

## ✓ Problem set 1 (55 pts)

### Important information

1. We provide signatures of the functions that you have to implement. Make sure you follow the signatures defined, otherwise your coding solutions will not be graded.
2. Please submit the single Jupyter Notebook file, where only Python and Markdown/*LATEX* are used. Any hand-written solutions inserted by photos or in any other way are prohibitive and will not be graded. If you will have any questions about using Markdown, ask them!
3. The works will be checked for plagiarism. The score will be divided by the number of similar works.

### Problem 1 (25 pts)

Assume we have a set of data points  $x^{(i)} \in \mathbb{R}^n$ ,  $i = 1, \dots, m$ , and decide to represent this data as a matrix

$$X = \begin{pmatrix} | & & | \\ x^{(1)} & \dots & x^{(m)} \\ | & & | \end{pmatrix} \in \mathbb{R}^{n \times m}.$$

We suppose that  $\text{rank } X = r$ .

In all tasks below, we ask you to find the rank of some matrix  $M$  related to  $X$ . In particular, you need to find relation between  $\text{rank } X = r$  and  $\text{rank } M$ , e.g., that the rank of  $M$  is always larger/smaller than the rank of  $X$  or that  $\text{rank } M = \text{rank } X/35$ . Please support your answer with legitimate arguments and make the answer as accurate as possible.

Note that depending on the structure of the matrix  $X$ , border cases are possible. Make sure to cover them in your answer correctly.

**Task 1.** (5 pts) In applied statistics and machine learning, data is often normalized. One particularly popular strategy is to subtract the estimated mean  $\mu$  and divide by the square root of the estimated variance  $\sigma^2$ . i.e.

$$x \rightarrow (x - \mu)/\sigma.$$

After the normalization, we get a new matrix

$$Y := \begin{pmatrix} | & & | \\ y^{(1)} & \dots & y^{(m)} \\ | & & | \end{pmatrix},$$

$$y^{(i)} := \frac{x^{(i)} - \frac{1}{m} \sum_{i=1}^m x^{(i)}}{\sqrt{\frac{1}{m} \sum_{i=1}^m (x^{(i)})^2 - \left(\frac{1}{m} \sum_{i=1}^m x^{(i)}\right)^2}}.$$

What is the rank of  $Y$  if  $\text{rank } X = r$ ?

**Task 2.** (5 pts) To reveal the structure of data one often considers similarity measures such as [cosine distance](#)

$$d_c(x^{(i)}, x^{(j)}) := 1 - \frac{\sum_{k=1}^n x_k^{(i)} x_k^{(j)}}{\sqrt{\sum_{k=1}^n (x_k^{(i)})^2} \sqrt{\sum_{k=1}^n (x_k^{(j)})^2}} = 1 - \frac{(x^{(i)})^\top x^{(j)}}{\|x^{(i)}\|_2 \|x^{(j)}\|_2}.$$

Consider a matrix with cosine distances  $D = [d_{ij}]$  such that

$$d_{ij} = d_c(x^{(i)}, x^{(j)}).$$

What is the rank of  $D$  if  $\text{rank } X = r$ ?

**Task 3.** (15 pts) Transformation in Task 1 has a form  $x \rightarrow ax + b$ , where  $a \in \mathbb{R}$  and  $b \in \mathbb{R}^n$ . In this task, we explore more general transformations.

- Let  $y^{(i)} = Ax^{(i)} + b$ , where  $A \in \mathbb{R}^{p \times n}$ ,  $b \in \mathbb{R}^p$ . What possible ranks matrix

$$Y := \begin{pmatrix} | & & | \\ y^{(1)} & \dots & y^{(m)} \\ | & & | \end{pmatrix}$$

may have for different  $A$ ?

- If  $p > n$  is it possible that  $\text{rank } Y > \text{rank } X$ ?
- If  $p = n$  and both matrix  $A$  and vector  $b$  are not identically zeros how small the rank of  $Y$  can be?
- How large the rank of  $Y$  can be?
- Let  $y^{(i)} = w \odot x^{(i)} + b$ , where  $w, b \in \mathbb{R}^n$  and  $\odot$  is an [Hadamard product](#).

What possible ranks matrix  $Y$  may have for different  $w$  and  $b$ ?

