EN.553.662: Optimization for Data Science
Homework 3

Ha Manh Bui (CS Department)
hbui13@jhu.edu

Spring 2023

1 Problem 1

Fix an integer n and a number c, with $0 < c < n$. Let

$$\sum_c = \left\{ x = \begin{pmatrix} x^{(1)} \\ \vdots \\ x^{(n)} \end{pmatrix} \in \mathbb{R}^n : \sum_{k=1}^{n} x^{(k)} = c, 0 \leq x^{(k)} \leq 1, k = 1, \ldots, n \right\}$$

(1) Prove that $\sum_c$ is a closed convex set.

Proof. Let $x, y \in \sum_c$ and $\lambda \in [0,1]$, we have

$$(1-\lambda)x + \lambda y = \begin{pmatrix} x^{(1)} - \lambda x^{(1)} + \lambda y^{(1)} \\ \vdots \\ x^{(n)} - \lambda x^{(n)} + \lambda y^{(n)} \end{pmatrix} \in \mathbb{R}^n. \tag{1}$$

Since $x, y \in \sum_c$, i.e., $\sum_{k=1}^{n} x^{(k)} = c$ and $\sum_{k=1}^{n} y^{(k)} = c$, then

$$\sum_{k=1}^{n} x^{(k)} - \lambda x^{(k)} + \lambda y^{(k)} = \sum_{k=1}^{n} x^{(k)} - \lambda \sum_{k=1}^{n} x^{(k)} + \lambda \sum_{k=1}^{n} y^{(k)} = c. \tag{2}$$

On the other hand, $0 \leq x^{(k)} \leq 1$, $0 \leq y^{(k)} \leq 1$, and $0 \leq \lambda \leq 1$, then

$$0 \leq x^{(k)} - \lambda x^{(k)} + \lambda y^{(k)} \leq 1. \tag{3}$$

From 1, 2, and 3, we obtain $\forall x, y \in \sum_c$, $\forall \lambda \in [0,1]$: $(1-\lambda)x + \lambda y \in \sum_c$. Therefore, $\sum_c$ is a convex set.
Let $z \in \sum_c$ is the limit point of $\sum_c$ and $z_m \in \sum_c$, we have

$$\lim_{z_m \to z} \begin{pmatrix} z_m^{(1)} \\ \vdots \\ z_m^{(n)} \end{pmatrix} = \begin{pmatrix} z^{(1)} \\ \vdots \\ z^{(n)} \end{pmatrix}.$$

As a consequence, $\sum_c$ is a closed convex set. $\square$

(2) Let $y \in \mathbb{R}^n$ be a fixed vector. Define the function $F(x) = \frac{1}{2}|x-y|^2$ so that $\underset{\sum_c}{\operatorname{argmin}} F$ is a singleton that provides the projection of y on $\sum_c$.
Justify the fact that the KKT conditions are satisfied for the minimizer of F on $\sum_c$ and prove that this minimizer $\bar{x} \in \sum_c$ is characterized by the existence of $\lambda \in \mathbb{R}$, $\alpha^{(k)} \geq 0$, $\beta^{(k)} \geq 0$ such that

$$\begin{cases} \bar{x}^{(k)} - y^{(k)} + \lambda - \alpha^{(k)} + \beta^{(k)} = 0 \\ \alpha^{(k)} \bar{x}^{(k)} = 0 \\ \beta^{(k)} (\bar{x}^{(k)} - 1) = 0 \end{cases}$$

Proof. Consider the feasible set with $\mathcal{C} = \mathcal{E} \cup \mathcal{I}$,

$$\sum_c = \left\{ x = \begin{pmatrix} x^{(1)} \\ \vdots \\ x^{(n)} \end{pmatrix} \in \mathbb{R}^n : \sum_{k=1}^{n} x^{(k)} = c, 0 \leq x^{(k)} \leq 1, k = 1, \ldots, n \right\}$$

$$= \left\{ x = \begin{pmatrix} x^{(1)} \\ \vdots \\ x^{(n)} \end{pmatrix} \in \mathbb{R}^n : \gamma_i(x) := \left\{ \sum_{k=1}^{n} x^{(k)} - c = 0 \right\}, i \in \mathcal{E}, \gamma_k(x) := \left\{ -x^{(k)} \leq 0, x^{(k)} - 1 \leq 0 \right\}, k \in \mathcal{I}, k = 1, \ldots, n \right\}.$$

Therefore, we have the set $A(x)$ of the active constraints at $x \in \sum_c$ is

$$A(x) = \left\{ i \in \mathcal{C}, \gamma_i(x) = \sum_{k=1}^{n} x^{(k)} - c = 0 \right\}.$$

