Homework 5

2022.11.2

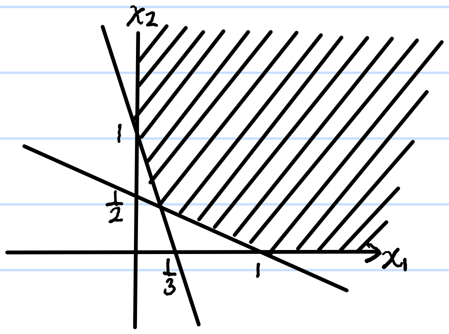
1、 Since $\hat{x}_0^T x \le \|\hat{x}_0\| \cdot \|x\| \le 1 = \hat{x}_0^T \hat{x}_0$, we have $\hat{x}_0^T (x - \hat{x}_0) \le 0$. $(x \in B)$

Since $x_0 \notin \bar{B}$, we have $\|x_0\| > 1$.

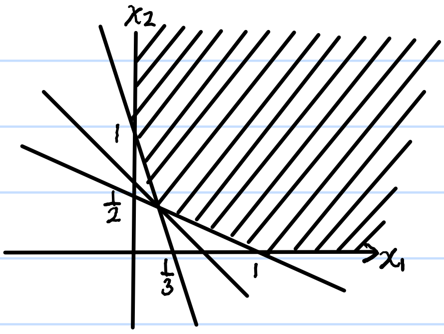
Then
$$\nabla f(\hat{x}_0)^T (x - \hat{x}_0) = (\hat{x}_0 - x_0)^T (x - \hat{x}_0)$$
$$= (1 - \|x_0\|)\, \hat{x}_0^T (x - \hat{x}_0)$$
$$\ge 0$$

By first-order optimality condition, we have $proj_{\bar{B}}(x_0) = \frac{x_0}{\|x_0\|}$.

2、 Feasible set :



(a)



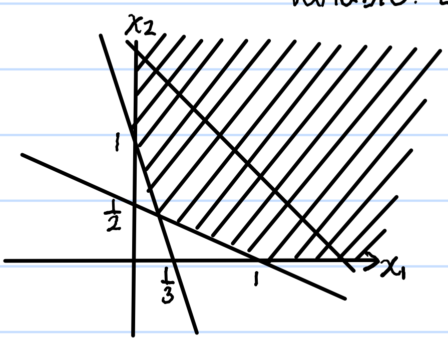
Graphically : $x^* = \begin{pmatrix} \frac{1}{5} \\ \frac{2}{5} \end{pmatrix}$, $f^* = \frac{3}{5}$

CVXPY : status : optimal

value : 0.6

variable: [0.2 0.4]

(b)



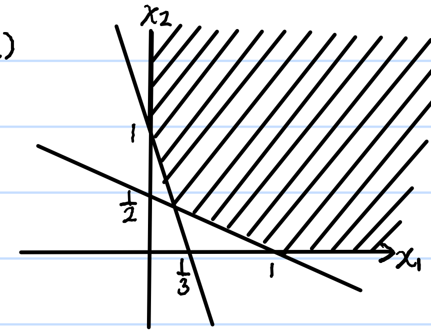
Graphically : x^* does not exist, $f^* = -\infty$

CVXPY : status : unbounded

value : -inf

variable: None

(c)



Graphically : $x^* = \begin{pmatrix} 0 \\ x_2 \end{pmatrix}$, $x_2 \geq 1$, $f^* = 0$

CVXPY : status : optimal

value : 0

variable: [0 1.767]

(d) status : optimal

value : 0.333

variable: [0.333 0.333]

Namely, $x^* = \begin{pmatrix} \frac{1}{3} \\ \frac{1}{3} \end{pmatrix}$, $f^* = \frac{1}{3}$

(e) status : optimal

value : 0.692

variable: [0.692 0.154]

Namely, $x^* \approx \begin{pmatrix} 0.692 \\ 0.154 \end{pmatrix}$, $f^* \approx 0.154$

3、(a) Introduce $t \in \mathbb{R}$, it can be reformulated as following LP:

$$\begin{aligned} \min_{x,t} \quad & t \\ \text{s.t.} \quad & -t \leq Ax - b \leq t \\ & -1 \leq x \leq 1 \end{aligned}$$

(b) status : optimal

value : 5.333

variable: [-0.333 0.333]

Namely, $x^* = (-\frac{1}{3}, \frac{1}{3})$, $f^* = \frac{16}{3}$

(c) status : optimal

value : 5.333

variable: x = [-0.333 0.333] , t = 5.333

Namely, $x^* = (-\frac{1}{3}, \frac{1}{3})$, $f^* = \frac{16}{3}$

4、(a) Normal equation is $X^T X w = X^T y$.