## ✓ Your solution is here

**Problem 1; Task 1:** Let  $\mu = \frac{1}{m} \sum_{i=1}^m x^{(i)}$  and each  $\mu_k = \frac{1}{m} \sum_{i=1}^m x_k^{(i)}$ , also, let  $\sigma^2 = \frac{1}{m} \sum_{i=1}^m (x^{(i)})^2 - \mu$  and each  $\sigma_k^2 = \frac{1}{m} \sum_{i=1}^m (x_k^{(i)})^2 - \mu_k$ . Then the transformation defined in the problem can be represented in two ways:

First way:

$$y_k^{(i)} := \frac{x_k^{(i)} - \mu_k}{\sigma_k}$$

Let

$$M := \begin{pmatrix} | & & | \\ \mu & \dots & \mu \\ | & & | \end{pmatrix} \in \mathbb{R}^{n \times m}$$

$$S := \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & \sigma_m \end{bmatrix} \in \mathbb{R}^{m \times m}$$

Where  $S$  is a diagonal matrix. The transformation assumes that for each  $k$ ,  $\sigma_k \neq 0$ . So, I will be going with the assumption that  $S$  is invertible and hence has full rank (**Note that if  $X$  has a row with same entries through out, this fails to make  $S$  invertible as standard deviation would be zero**). The transformation to then get matrix  $Y$  is:

$$Y = (X - M) S^{-1}$$

Second way:

$$\begin{aligned} x^{(j)} - \frac{1}{m} \sum_{i=1}^m x^{(i)} &= \frac{1}{m} \left( mx^{(j)} - \sum_{i=1}^m x^{(i)} \right) \\ &= \frac{1}{m} \left( mx^{(j)} - x^{(j)} - \sum_{i=1, i \neq j}^m x^{(i)} \right) \\ &= \frac{1}{m} \left( (m-1)x^{(j)} - \sum_{i=1, i \neq j}^m x^{(i)} \right) \\ &= \frac{m-1}{m} x^{(j)} - \frac{1}{m} \sum_{i=1, i \neq j}^m x^{(i)} \end{aligned}$$

So we see a transformation that consists of the linear combination of the columns of  $X$  for which the  $j$ th column is formed by taking  $\frac{m-1}{m} x^{(j)}$  and subtracting the sum of the other vectors multiplied by  $\frac{1}{m}$ . We get a symmetric,  $T_m$  matrix defined thus:  $T_m = [t_{ij}]$ ,  $t_{ij} = \frac{1}{m}$  for  $i \neq j$  and  $t_{ii} = -\frac{1}{m-1}$ .

$$T_m := \begin{bmatrix} \frac{m-1}{m} & -\frac{1}{m} & \dots & -\frac{1}{m} \\ -\frac{1}{m} & \frac{m-1}{m} & \dots & -\frac{1}{m} \\ \vdots & \vdots & \dots & \vdots \\ -\frac{1}{m} & -\frac{1}{m} & \dots & \frac{m-1}{m} \end{bmatrix} \in \mathbb{R}^{m \times m}$$

Then the transformation to get  $Y$  is:

$$Y = XT_m S^{-1}$$

Next, we try to get information on the ranks of  $T_m$ ,  $M$  and  $S$  (although we are moving with the assumption that  $S$  is full rank -  $\text{rank } S = m$ )

Rank of  $M$ : Since the column space of  $M$  consists of only one vector,  $\mu$ , the dimension of the null space is 1 if  $\mu$  is not 0 vector and the dimension is 0 otherwise. Hence,  $\text{rank } M \leq 1$ . (No need to add the lower bound since rank is always nonnegative integer)

Rank of  $T_m$ : We shall prove that this is equal to  $m - 1$  by showing that the dimension of the column space is 1. Consider an arbitrary vector  $\mathbf{v} \in \mathbb{R}^m$  from the null space  $\Rightarrow T_m \mathbf{v} = 0$ . Let  $v_i$  be the  $i$ th component of vector  $\mathbf{v}$ . We then have,

$$\begin{aligned}
T_m \mathbf{v} = 0 &\iff \left(-\frac{1}{m}\right)v_1 + \left(-\frac{1}{m}\right)v_2 + \dots + \left(\frac{m-1}{m}\right)v_i + \dots + \left(-\frac{1}{m}\right)v_m = 0 \forall i \\
&\iff v_i - \frac{1}{m} \sum_{j=1}^m v_j = 0 \forall i \\
&\iff v_i = \frac{\sum_{j=1}^m v_j}{m} \forall i \\
&\Rightarrow v_i = \frac{\sum_{j=1}^m v_j}{m} = v_k \text{ for arbitrary } k \\
&\Rightarrow \sum_{j=1}^m v_j = mv_j \\
&\Rightarrow \mathbf{v} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} v_j
\end{aligned}$$

It can also be easily shown that the column vector with inputs  $1$ s is in the null space of  $T_m$ . Hence the dimension of the null space is 1 and the rank of the matrix is  $\text{rank } T_m = m - 1$ .