Calculate the gradient of $\gamma_i(x)$ in this set, we get

$$\nabla_x \gamma_i(x) = \left(\frac{\partial \gamma_i(x)}{\partial x^{(1)}}, \ldots, \frac{\partial \gamma_i(x)}{\partial x^{(n)}} \right) = (1. \ldots, 1),$$

i.e., $(\nabla_x \gamma_i(x), i \in A(x))$ is linear independence, and as a consequence, we obtain the KKT conditions are satisfied for the minimizer $\bar{x}$ of F on $\sum_c$.
On the other hand, we have the Lagrange function:

$$\begin{aligned} L(x, \lambda) &= F(x) + \sum_{i \in \mathcal{C}} \lambda_i \gamma_i(x) \\ &= \frac{1}{2} |x - y|^2 + \lambda \left(\sum_{k=1}^{n} x^{(k)} - c \right) + \sum_{k=1}^{n} \alpha^{(k)}(-x^{(k)}) + \sum_{k=1}^{n} \beta^{(k)}(x^{(k)} - 1) \\ &= \frac{1}{2} |x|^2 - x^\top y + \frac{1}{2} |y|^2 + \lambda \left(\sum_{k=1}^{n} x^{(k)} - c \right) + \sum_{k=1}^{n} \alpha^{(k)}(-x^{(k)}) + \sum_{k=1}^{n} \beta^{(k)}(x^{(k)} - 1). \end{aligned}$$

Calculate the gradient, and we obtain

$$\nabla_x L(x, \lambda) = \left(\frac{\partial L}{\partial x^{(1)}}, \ldots, \frac{\partial L}{\partial x^{(n)}} \right) = \left(x^{(1)} - y^{(1)} + \lambda - \alpha^{(1)} + \beta^{(1)}, \ldots, x^{(n)} - y^{(n)} + \lambda - \alpha^{(n)} + \beta^{(n)} \right).$$

If $\bar{x} \in \underset{\sum_c}{\operatorname{argmin}} F$, then $\bar{x}$ satisfies the KKT conditions

$$\begin{cases} \nabla_x L(\bar{x}, \lambda) = 0 \\ \lambda_i \geq 0, i \in \mathcal{I} \\ \lambda_i = 0, i \notin A(\bar{x}) \end{cases}$$

Consider the condition $\nabla_x L(\bar{x}, \lambda) = 0$, then

$$\bar{x}^{(k)} - y^{(k)} + \lambda - \alpha^{(k)} + \beta^{(k)} = 0, k = 1, \ldots, n. \tag{4}$$

Consider the condition $\lambda_i \geq 0, i \in \mathcal{I}$, then

$$\alpha^{(k)} \geq 0, \beta^{(k)} \geq 0, k = 1, \ldots, n. \tag{5}$$

Consider the condition $\lambda_i = 0, i \notin A(\bar{x})$, i.e., $\lambda_i = 0, i \notin \left\{ i \in \mathcal{C}, \gamma_i(\bar{x}^{(k)}) = 0 \right\}$. This equivalents to the complementary slackness condition $\lambda_i \gamma_i(\bar{x}^{(k)}) = 0, i \in \mathcal{I}$, so with $k = 1, \ldots, n$, we obtain

$$\begin{cases} \alpha^{(k)} \bar{x}^{(k)} = 0 \\ \beta^{(k)}(\bar{x}^{(k)} - 1) = 0 \end{cases} \tag{6}$$

From 4, 5, and 6, we obtain the minimizer $\bar{x} \in \sum_c$ is characterized by the existence of $\lambda \in \mathbb{R}$, $\alpha^{(k)} \geq 0$, $\beta^{(k)} \geq 0$ such that

$$\begin{cases} \bar{x}^{(k)} - y^{(k)} + \lambda - \alpha^{(k)} + \beta^{(k)} = 0 \\ \alpha^{(k)} \bar{x}^{(k)} = 0 \\ \beta^{(k)} (\bar{x}^{(k)} - 1) = 0 \end{cases}$$

□

(3) Prove that these KKT conditions are true if and only if there exists $\lambda \in \mathbb{R}$ such that

$$\begin{cases} \bar{x}^{(k)} = \max(0, \min(1, y^{(k)} - \lambda)) \\ \sum_{k=1}^{n} \bar{x}^{(k)} = c \end{cases}$$

