Homework 10

Zhen

December 6, 2021

Problem 1

The KKT conditions are:

$$
\begin{aligned}
&2x_1 + 2\mu_1(x_1 - 1) + 2\mu_2(x_1 - 1) = 0 \\
&2x_2 + 2\mu_1(x_2 - 1) + 2\mu_2 x_2 = 0 \\
&\mu_i \geq 0,\ i = 1, 2 \\
&\mu_i g_i(\boldsymbol{x}) = 0,\ i = 1, 2
\end{aligned}
$$

Case I: g_2 is inactive and g_1 is active. Then $\mu_2 = 0$ and $g_1(\boldsymbol{x}) = 0$.

$$
\begin{cases} x_1 = x_2 = 1 - \frac{\sqrt{2}}{2} \\ \mu_1 = \sqrt{2} - 1 \end{cases} \quad \text{or} \quad \begin{cases} x_1 = x_2 = 1 + \frac{\sqrt{2}}{2} \\ \mu_1 = -1 - \sqrt{2} \end{cases} \quad (\text{contradiction to } \mu_1 \geq 0)
$$

By the sufficiency of KKT conditions for convex problems, $\boldsymbol{x}^* = (1 - \frac{\sqrt{2}}{2}, 1 - \frac{\sqrt{2}}{2}),\ \mu_1 = \sqrt{2} - 1,\ \mu_2 = 0$.

Note that: this theorem(the sufficiency of KKT conditions for convex problems) will be used for many times. For the sake of convenience we denote it as "Thm 1".

Problem 2

(a) The feasible set is $\{(x_1, x_2) : x_1 = 1 \wedge x_2 = 0\}$. Then $\boldsymbol{x}^* = (1, 0), f^* = 1$.

(b) The KKT conditions are:

$$
\begin{aligned}
&2x_1 + 2\mu_1(x_1 - 1) + 2\mu_2(x_1 - 1) = 0 \\
&2x_2 + 2\mu_1(x_2 - 1) + 2\mu_2(x_2 + 1) = 0 \\
&\mu_i \geq 0,\ i = 1, 2 \\
&\mu_1[(x_1 - 1)^2 + (x_2 - 1)^2 - 1] = 0 \\
&\mu_2[(x_1 - 1)^2 + (x_2 + 1)^2 - 1] = 0
\end{aligned}
$$

$x_1 = 1$ and $x_2 = 0$ will lead to contradiction. Thus there does not exist Lagrange multipliers satisfying the KKT conditions and both g are active.

$$
\nabla \boldsymbol{g}(\boldsymbol{x}^*) = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}
$$

Thus $\boldsymbol{x}^*$ is not regular.

Problem 3

Let

$$\boldsymbol{g}(\boldsymbol{x}^*) = \begin{bmatrix} g_1(\boldsymbol{x}^*) \\ g_2(\boldsymbol{x}^*) \\ g_3(\boldsymbol{x}^*) \\ g_4(\boldsymbol{x}^*) \end{bmatrix} = \begin{bmatrix} x_1^2 - x_2 \\ x_1 + x_2 - 6 \\ -x_1 \\ -x_2 \end{bmatrix}$$

The KKT conditions are:

$$\begin{aligned} &2(x_1 - \frac{9}{4}) + 2\mu_1 x_1 + \mu_2 - \mu_3 = 0 \\ &2(x_2 - 2) - \mu_1 + \mu_2 - \mu_4 = 0 \\ &\mu_i \geq 0, \ i = 1, 2, 3, 4 \\ &\mu_i g_i(\boldsymbol{x}) = 0, \ i = 1, 2, 3, 4 \end{aligned}$$