Now, by applying some [rank properties](#). First, we can use subadditivity property to get a weak bound.  $Y = XT_m S^{-1} \iff YS = XT_m$  and  $Y = (X - M)S^{-1} \iff YS = X - M$ . Since we assume that  $S$  has full rank, we have that

$$\text{rank}(YS) = \text{rank}(Y) = \text{rank}(XT_m) = \text{rank}(X - M)$$

By subadditivity, we have that,

$$\text{rank}(Y) = \text{rank}(X - M) \leq \text{rank}(X) + \text{rank}(-M) = \text{rank}(X) + \text{rank}(M) \leq r + 1$$

Now let us get a stronger bound.

$$\text{rank}(AB) \leq \min(\text{rank}(A), \text{rank}(B)) \Rightarrow \text{rank}(Y) = \text{rank}(XT_m) \leq \min(\text{rank}(X), \text{rank}(T_m)) = \min(r, m - 1) \leq r$$

. Also, by Sylvester's rank inequality,

$$\text{rank}(Y) = \text{rank}(XT_m) \geq \text{rank}(X) + \text{rank}(T_m) - m \Rightarrow \text{rank}(Y) \geq r + m - 1 - m = r - 1$$

Thus, we have that  $r - 1 \leq \text{rank}(Y) \leq r$ . Since rank is a non negative integer, we have  $\text{rank}(Y) = r$  or  $\text{rank}(Y) = r - 1$ .

For case  $\text{rank}(Y) = r - 1$  we provide an example.

$$X = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \rightarrow Y = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

For case  $\text{rank}(Y) = r$  we provide an example.

$$X = \begin{pmatrix} 2 & 0 \\ 2 & 0 \end{pmatrix} \rightarrow Y = \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}$$

**Problem 1; Task 2:** Let  $\hat{\mathbf{x}}$  be the unit vector parallel and in same direction to  $\mathbf{x}$ . Let

$$\hat{X} = \begin{pmatrix} | & & | \\ \hat{x}^{(1)} & \dots & \hat{x}^{(m)} \\ | & & | \end{pmatrix} \in \mathbb{R}^{n \times m}.$$

It is obvious that the column space of  $\hat{X}$  and  $X$  are the same. Hence, the rank of  $\hat{X}$  is  $r$ . Let,  $F \in \mathbb{R}^{m \times m}$  be a matrix where all entries are 1.

$$F := \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \dots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

Next, we can show that the transformation  $D$  is equivalent to  $D = F - \hat{X}^T \hat{X}$  (**Note that if any column of  $X$  is all zeroes, this transformation is not possible as the length of that column vector would zero**). By property of [Gram matrix](#), we have that

$$\text{rank}(\hat{X}^T \hat{X}) = \text{rank}(\hat{X}) = \text{rank}(X) = r. \text{ It can also be easily gotten that } \text{rank}(F) = 1$$

By subadditivity, we have that,

$$\begin{aligned}\text{rank}(D) &= \text{rank}(F - \hat{X}^T \hat{X}) \leq \text{rank}(F) + \text{rank}(-\hat{X}^T \hat{X}) \\ &= \text{rank}(F) + \text{rank}(\hat{X}^T \hat{X}) \\ &= 1 + r \\ &\Rightarrow \text{rank}(D) \leq r + 1\end{aligned}$$

$$\text{Also, } \hat{X}^T \hat{X} = F - D$$

$$\begin{aligned}\text{rank}(\hat{X}^T \hat{X}) &= r = \text{rank}(F - D) \leq \text{rank}(F) + \text{rank}(-D) \\ &= \text{rank}(F) + \text{rank}(D) \\ &= 1 + \text{rank}(D) \\ &\Rightarrow r - 1 \leq \text{rank}(D)\end{aligned}$$

$$\text{Hence, } r - 1 \leq \text{rank}(D) \leq r + 1$$

$$\text{rank}(D) = r - 1 \text{ or } \text{rank}(D) = r \text{ or } \text{rank}(D) = r + 1$$

For case  $\text{rank}(D) = r + 1$  we provide an example.

$$X = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix} \rightarrow D = \begin{pmatrix} 0 & 1 & 1 - \frac{\sqrt{2}}{2} \\ 1 & 0 & 1 - \frac{\sqrt{2}}{2} \\ 1 - \frac{\sqrt{2}}{2} & 1 - \frac{\sqrt{2}}{2} & 0 \end{pmatrix}$$

For case  $\text{rank}(D) = r$  we provide an example.

$$X = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \rightarrow D = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

For case  $\text{rank}(D) = r - 1$  we provide an example.

$$X = \begin{pmatrix} 1 & 1 \end{pmatrix} \rightarrow D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

**Problem 1; Tasks 3.1, 3.2:** Let matrix  $B$  be defined thus:

$$B := \begin{pmatrix} | & & | \\ b & \dots & b \\ | & & | \end{pmatrix} \in \mathbb{R}^{p \times m}$$

Where the column space of  $B$  consists of one vector, so  $\text{rank}(B) = 1$  or 0 if  $b$  is the zero vector.  $\text{rank}(B) \leq 1$ . It can then be shown that  $Y = AX + B$ . Also,  $\text{rank}(AX) \leq \min(\text{rank}(A), \text{rank}(X)) \leq \text{rank}(X)$

By subadditivity, we have,

$$\begin{aligned}\text{rank}(Y) &= \text{rank}(AX + B) \leq \text{rank}(AX) + \text{rank}(B) \\ &\Rightarrow \text{rank}(Y) \leq \text{rank}(X) + 1\end{aligned}$$

This inequality shows that it might be possible, now let's find an example. Set  $p = 4, n = 3, m = 3$

$$\begin{aligned}A &:= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \\ X &:= \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{bmatrix} \\ b &:= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 3 \end{bmatrix}\end{aligned}$$

This gives rise to such matrix for  $Y$

$$Y := \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \\ 3 & 3 & 3 \end{bmatrix}$$

It is easy to check that  $\text{rank}(Y) = 3 > \text{rank}(X) = 2$ .