Proof. Let $J_0 = \{k : \bar{x}^{(k)} = 0\}$, $J_1 = \{k : \bar{x}^{(k)} = 1\}$, and $J = \{k : 0 < \bar{x}^{(k)} < 1\}$. From the previous result, we have the KKT conditions

$$\begin{cases} \bar{x}^{(k)} - y^{(k)} + \lambda - \alpha^{(k)} + \beta^{(k)} = 0 \quad \text{(I)} \\ \alpha^{(k)} \bar{x}^{(k)} = 0 \quad \text{(II)} \\ \beta^{(k)} (\bar{x}^{(k)} - 1) = 0 \quad \text{(III)} \end{cases} \tag{7}$$

Let's consider the two conditions $\alpha^{(k)} \bar{x}^{(k)} = 0$ (II) and $\beta^{(k)} (\bar{x}^{(k)} - 1) = 0$ (III), the solution of the equation follows

$$\text{(II)} \Rightarrow \begin{cases} \alpha^{(k)} = 0, k \in J_0 \\ \alpha^{(k)} = 0, k \in J_1 \\ \alpha^{(k)} = 0, k \in J \\ \alpha^{(k)} \neq 0, k \in J_0 \end{cases} \Rightarrow \text{(III)} \Rightarrow \begin{cases} \alpha^{(k)} = 0, k \in J_0, \beta^{(k)} = 0 \\ \alpha^{(k)} = 0, k \in J_1, \beta^{(k)} = 0 \\ \alpha^{(k)} = 0, k \in J_1, \beta^{(k)} \neq 0 \\ \alpha^{(k)} = 0, k \in J, \beta^{(k)} = 0 \\ \alpha^{(k)} \neq 0, k \in J_0, \beta^{(k)} = 0 \end{cases} \tag{8}$$

Continue to consider the condition $\bar{x}^{(k)} - y^{(k)} + \lambda - \alpha^{(k)} + \beta^{(k)} = 0$ (I), the solution of the equation follows

$$\begin{cases} \alpha^{(k)} = 0, k \in J_0, \beta^{(k)} = 0, -y^{(k)} + \lambda = 0 \\ \alpha^{(k)} = 0, k \in J_1, \beta^{(k)} = 0, 1 - y^{(k)} + \lambda = 0 \\ \alpha^{(k)} = 0, k \in J_1, \beta^{(k)} \neq 0, 1 - y^{(k)} + \lambda + \beta^{(k)} = 0 \\ \alpha^{(k)} = 0, k \in J, \beta^{(k)} = 0, \bar{x}^{(k)} - y^{(k)} + \lambda = 0 \\ \alpha^{(k)} \neq 0, k \in J_0, \beta^{(k)} = 0, -y^{(k)} + \lambda - \alpha^{(k)} = 0 \end{cases} \tag{9}$$

Therefore, there are three cases of $\bar{x}^{(k)} = \{0, y^{(k)} - \lambda, 1\}$, $k = 1, \ldots, n$ to satisfies the KKT conditions. As a consequence, these KKT conditions are true if and only if there exists $\lambda \in \mathbb{R}$ such that

$$\begin{cases} \bar{x}^{(k)} = \max(0, \min(1, y^{(k)} - \lambda)) \\ \sum_{k=1}^{n} \bar{x}^{(k)} = c \end{cases}$$

□

(4) Let $y_{\min} = \min_k y^{(k)}$ and $y_{\max} = \max_k y^{(k)}$. Prove that the function

$$\varphi : \lambda \mapsto \sum_{k=1}^{n} \max(0, \min(1, y^{(k)} - \lambda))$$

is constant on $(-\infty, y_{\min} - 1]$ and $[y_{\max}, +\infty)$ and non-increasing and piecewise linear on $[y_{\min} - 1, y_{\max}]$.

Proof. Let's consider $\lambda \in (-\infty, y_{\min} - 1]$, we have

$$\lambda \leq y_{\min} - 1 \Leftrightarrow \min_k y^{(k)} - \lambda \geq 1 \Rightarrow y^{(k)} - \lambda \geq 1, k = 1, \ldots, n \Rightarrow \varphi(\lambda) = \sum_{k=1}^{n} 1 = n.$$

So, we obtain φ is constant on $(-\infty, y_{\min} - 1]$.
Let's consider $\lambda \in [y_{\max}, +\infty)$, we have

$$\lambda \geq y_{\max} \Leftrightarrow \max_k y^{(k)} \leq \lambda \Rightarrow y^{(k)} \leq \lambda, k = 1, \ldots, n \Rightarrow \varphi(\lambda) = \sum_{k=1}^{n} 0 = 0.$$

