimizing the location of a sensor inside of a convex hull.

Figure 1: Graphical representation of anchors (red nodes number 1, 2, and 3) and sensor locations (node 4) for the SOCP relaxation problem with sensor located inside of the convex hull. The black diamond indicated the estimated location of the sensor.

```
%% Inside of the Convex Hull
a = [1, -1, 0; 0, 0, 2];
lambda = rand(3,1);
lambda = lambda / sum(lambda);

% Finds the true location of the sensor.
xtrue = a * lambda;

% Determines distances between the sensor and the anchors.
d = pdist2(xtrue', a');

% Finds the estimated location of the sensor.
cvx_begin quiet
variable x(2);
minimize(0);
subject to
    for i = 1:3
        norm(x - a(:, i)) <= d(i);
    end
cvx_end
norm(x-xtrue);
```

Note that all plots were generated using the same plot. The code is attached at the end of the homework.

In addition, we provided the following set of distances between nodes (arranged 1, 2, and 3 as in Fig. 2) and the sensor (node 4 in Fig. 2): $7.0711, 8.6023, 6.7082$. The output of the computation was as follows; the norm of the difference between the true and estimated locations was 7.2351. This clearly indicates that the estimated location is is incorrect and far

Table 2: Information Provided to the Solver for the SOCP relaxation problem with sensor located outside of the convex hull.

Elements	x-coordinate	y-coordinate
Anchor 1	1.0	0.0
Anchor 2	-1.0	0.0
Anchor 3	0.0	2.0
Sensor	5.0	6.0
Estimated Sensor	0.11406717577	0.79254993959

from the actual sensor when the location of the sensor is outside of the convex hull created by the anchors.

`Matlab` code optimizing the location of a sensor inside of a convex hull.

```
%% Outside of the Convex Hull
a = [1, -1, 0; 0, 0, 2];

% Defines the true location of the sensor.
xtrue = [6; 5];

% Finds distances between the anchors and the sensor.
d = pdist2(xtrue', a');

% Finds the estimated location of the sensor.
cvx_begin quiet
variable x(2);
minimize(0);
subject to
    for i = 1:3
        norm(x - a(:, i)) <= d(i);
    end
cvx_end
norm(x-xtrue);
```

Figure 2: Graphical representation of anchors (red nodes number 1, 2, and 3) and sensor locations (node 4) for the SOCP relaxation problem with sensor located outside of the convex hull. The black diamond indicated the estimated location of the sensor.

Now try the SDP relaxation

$$(a_i;-1)(a_i;-1)^T \bullet \begin{pmatrix} I & x \\ x^T & y \end{pmatrix} = d_i^2,\ i=1,2,3;\ \begin{pmatrix} I & x \\ x^T & y \end{pmatrix} \succeq 0 \in S^3,$$

which can be written in the standard form

$$\begin{aligned} (1;0;0)(1;0;0)^T \bullet Z &= 1, \\ (0;1;0)(0;1;0)^T \bullet Z &= 1, \\ (1;1;0)(1;1;0)^T \bullet Z &= 2, \\ (a_i;-1)(a_i;-1)^T \bullet Z &= d_i^2,\ i=1,2,3, \\ Z &\succeq 0 \in S^3. \end{aligned}$$

Did you find the correct location everywhere on the plane?

Table 3: Information Provided to the Solver for the SDP relaxation problem with sensor located inside of the convex hull.

Elements	x-coordinate	y-coordinate
Anchor 1	1.0	0.0
Anchor 2	-1.0	0.0
Anchor 3	0.0	2.0
Sensor	0.152893627064573	0.807534375184778
Estimated Sensor	0.152893627064573	0.807534375184778

In addition, we provided the following set of distances between nodes (arranged 1, 2, and 3 as in Fig. 3) and the sensor (node 4 in Fig. 3): $1.1703, 1.4076, 1.2022$. The output of the computation was as follows; the norm of the difference between the true and estimated locations was 8.3267×10^{-17}. This clearly indicates that the estimated location is correct when the location of the sensor is inside of the convex hull created by the anchors.