**Problem 1; Tasks 3.3, 3.4:** If  $b$  is not identically zero, then  $\text{rank}(B) = 1$ . To bound the rank of  $Y$ , we can Sylvester's rank inequality and get  $\text{rank}(Y - B) = \text{rank}(AX) \geq \text{rank}(A) + \text{rank}(X) - n = \text{rank}(A) + \text{rank}(X) - p$ . (Since we are given the condition that  $p = n$ )

By subadditivity,

$$\text{rank}(Y) + 1 = \text{rank}(Y) + \text{rank}(B) = \text{rank}(Y) + \text{rank}(-B) \geq \text{rank}(Y - B)$$

Combining these results,

$$\begin{aligned} \text{rank}(Y) + 1 &\geq \text{rank}(A) + \text{rank}(X) - p \\ \Rightarrow \text{rank}(Y) &\geq \text{rank}(A) + \text{rank}(X) - p - 1 \end{aligned}$$

So,  $\text{rank}(Y)$  is at least  $\max(0, \text{rank}(A) + \text{rank}(X) - 1 - p)$ , since rank is always greater than zero.

As earlier gotten in this task,  $\text{rank}(Y)$  is at most  $\text{rank}(X) + 1$ . ( $\text{rank}(Y) \leq \text{rank}(X) + 1$ )

**Problem 1; Task 3.5:** Let  $\Omega := \begin{bmatrix} \omega_1 & 0 & \dots & 0 \\ 0 & \omega_2 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & \omega_n \end{bmatrix} \in \mathbb{R}^{n \times n}$  where  $\omega_i$  is the  $i$ th component of the vector  $\omega$

Then the Hadamard task can be interpreted in the language of linear transformation as  $Y = \Omega X + B$ , where the definition of  $B$  is analogous as in previous task ( $m$  columns of vector  $b$ ). Then by subadditivity,

$$\text{rank}(Y) + 1 \geq \text{rank}(Y) + \text{rank}(B) = \text{rank}(Y) + \text{rank}(-B) \geq \text{rank}(Y - B)$$

And, by Sylvester's rank inequality,

$$\begin{aligned} \text{rank}(Y - B) &= \text{rank}(\Omega X) \geq \text{rank}(\Omega) + \text{rank}(X) - m \\ \Rightarrow \text{rank}(Y) &\geq \text{rank}(\Omega) + \text{rank}(X) - m - 1 \\ \Rightarrow \text{rank}(Y) &\geq \max(0, \text{rank}(\Omega) + \text{rank}(X) - m - 1) \\ \Rightarrow \min(n, m) &\geq \text{rank}(Y) \geq \max(0, \text{rank}(\Omega) + \text{rank}(X) - m - 1) \end{aligned}$$

since we know the rank is at most the minimum of the dimension of a matrix. Setting  $\omega$  and  $b$  to be the zero vectors or vectors with only 1s as entries, we can get some edge cases.

## Problem 2 (10 pts)

- Let  $\|\cdot\|$  and  $\|\cdot\|'$  be norms on a vector space  $X$ , and let  $B$  and  $B'$  denote the corresponding close unit balls. Prove that  $B \subseteq B'$  iff  $\|\cdot\|' \leq \|\cdot\|$  for any input vectors.

Let  $1 \leq p \leq q \leq \infty$ :

- Prove that  $\|\cdot\|_q \leq \|\cdot\|_p$  on  $\mathbb{R}^n$
- Show that there exists a constant  $C = C(n, p, q) > 0$  such that  $\|\cdot\|_p \leq C\|\cdot\|_q$  on  $\mathbb{R}^n$
- Can the above constant be chosen in such a way that it does not depend on  $n$ ?
- Find the smallest possible  $C(n, p, q)$  with the above property.