So, we obtain φ is constant on $[y_{\max}, +\infty)$.
Let's consider $\lambda_1, \lambda_2 \in [y_{\min} - 1, y_{\max}]$, $\lambda_1 < \lambda_2$, we have

$$\begin{aligned} y^{(k)} - \lambda_1 > y^{(k)} - \lambda_2, k = 1, \ldots, n \\ \Rightarrow \sum_{k=1}^{n} \max(0, \min(1, y^{(k)} - \lambda_1)) \geq \sum_{k=1}^{n} \max(0, \min(1, y^{(k)} - \lambda_2)) \\ \Leftrightarrow \varphi(\lambda_1) \geq \varphi(\lambda_2). \end{aligned}$$

So, we obtain φ is non-increasing on $[y_{\min} - 1, y_{\max}]$.
Continue consider $\lambda_1, \lambda_2 \in [y_{\min} - 1, y_{\max}]$ and $t \in \mathbb{R}$, we have

$$\varphi(t\lambda_1 + (1-t)\lambda_2) = \sum_{k=1}^{n} \max(0, \min(1, y^{(k)} - t\lambda_1 - \lambda_2 + t\lambda_2)) \tag{10}$$

and

$$\begin{aligned} t\varphi(\lambda_1) + (1-t)\varphi(\lambda_2) &= t\sum_{k=1}^{n} \max(0, \min(1, y^{(k)} - \lambda_1)) + (1-t)\sum_{k=1}^{n} \max(0, \min(1, y^{(k)} - \lambda_2)) \\ &= \sum_{k=1}^{n} \max(0, \min(1, ty^{(k)} - t\lambda_1 + (1-t)(y^{(k)} - \lambda_2))) \\ &= \sum_{k=1}^{n} \max(0, \min(1, ty^{(k)} - t\lambda_1 + y^{(k)} - \lambda_2 - ty^{(k)} + t\lambda_2)) \\ &= \sum_{k=1}^{n} \max(0, \min(1, y^{(k)} - t\lambda_1 - \lambda_2 + t\lambda_2)). \end{aligned} \tag{11}$$

From 10 and 11, we obtain

$$\varphi(t\lambda_1 + (1-t)\lambda_2) = t\varphi(\lambda_1) + (1-t)\varphi(\lambda_2)$$

with $t \in \mathbb{R}$ and $\lambda_1, \lambda_2 \in [y_{\min} - 1, y_{\max}]$, i.e., φ is affine on $[y_{\min} - 1, y_{\max}]$. As a consequence, φ is piecewise linear on $[y_{\min} - 1, y_{\max}]$. $\square$

(5) Given c and y, one proposes the following algorithm:

1. Sort, in increasing order, the $2n$ numbers $y^{(1)}, y^{(1)} - 1, \ldots, y^{(n)}, y^{(n)} - 1$. The sorted coordinates are denoted $\hat{y}^{(k)}, k = 1, \ldots, 2n$ below.

2. Compute for all $k = 1, \ldots, 2n$ the sum

$$S_k = \sum_{l=1}^{n} \max(0, \min(1, y^{(l)} - \hat{y}^{(k)})).$$

   Note that this sum depends on both y and $\hat{y}$.

3. Determine the largest $k \geq 1$ such that $S_k \geq c$ and let

$$\lambda = \hat{y}^{(k)} + \frac{c - S_k}{S_{k+1} - S_k}(\hat{y}^{(k+1)} - \hat{y}^{(k)})$$

   Show that this algorithm always provides the value of λ introduced in Question (3).

Solution. We have

$$\lambda = \hat{y}^{(k)} + \frac{c - S_k}{S_{k+1} - S_k}(\hat{y}^{(k+1)} - \hat{y}^{(k)}) \tag{12}$$

$$\Rightarrow \frac{(S_{k+1} - S_k)(\lambda - \hat{y}^{(k)})}{\hat{y}^{(k+1)} - \hat{y}^{(k)}} + S_k = c \Leftrightarrow \frac{S_k(\hat{y}^{(k+1)} - \lambda) - S_{k+1}(\hat{y}^{(k)} - \lambda)}{\hat{y}^{(k+1)} - \hat{y}^{(k)}} = c. \tag{13}$$