Matlab code optimizing the location of a sensor inside of a convex hull.

```
%% SDP Relaxation Inside of the convex hull.
a = [1, -1, 0; 0, 0, 2];
lambda = rand(3,1);
lambda = lambda / sum(lambda);

% Defines the true location of the sensor.
xtrue = a * lambda;

% Finds distances between the anchors and the sensor.
d = pdist2(xtrue', a');
A= [1, 0, 1; 0, 1, 1; 0, 0, 0] ;

% Generates the graph of the anchors and the sensor.
cvx_begin sdp
variable Z(3,3) symmetric
minimize(0);
subject to
   sum(dot(A(:,1)*A(:,1)', Z)) == 1;
   sum(dot(A(:,2)*A(:,2)', Z)) == 1;
   sum(dot(A(:,3)*A(:,3)', Z)) == 2;
   Z >= 0

   for i = 1:3
        sum(dot([a(:, i); -1] * [a(:, i); -1]', Z)) == d(i)^2;
    end

cvx_end
x = [Z(1, 3); Z(2, 3)]
norm(x-xtrue)
```

Figure 3: Graphical representation of anchors (red nodes number 1, 2, and 3) and sensor locations (node 4) for the SDP relaxation problem with sensor located inside of the convex hull. The black diamond indicated the estimated location of the sensor.

Table 4: Information Provided to the Solver for the SDP relaxation problem with sensor located outside of the convex hull.

Elements	x-coordinate	y-coordinate
Anchor 1	1.0	0.0
Anchor 2	-1.0	0.0
Anchor 3	0.0	2.0
Sensor	6.0	5.0
Estimated Sensor	5.999999999999996	4.999999999999998

In addition, we provided the following set of distances between nodes (arranged 1, 2, and 3 as in Fig. 4) and the sensor (node 4 in Fig. 4): $7.0711, 8.6023, 6.7082$. The output of the computation was as follows; the norm of the difference between the true and estimated locations was 3.9720×10^{-15}. This clearly indicates that the estimated location is correct when the location of the sensor is inside of the convex hull created by the anchors.

Matlab code optimizing the location of a sensor inside of a convex hull.

```
%% SDP Relaxation Inside of the convex hull.
a = [1, -1, 0; 0, 0, 2];
lambda = rand(3,1);
lambda = lambda / sum(lambda);

% Defines the true location of the sensor.
xtrue = a * lambda;

% Finds distances between the anchors and the sensor.
d = pdist2(xtrue', a');
A= [1, 0, 1; 0, 1, 1; 0, 0, 0] ;

% Generates the graph of the anchors and the sensor.
cvx_begin sdp
variable Z(3,3) symmetric
minimize(0);
subject to
   sum(dot(A(:,1)*A(:,1)', Z)) == 1;
   sum(dot(A(:,2)*A(:,2)', Z)) == 1;
   sum(dot(A(:,3)*A(:,3)', Z)) == 2;
   Z >= 0

   for i = 1:3
        sum(dot([a(:, i); -1] * [a(:, i); -1]', Z)) == d(i)^2;
   end

cvx_end
x = [Z(1, 3); Z(2, 3)]
norm(x-xtrue)
```

Figure 4: Graphical representation of anchors (red nodes number 1, 2, and 3) and sensor locations (node 4) for the SDP relaxation problem with sensor located outside of the convex hull. The black diamond indicated the estimated location of the sensor.

`Matlab` code generating a plot of the anchor and sensor locations.

```
% Generates the graph of the anchors and the sensor.
nodes = [1 1 2 1 2 3]; edges = [2 3 3 4 4 4];
G = graph(nodes,edges);
xx=[a(1,1), a(1,2), a(1,3), xtrue(1)];
y = [a(2,1), a(2,2), a(2,3), xtrue(2)];
p = plot(G, 'XData', xx, 'YData', y);
p.Marker = 'nodes';
p.NodeColor = 'r'; p.MarkerSize = 10;
p.LineStyle = '-.'; p.EdgeColor = 'r';
hold on ;

% Plots the estimated location of the sensor.
plot(x(1), x(2), 'k-d', 'MarkerSize',10);
xlabel('x'); ylabel('y');
title('Graphical Representation of Anchors and Sensor Locations.')
```

To provide a better representation of the possible measurement error associated with finding an estimate of the sensor location, we generated a series of 100 random locations $xtrue$ inside, as well as the outside of the convex hull. We then proceeded with an estimation of the location for each one of the generated $xtrue$ vectors. That allowed for the computation of the norm of the difference between the true location of the sensor and its approximation. The following plot shows the magnitude of error for the optimization with SOCP and SDP Relaxations when xtrue is inside, as well as outside of the convex hull.