## ✓ Your solution is here

**Problem 2, task 1:** Let us define  $B$  and  $B'$  thus -  $B := \{x | \|x\| \leq 1\}$  and  $B' := \{x | \|x\|' \leq 1\}$ .

$$\begin{aligned} \|\cdot\|' \leq \|\cdot\| &\iff (\|x\| \leq C \Rightarrow \|x\|' \leq 1) \forall x \in X \text{ and } \forall C \in \mathbb{R}^+ \\ &\iff \left(\left\|\frac{x}{C}\right\| \leq 1 \Rightarrow \left\|\frac{x}{C}\right\|' \leq 1\right) \forall x \in X \text{ and } \forall C \in \mathbb{R}^+ \\ &\iff \left(\frac{x}{C} \in B \Rightarrow \frac{x}{C} \in B'\right) \forall x \in X \text{ and } \forall C \in \mathbb{R}^+ \\ &\iff (B \subseteq B') \end{aligned}$$

**Problem 2, task 2:** Define a function  $f: \mathbb{R}_0^+ \rightarrow \mathbb{R}$ , thus:

$$f(x) = (1+x)^p - x^p - 1 \text{ where } 1 \leq p$$

We want to show that,  $0 \leq f(x)$ . First we compute the derivative,  $f'(x) = p(1+x)^{p-1} - px^{p-1}$ . Now, for nonnegative  $x$ , we use the fact that  $g(x) = x^r \Rightarrow g'(x) = rx^{r-1} \geq 0$  given that  $r, x \geq 0$ , hence  $g$  is increasing if  $r, x \geq 0$ . Thus,

$$\begin{aligned} x &\leq (1+x) \\ \Rightarrow x^{p-1} &\leq (1+x)^{p-1} \\ \Rightarrow px^{p-1} &\leq p(1+x)^{p-1} \\ 0 &\leq f'(x) \end{aligned}$$

Hence, we have that  $f$  is increasing.  $f(0) = 0$  thus, we have that  $0 \leq f(x)$ . Next, we prove by induction the following fact:

$$\begin{aligned} (|x_1| + |x_2| + \dots + |x_n|)^p &\geq |x_1|^p + |x_2|^p + \dots + |x_n|^p \\ |x_1|^p &\geq |x_1|^p \end{aligned}$$

Statement holds obviously for  $n = 1$ . Suppose true up to  $n - 1$ .

$$\begin{aligned} & (|x_1| + |x_2| + \dots + |x_{n-1}|)^p \geq |x_1|^p + |x_2|^p + \dots + |x_{n-1}|^p \\ & f\left(1 + \frac{|x_n|}{|x_1| + |x_2| + \dots + |x_{n-1}|}\right) \geq 0 \\ \Rightarrow & \left(1 + \frac{|x_n|}{|x_1| + |x_2| + \dots + |x_{n-1}|}\right)^p \geq \left(\frac{|x_n|}{|x_1| + |x_2| + \dots + |x_{n-1}|}\right)^p + 1 \\ & (|x_1| + |x_2| + \dots + |x_{n-1}| + |x_n|)^p \geq |x_n|^p + (|x_1| + |x_2| + \dots + |x_{n-1}|)^p \\ & \text{And by the case } n - 1, \text{ we have} \\ & (|x_1| + |x_2| + \dots + |x_{n-1}| + |x_n|)^p \geq |x_1|^p + |x_2|^p + \dots + |x_{n-1}|^p + |x_n|^p \\ \Rightarrow & \sum_{i=1}^n |x_i|^p \leq \left(\sum_{i=1}^n |x_i|\right)^p \end{aligned}$$

Proving the fact. We can use this fact. Setting  $|x_i| \rightarrow |x_i|^p \geq 0$  and  $p \rightarrow \frac{q}{p} \geq 1$ . We get,

$$\begin{aligned} \sum_{i=1}^n (|x_i|^p)^{\frac{q}{p}} & \leq \left(\sum_{i=1}^n |x_i|^p\right)^{\frac{q}{p}} \iff \left(\sum_{i=1}^n |x_i|^q\right)^{\frac{1}{q}} \leq \left(\sum_{i=1}^n |x_i|^p\right)^{\frac{1}{p}} \\ & \iff \|\cdot\|_q \leq \|\cdot\|_p \end{aligned}$$

We know that  $\|\mathbf{x}\|_\infty = |x_{\max}|$ , now

$$\begin{aligned} \left(\sum_{i=1}^n |x_i|^p\right)^{\frac{1}{p}} & = |x_{\max}| \left(\sum_{i=1}^n \frac{|x_i|^p}{|x_{\max}|^p}\right)^{\frac{1}{p}} = |x_{\max}| \left(1 + \sum_{i=1, i \neq \max}^n \frac{|x_i|^p}{|x_{\max}|^p}\right)^{\frac{1}{p}} \geq |x_{\max}| \\ & \Rightarrow \|\cdot\|_p \geq \|\cdot\|_\infty \end{aligned}$$