Since $k \geq 1$ is the largest such that $S_k \geq c$, replace $S_k = \sum_{l=1}^{n} \max(0, \min(1, y^{(l)} - \hat{y}^{(k)}))$ into Equation 12, we have

$$\frac{\hat{y}^{(k+1)} - \lambda}{\hat{y}^{(k+1)} - \hat{y}^{(k)}} \sum_{l=1}^{n} \max(0, \min(1, y^{(l)} - \hat{y}^{(k)})) - \frac{\hat{y}^{(k)} - \lambda}{\hat{y}^{(k+1)} - \hat{y}^{(k)}} \sum_{l=1}^{n} \max(0, \min(1, y^{(l)} - \hat{y}^{(k+1)})) = c$$

$$\Leftrightarrow \sum_{l=1}^{n} \max(0, \min(1, \frac{\hat{y}^{(k+1)} - \lambda}{\hat{y}^{(k+1)} - \hat{y}^{(k)}} y^{(l)} - \hat{y}^{(k)})) - \sum_{l=1}^{n} \max(0, \min(1, \frac{\hat{y}^{(k)} - \lambda}{\hat{y}^{(k+1)} - \hat{y}^{(k)}} y^{(l)} - \hat{y}^{(k+1)})) = c$$

$$\Leftrightarrow \sum_{l=1}^{n} \max(0, \min(1, \frac{(\hat{y}^{(k+1)} - \lambda)(y^{(l)} - \hat{y}^{(k)})}{\hat{y}^{(k+1)} - \hat{y}^{(k)}} - \frac{(\hat{y}^{(k)} - \lambda)(y^{(l)} - \hat{y}^{(k+1)})}{\hat{y}^{(k+1)} - \hat{y}^{(k)}})) = c$$

$$\Leftrightarrow \sum_{l=1}^{n} \max(0, \min(1, \frac{y^{(l)}(\hat{y}^{(k+1)} - \hat{y}^{(k)}) - \lambda(\hat{y}^{(k+1)} - \hat{y}^{(k)})}{\hat{y}^{(k+1)} - \hat{y}^{(k)}})) = c \Leftrightarrow \sum_{l=1}^{n} \max(0, \min(1, y^{(l)} - \lambda)) = c.$$

This is therefore equivalent to the result of Question (3), which is $\lambda \in \mathbb{R}$ such that

$$\begin{cases} \bar{x}^{(k)} = \max(0, \min(1, y^{(k)} - \lambda)) \\ \sum_{k=1}^{n} \bar{x}^{(k)} = c \end{cases}$$

(6) Using this algorithm, program a function that takes as input the n-dimensional vector y and the constant c and returns as output the projection $\bar{x}$.

1. To test this algorithm, use the vector y provided in the file project3_vector.csv, which has dimension $n = 500$ and compute its projection $\bar{x}$ with $c = 5$. Provide in your answer the indexes k for which $\bar{x}^{(k)} > 0.001$ and the corresponding values of $\bar{x}^{(k)}$.

2. Provide the same answers with $c = 10$.

```
import numpy as np
import pandas as pd

def find_projection(y, c):
    y_hat = np.concatenate((y, y - 1), axis=0)
    y_hat = np.sort(y_hat)
    list_S = []
    for k in range(len(y_hat)):
        S_k = 0
        for l in range(len(y)):
            S_k += max(0, min(1, y[l] - y_hat[k]))
        list_S.append(S_k)
    max_k = 0
    for k in range(len(list_S) - 1):
        if list_S[k] >= c and max_k < k:
            max_k = k
    ld = y_hat[max_k] + ((c-list_S[max_k])/(list_S[max_k+1]-list_S[max_k])) * (y_hat[max_k+1]-y_hat[max_k])
    list_k, list_x_bar_k = [], []
    for k in range(len(y)):
        x_bar_k = max(0, min(1, y[k] - ld))
        if x_bar_k > 0.001:
            list_k.append(k + 1)
            list_x_bar_k.append(x_bar_k)
    return list_k, list_x_bar_k

if __name__ == "__main__":
    y = pd.read_csv('homework3_data/project3_vector-1.csv')
    y = y.drop(['Unnamed: 0'], axis=1).to_numpy()
    y = np.squeeze(y)
    list_k, list_x_bar_k = find_projection(y, 5)
    print("List of indexes k for which " + r'$\Bar{x}^{(k)}>0.001$' + " is: " + str(list_k))
    print("The corresponding values of " + r'$\Bar{x}^{(k)}$' + " is: " + str(list_x_bar_k))
```

1. Result with $c = 5$:
List of indexes k for which $\bar{x}^{(k)} > 0.001$ is: [49, 68, 206, 228, 281, 376, 452, 497, 498]
The corresponding values of $\bar{x}^{(k)}$ is: [0.0839953209484312, 0.25839430088050275, 0.10459693409061721, 1, 0.03567055250967144, 1, 1, 0.5173428915707792, 1]

```
if __name__ == "__main__":
    list_k, list_x_bar_k = find_projection(y, 10)
    print("List of indexes k for which " + r'$\Bar{x}^{(k)}>0.001$' + " is: " + str(list_k))
    print("The corresponding values of " + r'$\Bar{x}^{(k)}$' + " is: " + str(list_x_bar_k))
```

