CSE 203B W22 Homework 1

Due Time : 11:50pm, Wednesday Jan. 19, 2022 Submit to Gradescope
Gradescope: `https://gradescope.com/`

In this homework, we work on exercises from the textbook including midpoint convexity (2.3), Voronoi diagram (2.7, 2.9), quadratic function (2.10), general sets (2.12), cones and dual cones (2.28, 2.31, 2.32), and separation of cones (2.39). Extra assignments are given on convex sets.

Total points: 30. Exercises are graded by completion, assignments are graded by content.

I. Exercises from textbook chapter 2 (7 pts, 1pt for each problem)

2.3, 2.7, 2.9, 2.10, 2.12, 2.28, 2.31, 2.32, 2.39.

II. Assignments

II. 1. Qualification vs. enumeration of convex sets: Given

$$A = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$
$$b^T = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

we describe the convex sets as follows.
II.1.1. Convert set $\left\{x \mid Ax \leq b, x \in R^6_+\right\}$ from a qualification oriented expression to an enumeration oriented expression in the format of $\left\{U\theta \| I^T\theta = 1,\ \theta \in R^m_+\right\}$.(4pts)
Sol: Let $x = [x_1, \ldots, x_6]^T$. From

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

and $x \in R^6_+$ we obtain

$$x_i \geq 0\ \forall i \in \{1, \ldots, 6\},\ x_1 + x_4 \leq 1,\ x_2 + x_5 \leq 2,\ x_3 + x_6 \leq 1.$$

As the considered set is a convex hull, we may obtain U via the corner points of this expression. To obtain a corner point select any 6 equations of these 9 equations and solve them with inequality replaced with an equality. This results in **27** corner points that form the columns of U.

$$U = \begin{bmatrix} 0&0&0&0&1&1&1&1&0&0&1&1&0&0&1&1&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&2&2&0&0&2&2&0&2&0&2&0&0&0&0&0&0&0&0&2&2&0&2&0&0&0 \\ 0&1&0&1&0&1&0&1&0&0&0&0&0&1&0&1&0&0&0&1&0&1&0&0&0&1&0 \\ 1&1&1&1&0&0&0&0&1&1&0&0&1&1&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 2&2&0&0&2&2&0&0&2&0&2&0&0&0&0&0&0&0&2&2&0&0&2&0&0&0&0 \\ 1&0&1&0&1&0&1&0&0&0&0&0&1&0&1&0&0&0&1&0&1&0&0&0&1&0&0 \end{bmatrix}$$

II.1.2. Convert set $\{x \mid Ax = 0, x \in R^6\}$ from a qualification oriented expression to an enumeration oriented expression in the format of $\{Px \mid x \in R^6\}$. (4 pts)

Sol: The enumeration oriented expression is simply given by the null space of A. Obtain this via the QR decomposition of A^T that results in

$$A^T = QR = \begin{bmatrix} 1/\sqrt{2} & 0 & 0 \\ 0 & 1/\sqrt{2} & 0 \\ 0 & 0 & 1/\sqrt{2} \\ 1/\sqrt{2} & 0 & 0 \\ 0 & 1/\sqrt{2} & 0 \\ 0 & 0 & 1/\sqrt{2} \end{bmatrix} \cdot \begin{bmatrix} \sqrt{2} & 0 & 0 \\ 0 & \sqrt{2} & 0 \\ 0 & 0 & \sqrt{2} \end{bmatrix}.$$

The nullspace is generated by $\{Px = (I - Q \cdot Q^T) \cdot x \mid x \in R^6\}$.

II.1.3. Derive the dual cone of the set $\{x \mid Ax \leq 0, x \in R^6\}$. (2 pts)

Sol: Let $C = \{x \mid Ax \leq 0, x \in R^6\}$. The dual cone is the set

$$C^* = \{y | y^T x \geq 0, \forall x \in C\}$$

It may be shown that $C^* = \{A^T\theta | \theta \in R^3_-\}$.

II. 2. Support vector machine (SVM): Given two sets of points $C = \{x_1, \ldots, x_c\}$ and $D = \{y_1, \ldots, y_d\}$, where $x_i, y_i \in R^n$, we find a hyperplane with vector $a \in R^n$ to minimize the following objective function.

$$\begin{aligned} &\min \|a\|_2^2, \; a \in R^n \\ \text{s.t. } \; &a^T x_i \leq -1, \; i = 1, \ldots, c \\ &a^T y_i \geq 1, \; i = 1, \ldots, d \end{aligned}$$

(1) State the conditions with which the above formulation can have valid solutions. (2 pts)

Sol: The above can have valid solutions if the cones of C and D are disjoint.

Note: Points will be given even for an alternative solution that considers SVM with bias. E.g. An answer which states that a solution exists if the convex hulls of C and D are disjoint.