Case I: g_1 is active and g_2, g_3, g_4 are inactive. Then $x_1^2 = x_2$ and $\mu_2 = \mu_3 = \mu_4 = 0$.
Then we have

$$x_1 = \frac{3}{2}, \ x_2 = \frac{9}{4}, \ \mu_1 = \frac{1}{2}$$

By Thm 1 and the fact that $f, \boldsymbol{g}$ are convex:

$$\boldsymbol{x}^* = \begin{bmatrix} \frac{3}{2} & \frac{9}{4} \end{bmatrix}, \ \mu_1 = \frac{1}{2}, \ \mu_2 = \mu_3 = \mu_4 = 0$$

Problem 4

(a) The KKT conditions are:

$$\begin{aligned} &\boldsymbol{x} - \boldsymbol{z} + \lambda \boldsymbol{y} - \boldsymbol{\mu} = 0 \\ &\boldsymbol{y}^T \boldsymbol{x} = 0 \\ &\boldsymbol{\mu} \geq \boldsymbol{0} \\ &\boldsymbol{M}\boldsymbol{x} = \boldsymbol{0} \end{aligned}$$

where $\lambda \in \mathbb{R}, \ \mu = (\mu_1, \ldots, \mu_n)^T \in \mathbb{R}, \ \boldsymbol{M} = \text{diag}(\mu_1, \ldots, \mu_n)$.
Then we have:

- If $x_i^* = 0$ and $\mu_i \geq 0$, then $x_i = \mu_i + z_i - \lambda y_i = 0$ which means that $z_i - \lambda y_i \leq 0$. Thus $x_i = (z_i - \lambda y_i)^+$.
- If $\mu_i = 0$, then $x_i = z_i - \lambda y_i > 0$.

Thus $x_i = (z_i - \lambda y_i)^+$.
And by the fact that $\boldsymbol{y}^T \boldsymbol{x} = 0$, $\sum_{i=1}^n y_i (z_i - \lambda y_i)^+$.

(b) My code is in the next page (Also in the file "code.py").

The final result is: $\boldsymbol{x}^* = (\frac{1}{3}, \frac{4}{3}, \frac{5}{3})^T$ and $\lambda = \frac{2}{3}$.

```python
import numpy as np

def solve(z,y):
    """
    compute lamda for a particular optimization problem:
    \min 0.5*\norm{x-z}^2
    s.t. y^Tx = 0
         x >= 0

    Inputs:
    - z: vector in objective function, of shape (N,) or (N,1)
    - y: vector in equality constraint function, of any given shape (N,) or ...
                                             (N,1)

    Returns a tuple of:
    - x: the optimal point of the problem, of shape (N,) or (N,1). x = ...
                                             (z-lamda*y)^+.
    - lamda: the Lagrange multiplier of the equality constraint function in ...
                                             KKT conditions.
    """

    shape = z.shape
    z = z.reshape(-1)
    y = y.reshape(-1)
    # initialize some parameters
    p = np.sum(y>0) # number of the elements > 0 in y
    m = len(y) - p
    u = np.sort(z[y==1]) # u = {z_i: i \in I_+}
    w = np.sort(-z[y==-1])
    u = np.append([-np.inf],u)
    u = np.append(u,[np.inf])
    w = np.append([-np.inf],w)
    w = np.append(w,[np.inf])

    U = np.zeros(p+1)
    W = np.zeros(m+1)
    for i in range(p):
        U[p-i-1] = U[p-i] + u[p-i]
    for i in range(m):
        W[i+1] = W[i] + w[i+1]
    # initialization of lamda, l and k
    lam = -1
    k = 0
    l = 0
    # Nlog(N) algortithm
    while (k<=p) and (l<=m):
        lam = (U[k] + W[l]) / (p - k + l)
        if (u[k] <= lam <= u[k+1]) and (w[l] <= lam <= w[l+1]):
            break
        if  u[k+1] < w[l+1]:
            k = k+1
        else:
            l = l+1
    return (((z - lam * y) > 0) * (z - lam * y)).reshape(shape), lam


z_p4 = np.array([1.0, 2.0, 1.0])
y_p4 = np.array([1.0, 1.0, -1.0])
x,lamda = solve(z_p4, y_p4)
# print(x,lamda)
```