2. Result with $c = 10$:
List of indexes k for which $\bar{x}^{(k)} > 0.001$ is: [49, 50, 68, 147, 195, 206, 228, 269, 271, 281, 376, 442, 452, 462, 497, 498]
The corresponding values of $\bar{x}^{(k)}$ is: [0.7332409375537354, 0.5420354481210454, 0.907639917485807, 0.028254119021628554, 0.46771472880659104, 0.7538425506959214, 1, 0.2813180980938137, 0.0546738075137867, 0.6849161691149757, 1, 0.3267186965769273, 1, 0.21964552701576956, 1, 1]

2 Problem 2

We consider the following very special case of positive matrix factorization. Let X be an $n \times m$ matrix. Define, over $\mathbb{R}^n \times \mathbb{R}^m$, the function

$$F(y, z) = \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{m} \left(x_{ik} - y^{(i)} z^{(k)} \right)^2 .$$

Let $\Omega = \Omega_1 \times \Omega_2 \subset \mathbb{R}^n \times \mathbb{R}^m$, where Ω_1 is the set of all y such that $y^{(i)} \geq 0, i = 1, \ldots, n$ and $y^{(1)} + \cdots + y^{(n)} = 1$ and Ω_2 is the set of all z such that $z^{(k)} \geq 0, k = 1, \ldots, m$. We want to compute $\arg\min_{\Omega} F$.

(1) Assume that $y \in \Omega$ is fixed and let $F_y(z) = F(y, z)$.
(a) Write the KKT conditions for the minimization of F_y over Ω_2.
Solution. Let's $z^*(y) \in \arg\min_{z\in\Omega_2} F_y(z)$, $\mathcal{C} = \mathcal{E} \cup \mathcal{I}$ is the finite set of indices, we have the feasible set

$$\Omega_2 = \left\{ z \in \mathbb{R}^m, \gamma_k(z) := \left\{ -z^{(k)} \leq 0 \right\}, k \in \mathcal{I}, k = 1, \ldots, m \right\}.$$

Since $\mathcal{E} = \emptyset$, then the active constraints $A(z) = \emptyset$ and for $\lambda_i \in \mathbb{R}, i \in \mathcal{C}$, we have the Lagrange multiplier

$$\begin{aligned} L(z, \lambda) &= F_y(z) + \sum_{i \in \mathcal{C}} \lambda_i \gamma_i(x) \\ &= \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{m} \left(x_{ik} - y^{(i)} z^{(k)} \right)^2 - \sum_{k=1}^{m} \lambda_k z^{(k)} \\ &= \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{m} x_{ik}^2 - 2 x_{ik} y^{(i)} z^{(k)} + \left(y^{(i)} z^{(k)} \right)^2 - \sum_{k=1}^{m} \lambda_k z^{(k)}. \end{aligned}$$

Calculate the gradient, and we get

$$\nabla_z L(z, \lambda) = \left(\frac{\partial L}{\partial z^{(1)}}, \ldots, \frac{\partial L}{\partial z^{(m)}} \right) = \left(-\lambda_1 + \sum_{i=1}^{n} (y^{(i)})^2 z^{(1)} - x_{i1} y^{(i)}, \ldots, -\lambda_m + \sum_{i=1}^{n} (y^{(i)})^2 z^{(m)} - x_{im} y^{(i)} \right).$$

Therefore, we obtain the KKT conditions for $z^*(y)$ equivalents

$$\begin{cases} \nabla_z L(z^*(y), \lambda) = 0 \\ \lambda_i \geq 0, i \in \mathcal{I} \\ \lambda_i = 0, i \notin A(z^*(y)) \end{cases} \Leftrightarrow \begin{cases} \nabla_z L(z^*(y), \lambda) = 0 \\ \lambda_i \geq 0, i \in \mathcal{I} \\ \lambda_i \gamma_i(z^*(y)) = 0, i \in \mathcal{I} \end{cases} \Leftrightarrow \begin{cases} -\lambda_k + \sum_{i=1}^{n} (y^{(i)})^2 z^*(y)^{(k)} - x_{ik} y^{(i)} = 0 \quad \text{(I)} \\ \lambda_k \geq 0 \quad \text{(II)} \\ \lambda_k z^*(y)^{(k)} = 0 \quad \text{(III)} \end{cases}$$

where $k = 1, \ldots, m$.
(b) Use these conditions to prove that the optimal z is such that

$$z^{(k)} = \max \left(0, \frac{\sum_{i=1}^{n} x_{ik} y_i}{|y|^2} \right).$$

Denote this solution by $z^*(y)$.