(2) Rewrite equations 2&3 as two simultaneous linear inequalities in matrix format. (2 pts)

Sol: It may be rewritten as

$$\begin{bmatrix} x_1^T \\ \vdots \\ x_c^T \\ -y_1^T \\ \vdots \\ -y_d^T \end{bmatrix} a \leq \begin{bmatrix} -1 \\ \vdots \\ -1 \end{bmatrix}.$$

(3) Use enumeration oriented format to express the two convex hulls of sets C and D. (2 pts)

Sol: Let $X = [x_1 \ \cdots \ x_m]$ and $Y = [y_1 \ \cdots \ y_m]$. The enumeration oriented convex hull of set C is given by

$$\{X\theta | I^T\theta = 1, \ \theta \in R_+^c\}$$

Similarly the convex hull of set D is given by

$$\{Y\theta | I^T\theta = 1, \ \theta \in R_+^d\}$$

(4) Create a numerical example with $c = 5, \ d = 4, \ n = 2$. (2 pts)

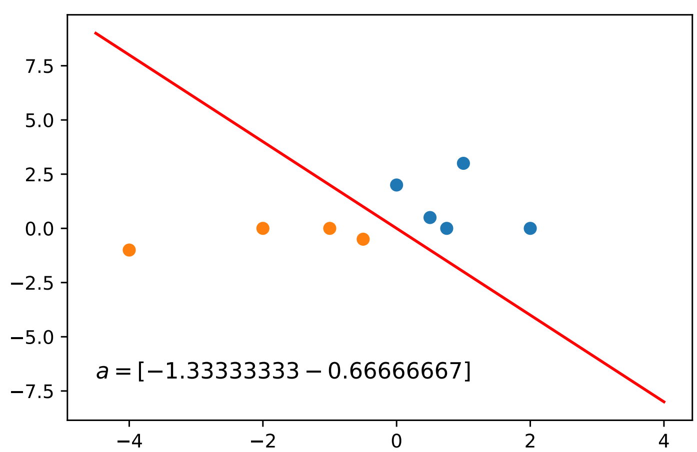
Sol: Consider

$$C = \begin{bmatrix} 0.75 & 2 & 0.5 & 0 & 1 \\ 0 & 0 & 0.5 & 2 & 3 \end{bmatrix}$$

$$D = \begin{bmatrix} -1 & -2 & -0.5 & -4 \\ 0 & 0 & -0.5 & -1 \end{bmatrix}$$

(5) Use a nonlinear programming package (e.g. Matlab) to derive the solution of item (4). Demonstrate your result with a two-dimensional plot.

Sol:




```
import numpy as np
from scipy.optimize import minimize, rosen, rosen_der

C=[[.75,0],[2,0],[0.5,0.5],[0,2],[1,3]]
D=[[-1,0],[-2,0],[-0.5,-0.5],[-4,-1]]

def func(x):
  return np.dot(x,x)

# # for inequality constraints refer
```

```
# # https://docs.scipy.org/doc/scipy-0.18.1/reference/tutorial/optimize.html
# # basically uses lambda x: func(x)  to specify func(x) >= 0
cons = []
# for c in C:
#   cons.append({'type': 'ineq', 'fun' : lambda x: -1-np.dot(x,c)})
# for d in D:
#   cons.append({'type': 'ineq', 'fun' : lambda x: -1+np.dot(x,d)})

# somehow python minimize doesn't seem to correctly append the constraint
# lambda functions in for loops, so writing it out explicitly below

cons.append({'type': 'ineq', 'fun' : lambda x: -1-np.dot(x,C[0])})
cons.append({'type': 'ineq', 'fun' : lambda x: -1-np.dot(x,C[1])})
cons.append({'type': 'ineq', 'fun' : lambda x: -1-np.dot(x,C[2])})
cons.append({'type': 'ineq', 'fun' : lambda x: -1-np.dot(x,C[3])})
cons.append({'type': 'ineq', 'fun' : lambda x: -1-np.dot(x,C[4])})
cons.append({'type': 'ineq', 'fun' : lambda x: -1+np.dot(x,D[0])})
cons.append({'type': 'ineq', 'fun' : lambda x: -1+np.dot(x,D[1])})
cons.append({'type': 'ineq', 'fun' : lambda x: -1+np.dot(x,D[2])})
cons.append({'type': 'ineq', 'fun' : lambda x: -1+np.dot(x,D[3])})

res = minimize(func, [0,0], method='SLSQP', constraints=cons)

a = res.x
print("a is {0}".format(a))

from matplotlib import pyplot as plt

fig = plt.figure()
A = C
A = np.array(A)
x, y = A.T
plt.scatter(x,y)

A = D
A = np.array(A)
x, y = A.T
plt.scatter(x,y)

x = np.linspace(-4.5,4,100)
y = (-a[0]*x)/a[1]
plt.plot(x, y, '-r')
plt.text(-4.5,-7,'$a= {}$'.format(a), fontsize=12)

plt.show()
```

```
fig.savefig('svm.pdf', dpi = 200)
```