In this homework, we work on exercises from the textbook including midpoint convexity (2.3), Voronoi diagram (2.7), 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 (8 pts, 1pt for each problem)

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

II. Assignments (42 pts)

II.1. Qualification vs. enumeration of convex sets:
Given
\[ A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}, \]
\[ b^T = [2 \ 1 \ 3], \]
we describe the convex sets as follows.
II.1.1. Convert set \( \{ x | Ax \leq b, x \in \mathbb{R}_+^4 \} \) from a qualification oriented expression to an enumeration oriented expression in the format of \( \{ U\theta | 1^T \theta = 1, \theta \in \mathbb{R}_+^m \} \). (11 pts)

[Rubrics]
- 11 pts: Correct
- 8 pts: U includes non-vertex columns / Columns of U do not include all vertices
- 5 pts: Derivation of U is wrong
- 0 pts: Missing

[Solution] The set \( \{ x | Ax \leq b, x \in \mathbb{R}_+^4 \} \) is equivalent to the set \( \{ x | \tilde{A}x \leq \tilde{b}, x \in \mathbb{R}^4 \} \), where

\[ \tilde{A} = \begin{bmatrix} A \\ -I \end{bmatrix} \in \mathbb{R}^{7 \times 4}, \tilde{b} = \begin{bmatrix} b \\ 0 \\ 0 \\ 0 \end{bmatrix} \in \mathbb{R}^7. \]

Since the set is a polyhedron in \( \mathbb{R}^4 \), we find its vertices so that we represent the set as their convex combination. In general, a polyhedron is represented by a set \( \{ U\theta | 1^T \theta = 1, \theta \in \mathbb{R}_+^m \} \), where each
column of $U$ corresponds to its vertex. A vertex is the intersection of 4 hyperplanes from $\hat{A}x = \hat{b}$, which satisfies $\hat{A}x \leq \hat{b}$. Thus, to find all vertices, we first enumerate all combinations of 4 equations from $\hat{A}x = \hat{b}$ (in total 35), whose matrix form is $Cx = d$, where $C \in \mathbb{R}^{4 \times 4}$ and $d \in \mathbb{R}^4$. Then, we solve all such $Cx = d$ when $\text{rank}(C) = 4$ and keep the solutions which satisfy $\hat{A}x \leq \hat{b}$. This results in 9 vertices. Finally, we construct the matrix $U$ whose columns are the found vertices:

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

II.1.2. Convert set $\{x|Ax = 0, x \in \mathbb{R}^4\}$ from a qualification oriented expression to an enumeration oriented expression in the format of $\{Py | y \in \mathbb{R}^m\}$. (5 pts)

[Rubrics]
- 5 pts: Correct
- 3 pts: P is wrong
- 0 pts: Missing

[Solution] The set $\{x|Ax = 0, x \in \mathbb{R}^4\}$ is the null space of $A$. By solving $Ax = 0$, we have: $x_1 = 0$, $x_2 = -x_4$, and $x_3 = 0$. Thus, we have the set in an enumeration oriented expression as follows:

$$\text{Null}(A) = \left\{ \begin{bmatrix} 0 \\ -1 \\ 0 \\ 1 \end{bmatrix} y | y \in \mathbb{R} \right\}.$$ 

[Solution 2] We want to find out the projection matrix $P$ of the null space of $A$. Then first perform QR factorization on $A^T$, which gives $Q = \begin{bmatrix}
\frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \\
0 & \frac{\sqrt{2}}{2} & 0 \\
\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\
0 & \frac{\sqrt{2}}{2} & 0
\end{bmatrix}$

Then $P = I - QQ^T = I - \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1/2 & 0 & 1/2 \\
0 & 0 & 1 & 0 \\
0 & 1/2 & 0 & 1/2
\end{bmatrix} = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 1/2 & 0 & -1/2 \\
0 & 0 & 0 & 0 \\
0 & -1/2 & 0 & 1/2
\end{bmatrix}$

II.1.3. Derive the dual cone of the set $\{x|Ax \leq 0, x \in \mathbb{R}^4\}$. (5 pts)

[Rubrics]
- 5 pts: Correct
- 3 pts: Derived dual cone is wrong
- 0 pts: Missing
[Solution] Let $K = \{x | Ax \leq 0, x \in R^4\}$. By definition, $K_1^* = \{y | x^T y \geq 0, \forall x \in K\} = \{y | x^T y \geq 0, Ax \leq 0, x \in R^4\}$ is the dual cone of $K$. By letting $y = -A^Tv$, we have $x^T y = -x^T A^Tv = -(Ax)^Tv \geq 0$ if $v \geq 0$ for all $x$ s.t. $Ax \leq 0$. We then define $K_2^* = \{-A^Tv | v \geq 0\}$ and show that $K_1^* = K_2^*$ and thus $K_2^*$ is the dual cone of $K$. The proof is as follows.