From the image (Fig. 5) generated based on estimation of a sensor location using the SDP relaxation, it should be clear that even though the error varies substantially, its magnitude is of order 10^{-16}. That is, regardless of the true location of the sensor, i.e., inside or the outside of the convex hull, our estimation is very accurate. On the other hand, the error associated

with the SOCP relaxation problem depends on the relative location of the sensor, i.e., inside or outside of the convex hull. The error for estimated sensor location is of magnitude 10^{-16} when the true location of the sensor is inside of the convex hull. That makes the found location very accurate. The error, however, increases approximately linearly as the distance between the true location of the sensor and the outside edge of the convex hull increases. The magnitude of the error is at least 10 (10 when true location is not far from the outside edge of the hull and much bigger when the distance increases) making the found location incorrect (estimated location is always inside of the convex hull).

Figure 5: Measurement error for the SOCP and SDP relaxation problems.

`Matlab` code generating error plots for the sensor location estimation.

```
%% Inside of the Convex Hull
a = [1, -1, 0; 0, 0, 2];
lambda = rand(3,1);
lambda = lambda / sum(lambda);
xtrue = a * lambda;
d = pdist2(xtrue', a');

cvx_begin quiet
variable x(2);
minimize([0.0]);
subject to
for i = 1:3
    norm(x - a(:, i)) <= d(i);
end
cvx_end
norm(x-xtrue)

%% Outside of the Convex Hull
a = [1, -1, 0; 0, 0, 2];
%lambda = rand(3,1);
%lambda = lambda / sum(lambda);
xtrue = [6; 5];
d = pdist2(xtrue', a');

cvx_begin quiet
variable x(2);
minimize(0);
subject to
for i = 1:3
    norm(x - a(:, i)) <= d(i);
end
```

```
cvx_end
norm(x-xtrue)

%% SDP Relaxation
a = [1, -1, 0; 0, 0, 2];
lambda = rand(3,1);
lambda = lambda / sum(lambda);
xtrue = a * lambda;
d = pdist2(xtrue', a');

A= [1, 0, 1; 0, 1,1; 0,0,0] ;

cvx_begin sdp
variable Z(3,3) symmetric
minimize(0);
subject to

sum(dot(A(:,1)*A(:,1)', Z)) == 1;
sum(dot(A(:,2)*A(:,2)', Z)) == 1;
sum(dot(A(:,3)*A(:,3)', Z)) == 2;

for i = 1:3
    sum(dot([a(:, i); -1] * [a(:, i); -1]', Z)) == d(i)^2;
end

Z >= 0;
cvx_end

%%
a = [1, -1, 0; 0, 0, 2];
[X_1, Y_1] = meshgrid(linspace(-1.5, 1.5,50)',linspace(-.5,2.5,50)')
norm_SPD = zeros(size(X_1));
norm_reg= zeros(size(X_1));
%%
A= [1, 0, 1; 0, 1,1; 0,0,0] ;
for j=1:size(X_1, 1)
    for l=1:size(Y_1,1)
```

```
        d = pdist2([X_1(j,l),Y_1(j,l)], a');

        cvx_begin sdp quiet
            variable Z(3,3) symmetric
            minimize(0);
            subject to

            sum(dot(A(:,1)*A(:,1)', Z)) == 1;
            sum(dot(A(:,2)*A(:,2)', Z)) == 1;
            sum(dot(A(:,3)*A(:,3)', Z)) == 2;

            for i = 1:3
                sum(dot([a(:, i); -1] * [a(:, i); -1]', Z)) == d(i)^2;
            end

            Z >= 0;

        cvx_end

        norm_SPD(j,l) = norm(Z(end,1:2) - [X_1(j,l), Y_1(j,l)]);

        cvx_begin quiet
            variable x(2);
            minimize(0.0);
            subject to
            for i = 1:3
                norm(x - a(:, i),2) <= d(i);
            end
        cvx_end

        norm_reg(j, l) = norm(x-[X_1(j,l); Y_1(j,l)],2);

    end
end

%%
```

```
figure()
subplot(2,1,1)
mesh(X_1, Y_1, norm_SPD)
title('Magntitude of Error with SDP Relaxation')
xlabel('X Data')
ylabel('Y Data')
zlabel('Error')

subplot(2,1,2)
mesh(X_1, Y_1, norm_reg)
title('Magntitude of Error with SOCP Relaxation')
xlabel('X Data')
ylabel('Y Data')
zlabel('Error')
```