**Problem 2; tasks 3, 4, 5:** Define the  $M_p(|x_1|, |x_2|, \dots, |x_n|)$  to be the [general mean with equal weights](#). Given  $1 \leq p \leq q$ , by general, mean inequality, we have,

$$\begin{aligned} M_p(|x_1|, |x_2|, \dots, |x_n|) & \leq M_q(|x_1|, |x_2|, \dots, |x_n|) \\ \Rightarrow & \left(\frac{1}{n} \sum_{i=1}^n |x_i|^p\right)^{\frac{1}{p}} \leq \left(\frac{1}{n} \sum_{i=1}^n |x_i|^q\right)^{\frac{1}{q}} \\ & \Rightarrow \frac{\|\cdot\|_p}{n^{\frac{1}{p}}} \leq \frac{\|\cdot\|_q}{n^{\frac{1}{q}}} \\ & \Rightarrow \|\cdot\|_p \leq n^{\frac{1}{p} - \frac{1}{q}} \|\cdot\|_q \\ & \Rightarrow \|\cdot\|_q \leq \|\cdot\|_p \leq n^{\frac{1}{p} - \frac{1}{q}} \|\cdot\|_q \\ & \Rightarrow 1 \leq \frac{\|\cdot\|_p}{\|\cdot\|_q} \leq n^{\frac{1}{p} - \frac{1}{q}} \end{aligned}$$

At some point above, we applied the result got from task 2.

We have that  $C = C(n, p, q) = n^{\frac{1}{p} - \frac{1}{q}}$ . It can be shown that  $C(n, p, q)$  is the minimum possible constant. We see that, for any  $p \leq q$ ,  $\frac{\|\cdot\|_p}{\|\cdot\|_q} \leq C(n, p, q)$  with equality holding when all  $x_i$ s are equal (The condition of General Mean Inequality). Suppose we can choose a constant,  $K$  independent of  $n$ , such that  $\|\cdot\|_p \leq K \|\cdot\|_q$ , since  $C$  is the minimum, we have that  $K \geq \sup \frac{\|\cdot\|_p}{\|\cdot\|_q} = C$ . Since,  $C$  can grow infinitely large depending on  $n$ , we have that such  $K$  does not exist.

## ✓ Problem 3: Recover 3D molecule structure via low-rank approximation (20 pts)

### Chromatin

Imagine you are given the task of understanding the structure of a complex geometric shape, but there's a catch: you can not observe the shape directly in three dimensions. Instead, you can only infer its structure by studying the points of contact between different parts of the shape. This is analogous to the challenge faced in 3D chromatin modeling, where the goal is to understand the three-dimensional organization of the DNA sequence within a cell's nucleus—a structure too tiny and complex to be directly observed in detail.

To tackle this, scientists employ a technique known as Hi-C. The Hi-C technique generates a symmetric matrix such that elements in the matrix quantify the interaction frequencies between pairs of chromatin segments. The higher values indicate a higher propensity for spatial proximity. We consider  $N$  segments, therefore the dimension of the resulting matrix is  $N \times N$ .

Mathematically, the matrix is sparse due to the infrequent interactions between most DNA segments and has a block diagonal structure reflecting clusters of interactions within certain genomic regions. The matrix patterns are analyzed using algorithms and statistical models to infer the three-dimensional chromatin structure.

Let us inspect what Hi-C looks like!

```
1 !pip install plotly
2 !pip install patsy
```

```

➦ Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: plotly in c:\programdata\anaconda3\lib\site-packages (5.9.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\programdata\anaconda3\lib\site-packages (from plotly) (8.0.1)
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: patsy in c:\programdata\anaconda3\lib\site-packages (0.5.3)
Requirement already satisfied: six in c:\users\acer\appdata\roaming\python\python310\site-packages (from patsy) (1.16.0)
Requirement already satisfied: numpy>=1.4 in c:\programdata\anaconda3\lib\site-packages (from patsy) (1.23.5)

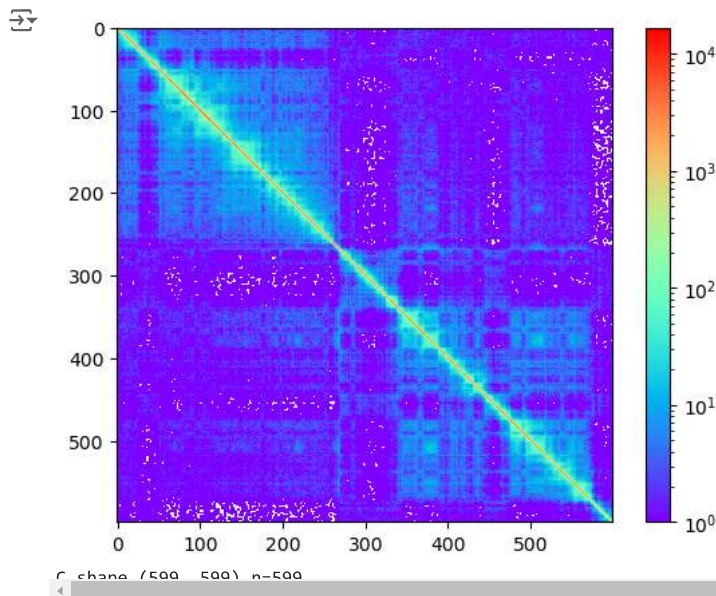
```

```

1 import numpy as np
2 import pandas as pd
3 import scipy as sp
4 from matplotlib import pyplot as plt, cm
5 from matplotlib import colors

1 path = 'IMR90_100kb_chr20.csv'
2 C = np.array(pd.read_csv(f'{path}', header=None))
3
4 # Removes the blanks obtained during the experiment
5 mask = (C == 0).all(0)
6 index = np.where(~mask)[0]
7 C = C[~mask,:]
8 C = C[:,~mask]
9
10
11 im = plt.imshow(C, cmap=cm.rainbow, norm=colors.LogNorm())
12 plt.colorbar(im)
13 plt.show()
14
15 print('C shape', C.shape, f'n={C.shape[0]}')