Proof. $(K_2^* \subseteq K_1^*)$ For any $v \geq 0$, it holds that for all $x$ s.t. $Ax \leq 0$, $x^T(-A^Tv) = -(Ax)^Tv \geq 0$.

$(K_1^* \subseteq K_2^*)$ For any $y$ s.t. $x^Ty \geq 0$ for all $x$ s.t. $Ax \leq 0$, it suffices to show $y = -A^Tv$ for some $v \geq 0$. We show this by contradiction. Suppose for any $v \geq 0$, $y \neq -A^Tv$. Then, either (1) $y \in \text{range}(A^T)$ and $y \neq -A^Tv$ for any $v \geq 0$, i.e., $y \in \{y | y \in \text{range}(A^T), \forall v \geq 0, y \neq -A^Tv\}$ or (2) $y \in R^4 \setminus \text{range}(A^T)$ holds.

(1) When $y \in \{y | y \in \text{range}(A^T), \forall v \geq 0, y \neq -A^Tv\}$. First notice that $\{y | y \in \text{range}(A^T), \forall v \geq 0, y \neq -A^Tv\} = \{-A^Tv | v \notin R^3\}$ since $A^T$ has full column rank and thus, the linear map defined by $A^T$ is injective. Therefore, $y = -A^Tv$ for some $v$ where there exists an index $i$ s.t. $v_i < 0$. Furthermore, there exists $x$ s.t. $(Ax)_i < 0$ and $(Ax)_j = 0$ for $j \neq i$ since rank($A$) = 3 and thus, range($A$) = $R^3$. Note that such $x$ yields $Ax \leq 0$. For such $x$, $x^Ty = -(Ax)^Tv = -(Ax)_i v_i < 0$, which is a contradiction.

(2) When $y \in R^4 \setminus \text{range}(A^T)$. It is easy to see that \text{range}(A^T) = $\{x | x_2 = x_4\}; therefore, $y \in \{x | x_2 \neq x_4\}$. We consider both cases: (a) $y_2 > y_4$ and (b) $y_2 < y_4$. In case (a), consider $x = (0, -1, 0, 1)^T$, which satisfies $Ax(=0) \leq 0$. Then, $x^Ty = -y_2 + y_4 < 0$, which is a contradiction. In case (b), consider $x = (0, 1, 0, -1)^T$, which satisfies $Ax(=0) \leq 0$. Then, $x^Ty = y_2 - y_4 < 0$, which is a contradiction. Therefore, both cases lead to a contradiction.

From the arguments above, we show that for any $y$ s.t. $x^Ty \geq 0$ for all $x$ s.t. $Ax \leq 0$, $y = -A^Tv$ for some $v \geq 0$. □

II. 2. Support vector machine (SVM): Given a set of points $\{(x_i, y_i) | i = 1, ..., m\}$, where $x_i \in R^n$, and $y_i \in \{-1, 1\}$. We find a hyperplane with vector $a \in R^n$ and bias $b \in R$ to minimize the following objective function.

\begin{align}
\min_{a,b} & |a|^2, \ a \in R^n, \ b \in R \\
\text{s.t.} & y_i(a^T x_i - b) \geq 1, \ i = 1, ..., m
\end{align}

(1) State the conditions with which the above formulation can have valid solutions. (5 pts)

[Rubrics]

• 5 pts: Correct
• 3 pts: Partial correct conditions
• 1 pts: Incorrect conditions
• 0 pts: Missing

[Solution] Valid solutions exist only if the convex hulls of the sets $\{x_i | y_i = -1, i = 1, ..., m\}$ and $\{x_i | y_i = 1, i = 1, ..., m\}$ are disjoint, and hence can be separated by a hyperplane.

[Solution] Terminology: "SVM hard margin". Valid solutions exist if and only if there exists a hyperplane such that separates the convex hull of the data points with the label -1 and the convex
hull of the data points with the label +1. That is, partition \( \{x_i \mid i = 1, \ldots, m\} = P \cup Q, P \cap Q = \emptyset \), where \( P = \{x_i \mid y_i = -1, i = 1, \ldots, m\}, Q = \{x_i \mid y_i = 1, i = 1, \ldots, m\} \). Valid solutions exist if and only if there exists a hyperplane \( H \) such that \( \text{conv}(P) \) lies on one side of \( H \), while \( \text{conv}(Q) \) lies on the other.

(2) Create a numerical example with a feasible solution. Let us set \( m = 6, n = 2 \). (5 pts)

[Rubrics]
- 5 pts: Correct
- 4.5 pts: A missing point
- 0 pts: Missing/Incorrect

[Solution] Consider the following example.
\[ C = \{((1,1), -1), ((-1, -1), -1), ((1, -2), -1), ((2, 2), 1), ((1, 3), 1), ((3, 4), 1)\} \]

(3) Create a numerical example with an infeasible solution. Let us set \( m = 6, n = 2 \). (5 pts)

[Rubrics]
- 5 pts: Correct
- 1 pts: Example has a feasible hard-margin solution
- 0 pts: Missing

[Solution] Consider the following example where the sets are not linearly separable.
\[ C = \{((1,1), -1), ((-1, 2), -1), ((1, -2), -1), ((\frac{1}{2}, \frac{1}{2}), 1), ((1, 3), 1), ((3, 4), 1)\} \]

(4) Revise the formulation so that we can derive a solution for the case created in item (3). (6 pts)
For items (2) and (4), use a nonlinear programming package (e.g. Matlab) to derive the solution. Demonstrate your result with a two-dimensional plot.