Proof. From condition (I), we have

$$\begin{aligned}
&-\lambda_k + \sum_{i=1}^{n} (y^{(i)})^2 z^*(y)^{(k)} - x_{ik} y^{(i)} = 0 \\
\Leftrightarrow &\sum_{i=1}^{n} (y^{(i)})^2 z^*(y)^{(k)} - x_{ik} y^{(i)} = \lambda_k \\
\Leftrightarrow &z^*(y)^{(k)} - \frac{\sum_{i=1}^{n} x_{ik} y^{(i)}}{\sum_{i=1}^{n} (y^{(i)})^2} = \frac{\lambda_k}{\sum_{i=1}^{n} (y^{(i)})^2} \\
\Leftrightarrow &z^*(y)^{(k)} \left(z^*(y)^{(k)} - \frac{\sum_{i=1}^{n} x_{ik} y^{(i)}}{|y|^2} \right) = \frac{\lambda_k z^*(y)^{(k)}}{|y|^2}.
\end{aligned}$$

From the condition (II) and (III), this equivalent to

$$z^*(y)^{(k)} \left(z^*(y)^{(k)} - \frac{\sum_{i=1}^{n} x_{ik} y^{(i)}}{|y|^2} \right) = 0.$$

The solution for this equation is $z^*(y)^{(k)} = 0$ or $z^*(y)^{(k)} = \frac{\sum_{i=1}^{n} x_{ik} y^{(i)}}{|y|^2}$. As a consequence, since $z^*(y) \in \arg\min_{z \in \Omega_2} F_y(z)$, we obtain the optimal z is such that

$$z^{(k)} = \max\left(0, \frac{\sum_{i=1}^{n} x_{ik} y_i}{|y|^2}\right).$$

$\square$

(2) For fixed $z \in \Omega_2$, let $F_y(z) = F(y, z)$. Program a projected gradient descent algorithm that takes as input $x, z, \eta_0 \in \Omega_1$, a scalar $\alpha > 0$ and an integer T and iterates

$$\eta_{t+1} = \mathrm{proj}_{\Omega_1}(y_t - \alpha \nabla F_z(\eta_t))$$

until η_T is computed. You can use the program written in Question 1, noting that Ω_1 in this question is precisely what was denoted $\sum_1$ in Question 1.
Run this program using the matrix x provided in project3_matrix.csv, which has size $(n = 100) \times (m = 1000)$, and with $\alpha = 10^{-5}, T = 50, \eta_0^{(k)} = 1/n$ for all k and

$$z = n^2 X^\top \eta_0$$

Provide, in your answer the indexes k for which $\eta_T^{(k)} > 0.001$ and the corresponding values of $\eta_T^{(k)}$.

```
import numpy as np
import pandas as pd
import torch

def find_projection(y, c):
    y_hat = np.concatenate((y, y - 1), axis=0)
    y_hat = np.sort(y_hat)
    list_S = []
    for k in range(len(y_hat)):
        S_k = 0
        for l in range(len(y)):
            S_k += max(0, min(1, y[l] - y_hat[k]))
        list_S.append(S_k)
    max_k = 0
    for k in range(len(list_S) - 1):
        if list_S[k] >= c and max_k < k:
            max_k = k
    ld = y_hat[max_k] + ((c-list_S[max_k])/(list_S[max_k+1]-list_S[max_k])) * (y_hat[max_k+1]-y_hat[max_k])
    list_x_bar_k = []
    for k in range(len(y)):
        x_bar_k = max(0, min(1, y[k] - ld))
        list_x_bar_k.append(x_bar_k)
    return list_x_bar_k
```

```
def func_F(y, z, X):
    out = 0
    for i in range(X.shape[0]):
        for k in range(X.shape[1]):
            out += (X[i][k] - y[i]*z[k])**2
    return (1/2) * out

def grad_F(y, z, X):
    y_clone = y.clone()
    y_clone = y_clone.detach().requires_grad_()
    out = func_F(y_clone, z, X)
    out.backward()
    return y_clone.grad

def project_grad_descent(X, z, eta, alpha, T):
    for i in range(T):
        eta = torch.tensor(find_projection((eta-alpha*grad_F(eta, z, X)).numpy(), 1), dtype = torch.float64)
    return eta.numpy()

if __name__ == "__main__":
    X = pd.read_csv('homework3_data/project3_matrix.csv')
    X = X.drop(['Unnamed: 0'], axis=1).to_numpy()
    X = torch.tensor(X, dtype = torch.float64)
    alpha, T = 1e-5, 50
    eta = torch.tensor([1/X.shape[0]] * X.shape[0], dtype = torch.float64)
    z = (X.shape[0])**2 * torch.matmul(X.t(), eta)
    eta = project_grad_descent(X, z, eta, alpha, T)
    list_k, list_eta_k = [], []
    for k in range(len(eta)):
        if eta[k] > 0.001:
            list_k.append(k + 1)
            list_eta_k.append(eta[k])
    print("List of indexes k for which " + r'$\eta_T^{(k)}>0.001$' + " is: " + str(list_k))
    print("The corresponding values of " + r'$\eta_T^{(k)}$' + " is: " + str(list_eta_k))
```