```



Recall that our goal is to infer the 3D chromatin structure. Mathematically, we model the chromatin as a smooth curve in 3D space. That is, the

points lie on a smooth parametrized curve  $x_1, \dots, x_n \in \gamma(t)$ , where  $\gamma(t) = \begin{pmatrix} \gamma_1(t) \\ \gamma_2(t) \\ \gamma_3(t) \end{pmatrix}$ .

Each coordinate can be expanded using [cubic spline basis functions](#)  $h_1(t), \dots, h_k(t)$  evaluated at the sequence of coordinates at  $t = 1, \dots, n = 599$ :

$$\gamma_j(t) = \sum_{\ell=1}^k \Theta_{\ell j} h_{\ell}(t), \quad j = 1, 2, 3$$

$$x_i = \gamma(i) = \begin{pmatrix} \sum_{\ell=1}^k \Theta_{\ell 1} h_{\ell}(i) \\ \sum_{\ell=1}^k \Theta_{\ell 2} h_{\ell}(i) \\ \sum_{\ell=1}^k \Theta_{\ell 3} h_{\ell}(i) \end{pmatrix}$$

$$X = \begin{pmatrix} -x_1^\top - \\ \dots \\ -x_n^\top - \end{pmatrix} = \begin{pmatrix} | & | & | \\ \gamma_1 & \gamma_2 & \gamma_3 \\ | & | & | \end{pmatrix} \in \mathbb{R}^{n \times 3}$$

Thus, we can rewrite the coordinates of the chromatin stacked in matrix  $X$  as product  $X = H\Theta$ , where

$$H = \begin{pmatrix} | & & | \\ h_1 & \dots & h_k \\ | & & | \end{pmatrix} \in \mathbb{R}^{n \times k}$$

The optimization problem that will infer the 3D structure is the following:

$$\ell(x_1, \dots, x_n) = \sum_{i=1}^n \sum_{j=1}^n (Z_{ij} - \langle x_i, x_j \rangle)^2 \rightarrow \min_{\Theta \in \mathbb{R}^{k \times 3}}$$

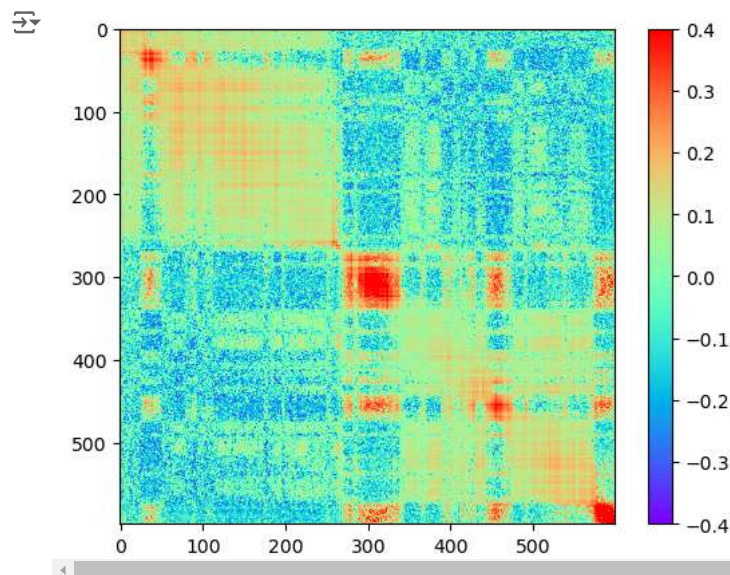
or equivalently

$$\min_{\Theta \in \mathbb{R}^{k \times 3}} \|Z - S(X)\|_F^2 = \min_{\Theta \in \mathbb{R}^{k \times 3}} \|Z - S(H\Theta)\|_F^2$$

where  $Z$  is a normalized Hi-C matrix and  $S$  is some additional map.