[Rubrics]
- 6 pts: Correct
- -1 pts: Revised formulation is not explicitly stated
- -1 pts: Missing hyperplane in plot/one incorrect plot
- -1 pts: Partially correct revised formulation
- -2 pts: One missing plot
- -2 pts: Incorrect revised formulation
[Solution] "SVM soft margin" One way to extend SVM to cases in which the convex hulls of the sets are not disjoint, such as in (3), is using the following formulation.

$$\min_{a, b, \lambda} \lambda ||a||^2_2 + \left[ \frac{1}{m} \sum_{i=1}^{m} \max(0, 1 - y_i (a^T x_i - b)) \right], \ a \in R^n, \ b \in R$$

(3)

where, the additional term in the objective function is the average hinge loss of the classification and the parameter $\lambda > 0$ determines the trade-off between increasing the margin size and ensuring that $x_i$ lies on the right side of the margin. The hinge loss is zero if the constraint in (2) is satisfied, in other words, if $x_i$ lies on the correct side of the hyperplane. If $x_i$ lies on the wrong side of the hyperplane, the hinge loss is proportional to the distance from the hyperplane.

Another formulation can also be,

$$\min_{a, b} \frac{1}{2} ||a||^2_2 + C \left[ \sum_{i=1}^{m} \max(0, 1 - y_i (a^T x_i - b)) \right], \ a \in R^n, \ b \in R$$

(4)

where $C$ is inversely proportional to $\lambda$ from the previous formulation. For large values of $C$, it will behave like the hard-margin SVM.

By deconstructing the hinge loss the following form of the formulation is obtained.

$$\min_{a, b, \zeta} ||a||^2_2 + C \sum_{i=1}^{n} \zeta_i$$

s.t $y_i (a^T x - b) \geq 1 - \zeta_i, \ \zeta_i \geq 0 \ \forall \ i = 1, ... m$

Listing 1: Deriving solutions in Python with scipy and sklearn

```python
from sklearn import svm
import matplotlib.pyplot as plt
import numpy as np
from scipy.optimize import minimize, rosen, rosen_der, LinearConstraint

X = np.array([[1,1], [-1,-1], [1,-2], [2,2], [1,3], [3,4]])
y = np.array([-1, -1, -1, 1, 1, 1])
xcoo, ycoo = zip(*X)

def obj(ab):
    a, b = ab[:2], ab[2]
    return np.dot(a, a)

cons = []
```
Figure 1: Plots

# The constraints here are defined in the form of A and a lower bound
# with A.dot(x) >= lower bound.
# Check documentation for further clarification
# docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.LinearConstraint.html

# y(a^Tx - b) >= 1 \Rightarrow x[0]a[0]y + x[1]a[1]y - by >= 1
# \Rightarrow [x[0]y, x[1]y, -y]^T[a[0], a[1], b] >= 1

for i in range(len(X)):
    cons.append(LinearConstraint([y[i]*X[i][0], y[i]*X[i][1], -1*y[i]], 1))
res = minimize(obj, [0, 0, 0], constraints=cons)
ab = res.x
a, b = ab[:2], ab[2:]
x_points = np.linspace(-1, 3)
y_points = -(a[0] / a[1]) * x_points + b / a[1]
plt.plot(x_points, y_points, c='r')
plt.scatter(xcoo[:3], ycoo[:3], c = 'orange', label = 'y=-1')
plt.scatter(xcoo[3:], ycoo[3:], c = 'blue', label = 'y=1')
plt.title('Hard Margin SVM for data that is linearly separable')
plt.legend()
plt.show()
y = [-1, -1, -1, 1, 1, 1]
xcoo, ycoo = zip(*X)

# sklearn.svm.LinearSVC uses the above formulation
# with hinge loss for the implementation of soft-Margin SVMs
# https://scikit-learn.org/stable/modules/svm.html#linearsvc

clf = svm.LinearSVC(loss = 'hinge')
clf.fit(X, y)
w = clf.coef_[0]
b = clf.intercept_[0]
x_points = np.linspace(-1, 3)
y_points = -(w[0] / w[1]) * x_points - b / w[1]

plt.plot(x_points, y_points, c='r')
plt.scatter(xcoo[:3], ycoo[:3], c = 'orange', label = 'y = -1')
plt.scatter(xcoo[3:], ycoo[3:], c = 'blue', label = 'y = 1')
plt.title('Soft-Margin SVM for data that is not linearly separable')
plt.legend()
plt.show()