Then $w^* = (X^T X)^{-1} X^T y = \left(\frac{6344612}{5193237}, -\frac{1114952}{5193237}, \frac{268168}{1731079}, -\frac{2382022}{5193237}, \frac{6155944}{5193237}, \frac{10617}{1731079} \right)^T$.

Least square error $\|Xw - y\|_2^2 \approx 13.3$.

(Solved by Wolfram Mathematica)

(b) For $t=1$: status : optimal

value : 31.315

variable: [0.554 0 0 0 0.436 0.015]

It is different from (a) and sparse.

For $t=10$: status : optimal

value : 13.296

variable: [1.222 -0.215 0.155 -0.459 1.185 0.006]

It is same as (a) and not sparse.

(c) For $t=1$: status : optimal

value : 16.173

variable: [0.525 0.086 0.094 0.125 0.830 0.063]

It is different from (a) and not sparse.

For $t=100$: status : optimal

value : 13.296

variable: [1.222 -0.215 0.155 -0.459 1.185 0.006]

It is same as (a) and not sparse.

Appendix

Code & Results

2.

```
import cvxpy as cp
x1 = cp.Variable()
x2 = cp.Variable()
constraint = [
    x1 + 2 * x2 >= 1,
    3 * x1 + x2 >= 1,
    x1 >= 0,
    x2 >= 0
]
```

- (a)

```
# 2(a)
objective = cp.Minimize(x1 + x2)
problem = cp.Problem(objective, constraint)
problem.solve()
print("Solution status: {}".format(problem.status))
print("Optimal value: {}".format(problem.value))
print("Optimal variable: x = [{}, {}]".format(x1.value, x2.value))
```

> Solution status: optimal
>
> Optimal value: 0.5999999999116254
>
> Optimal variable: x = [0.1999999999391762, 0.3999999999724492]

- (b)

```
# 2(b)
objective = cp.Minimize(-x1 - x2)
problem = cp.Problem(objective, constraint)
problem.solve()
print("Solution status: {}".format(problem.status))
print("Optimal value: {}".format(problem.value))
print("Optimal variable: x = [{}, {}]".format(x1.value, x2.value))
```

> Solution status: unbounded
>
> Optimal value: -inf
>
> Optimal variable: x = [None, None]

- (c)

```python
# 2(c)
objective = cp.Minimize(x1)
problem = cp.Problem(objective, constraint)
problem.solve()
print("Solution status: {}".format(problem.status))
print("Optimal value: {}".format(problem.value))
print("Optimal variable: x = [{}, {}]".format(x1.value, x2.value))
```

> Solution status: optimal
>
> Optimal value: -1.2322148010466685e-10
>
> Optimal variable: x = [-1.2322148010466685e-10, 1.7673174212389093]

- (d)

```python
# 2(d)
objective = cp.Minimize(cp.maximum(x1, x2))
problem = cp.Problem(objective, constraint)
problem.solve()
print("Solution status: {}".format(problem.status))
print("Optimal value: {}".format(problem.value))
print("Optimal variable: x = [{}, {}]".format(x1.value, x2.value))
```

> Solution status: optimal
>
> Optimal value: 0.3333333334080862
>
> Optimal variable: x = [0.33333333286259564, 0.3333333334080862]

- (e)

```python
# 2(e)
objective = cp.Minimize((x1 ** 2) + 9 * (x2 ** 2))
problem = cp.Problem(objective, constraint)
problem.solve()
print("Solution status: {}".format(problem.status))
print("Optimal value: {}".format(problem.value))
print("Optimal variable: x = [{}, {}]".format(x1.value, x2.value))
```

> Solution status: optimal
>
> Optimal value: 0.6923076923076925
>
> Optimal variable: x = [0.6923076923076924, 0.15384615384615388]

3.

```python
import numpy as np
import cvxpy as cp
m, n = 3, 2
A = np.array([
    [2, 1],
    [1, 3],
    [1, 2]
])
b = np.array([5, 6, -5])
x = cp.Variable(n)
```