Result:
List of indexes k for which $\eta_T^{(k)} > 0.001$ is: [24, 36, 63, 96]
The corresponding values of $\eta_T^{(k)}$ is: [0.2802846365138505, 0.4170689676071295, 0.18152550727653338, 0.1211208886024866]

(3) Consider the iterations, initialized with some $y_0 \in \Omega_1$:

$$y_{i+1} = \arg\min_{\Omega_1} F_{z^*(y_t)}.$$

Show that this algorithm is such that $F(y_t, z^*(y_t))$ is a decreasing sequence.

Solution. Due to $y_{i+1} = \arg\min_{\Omega_1} F_{z^*(y_t)}$, we get

$$F(y_{t+1}, z^*(y_t)) \leq F(y_t, z^*(y_t)). \tag{14}$$

On the other hand, $z^*(y)$ is the optimal z for the minimization of F_y over Ω_2, so we have

$$F(y_{t+1}, z^*(y_{t+1})) \leq F(y_{t+1}, z^*(y_t)) \tag{15}$$

From 14 and 15, we obtain

$$F(y_{t+1}, z^*(y_{t+1})) \leq F(y_t, z^*(y_t)),$$

i.e., this algorithm is such that $F(y_t, z^*(y_t))$ is a decreasing sequence.

(4) Using the notation of question (II-2), denote by $\eta_{\alpha,T}(y,z)$ the value after T updates of the projected gradient descent algorithm initialized with $\eta_0 = y$ and with step size α. Program an algorithm that takes as input x, y_0, T and α and iterates over 100 iterations:

$$y_{t+1} = \eta_{\alpha,T}(y_t, z^*(y_t)).$$

Run this program using the matrix provided in project3_matrix.csv, and with $\alpha = 10^{-5}, T = 10, y_0^{(k)} = 1/n$ for $k = 1, \ldots, n$. Provide, in your answer the indexes k for which $y^{(k)} > 0.001$ and the corresponding values of $\eta_T^{(k)}$, where y is the output of your program.

```python
def get_optim_z(y, X):
    list_z = []
    for k in range(X.shape[1]):
        sum_Xy = 0
        for i in range(X.shape[0]):
            sum_Xy += X[i][k] * y[i]
        sum_Xy = sum_Xy/(torch.norm(y)**2)
        list_z.append(max(0, sum_Xy))
    return list_z

def project_grad_descent_y(X, y, alpha, T):
    for i in range(100):
        z_y = torch.tensor(get_optim_z(y, X), dtype = torch.float64)
        y = project_grad_descent(X, z_y, y, alpha, T)
    return y

if __name__ == "__main__":
    alpha, T = 1e-5, 10
    y = torch.tensor([1/X.shape[0]] * X.shape[0], dtype = torch.float64)
    y = project_grad_descent_y(X, y, alpha, T).numpy()
    list_k, list_y_k = [], []
    for k in range(len(y)):
        if y[k] > 0.001:
            list_k.append(k + 1)
            list_y_k.append(y[k])
    print("List of indexes k for which " + r'$y^{(k)}>0.001$' + " is: " + str(list_k))
    print("The corresponding values of " + r'$y^{(k)}$' + " is: " + str(list_y_k))
```

Result:
List of indexes k for which $y^{(k)} > 0.001$ is: [18, 24, 36, 46, 52, 58, 63, 72, 80, 96]
The corresponding values of $y^{(k)}$ is: [0.109082660499241, 0.0962864473347163, 0.10164719426063612, 0.08132085817922366, 0.11305297011170733, 0.11095008460546285, 0.09241543700155856, 0.11335755192270829, 0.07842293120707078, 0.09005381861309705]