- (b)

  ```python
  # 3(b)
  constraint = [
      cp.norm_inf(x) <= 1
  ]
  objective = cp.Minimize(cp.norm_inf(A @ x - b))
  problem = cp.Problem(objective, constraint)
  problem.solve()
  print("Solution status: {}".format(problem.status))
  print("Optimal value: {}".format(problem.value))
  print("Optimal variable: x = {}".format(x.value))
  ```

  > Solution status: optimal
  >
  > Optimal value: 5.333333333553781
  >
  > Optimal variable: x = [-0.33333333 0.33333333]

- (c)

  -

  ```python
  # 3(c)
  t = cp.Variable()
  constraint = [
      A @ x - b >= -t,
      A @ x - b <= t,
      x >= -1,
      x <= 1
  ]
  objective = cp.Minimize(t)
  problem = cp.Problem(objective, constraint)
  problem.solve()
  print("Solution status: {}".format(problem.status))
  print("Optimal value: {}".format(problem.value))
  print("Optimal variable: x = {}".format(x.value))
  ```

  > Solution status: optimal
  >
  > Optimal value: 5.333333333260567
  >
  > Optimal variable: x = [-0.33333333 0.33333333]

4.

```
import numpy as np
import cvxpy as cp
m, n = 8, 6
X = np.array([
    [4, 1, 0, 4, 2, 0],
    [2, 4, 1, 1, 0, 2],
    [4, 4, 0, 4, 1, 4],
    [1, 0, 2, 3, 1, 2],
    [4, 4, 2, 2, 0, 1],
    [2, 2, 0, 1, 2, 4],
    [0, 1, 2, 1, 4, 2],
    [0, 0, 1, 0, 1, 3]
])
y = np.array([5, 0, 5, 0, 4, 2, 5, 3])
```

- (a)

```
# 4(a)
w = np.linalg.inv(X.T @ X) @ (X.T @ y)
e = np.sum(np.square(X @ w - y))
print("Least square solution: {}".format(w))
print("Least square error: {}".format(e))
```

> Least square solution: [1.22170662 -0.21469307 0.15549204 -0.4586777 1.18537706 0.00613317]
>
> Least square error: 13.295569218196665

- (b)

```
# 4(b)
for t in [1, 10]:
    print("---------- Case t = {} ----------".format(t))
    w = cp.Variable(n)
    constraint = [
        cp.norm1(w) <= t
    ]
    objective = cp.Minimize(cp.norm2(X @ w - y) ** 2)
    problem = cp.Problem(objective, constraint)
    problem.solve()
    print("Solution status: {}".format(problem.status))
    print("Optimal value: {}".format(problem.value))
    print("Optimal variable: x = {}".format(w.value))
```

> ---------- Case t = 1 ----------
>
> Solution status: optimal
>
> Optimal value: 31.314550054478023

> Optimal variable: x = [5.54241960e-01 4.31525539e-09 9.92071629e-10 9.38255329e-09 4.30602870e-01 1.51551568e-02]
>
> ---------- Case t = 10 ----------
>
> Solution status: optimal
>
> Optimal value: 13.295569218508422
>
> Optimal variable: x = [1.22171615 -0.21469843 0.15549443 -0.45868521 1.18537859 0.00613412]

- (c)

```
# 4(c)
for t in [1, 100]:
    print("---------- Case t = {} ----------".format(t))
    w = cp.Variable(n)
    constraint = [
        cp.norm2(w) ** 2 <= t
    ]
    objective = cp.Minimize(cp.norm2(X @ w - y) ** 2)
    problem = cp.Problem(objective, constraint)
    problem.solve()
    print("Solution status: {}".format(problem.status))
    print("Optimal value: {}".format(problem.value))
    print("Optimal variable: x = {}".format(w.value))
```

> ---------- Case t = 1 ----------
>
> Solution status: optimal
>
> Optimal value: 16.173131057359125
>
> Optimal variable: x = [0.52516383 0.08616926 0.09403005 0.12515129 0.82965381 0.06283205]
>
> ---------- Case t = 100 ----------
>
> Solution status: optimal
>
> Optimal value: 13.295569218196668
>
> Optimal variable: x = [1.22170662 -0.21469308 0.15549205 -0.4586777 1.18537705 0.00613318]