The following cell performs the normalization step so you do not have to worry about it.

```
1 import patsy
2
3 def scale(y, center=True, scale=True):
4     x = y.copy()
5     if center:
6         x -= x.mean(axis=0)
7     if scale and center:
8         x /= x.std()
9     return x
10
11 df = 50
12 n_knots = df - 4
13 knots = np.linspace(1, max(index), n_knots, dtype='int64')
14 knots = knots[1:n_knots-1]
15 B0 = patsy.bs(index, df = df, include_intercept=True)
16 H = B0[:, :]
17
18 D = 1/(C+1)
19 Z = -D**2/2
20 Z = scale(Z, center = True, scale = False)
21 Z = (scale(Z.T, center = True, scale = False)).T
22
23 im = plt.imshow(Z, cmap=cm.rainbow, norm=colors.Normalize(vmin=-0.4, vmax=0.4))
24 plt.colorbar(im)
25 plt.show()
```



**Task 1.** (5 pts) Prove the Lemma:

If  $H \in \mathbb{R}^{n \times k}$  is a matrix with orthogonal columns, i.e.  $H^\top H = I$ , and  $S(X) = XX^\top$  then problem

$$\min_{\Theta \in \mathbb{R}^{k \times 3}} \|Z - S(H\Theta)\|_F^2$$

is equivalent to

$$\min_{\Theta \in \mathbb{R}^{k \times 3}} \|H^\top ZH - \Theta\Theta^\top\|_F^2.$$

✓ Your solution is here



We can apply properties of cyclic properties of trace and properties of [Frobenius norm](#) ( $\|\cdot\|_F$ ) for real valued matrices:

$$\|A^T A\|_F = \|AA^T\|_F = \|A\|_F^2, \|A - B\|_F^2 = \|A\|_F^2 + \|B\|_F^2 - 2\text{tr}(AB).$$

$$\begin{aligned}\|Z - S(H\Theta)\|_F^2 &= \|Z - H\Theta\Theta^T H^T\|_F^2 \\ &= \|Z\|_F^2 + \|H\Theta(H\Theta)^T\|_F^2 - 2\text{tr}(ZH\Theta\Theta^T H^T) \\ &= \|Z\|_F^2 + \|(H\Theta)^T H\Theta\|_F^2 - 2\text{tr}(ZH\Theta\Theta^T H^T) \\ &= \|Z\|_F^2 + \|\Theta^T H^T H\Theta\|_F^2 - 2\text{tr}(ZH\Theta\Theta^T H^T) \\ &= \|Z\|_F^2 + \|\Theta^T \Theta\|_F^2 - 2\text{tr}(ZH\Theta\Theta^T H^T) \\ \|H^T ZH - \Theta\Theta^T\|_F^2 &= \|H^T ZH\|_F^2 + \|\Theta\Theta^T\|_F^2 - 2\text{tr}(H^T ZH\Theta\Theta^T) \\ &= \|H^T ZH\|_F^2 + \|\Theta^T \Theta\|_F^2 - 2\text{tr}(ZH\Theta\Theta^T H^T)\end{aligned}$$

Let  $\|\Theta^T \Theta\|_F^2 - 2\text{tr}(ZH\Theta\Theta^T H^T)$  be denoted by  $f(\Theta)$ . Then,

$$\begin{aligned}\min_{\Theta \in \mathbb{R}^{k \times 3}} \|Z - S(H\Theta)\|_F^2 &= \min_{\Theta \in \mathbb{R}^{k \times 3}} (\|Z\|_F^2 + \|\Theta^T \Theta\|_F^2 - 2\text{tr}(ZH\Theta\Theta^T H^T)) \\ &= \|Z\|_F^2 + \min_{\Theta \in \mathbb{R}^{k \times 3}} f(\Theta)\end{aligned}$$

And

$$\begin{aligned}\min_{\Theta \in \mathbb{R}^{k \times 3}} \|H^T ZH - \Theta\Theta^T\|_F^2 &= \min_{\Theta \in \mathbb{R}^{k \times 3}} (\|H^T ZH\|_F^2 + \|\Theta^T \Theta\|_F^2 - 2\text{tr}(ZH\Theta\Theta^T H^T)) \\ &= \|H^T ZH\|_F^2 + \min_{\Theta \in \mathbb{R}^{k \times 3}} f(\Theta)\end{aligned}$$

We can see that in both cases, it boils down to minimising  $f(\Theta)$ .

## Task 2. Solving the problem

$$\min_{\Theta \in \mathbb{R}^{k \times 3}} \|H^\top ZH - \Theta\Theta^\top\|_F^2.$$

can be interpreted as approximating the matrix  $H^\top ZH$  by a positive semidefinite rank-3 matrix  $\Theta\Theta^\top$ .

Assuming that the symmetric matrix  $H^\top ZH$  has at least 3 positive eigenvalues the solution can be found via eigendecomposition of  $H^\top ZH$ :

Let  $H^\top ZH = Q\Lambda Q^\top$  for orthogonal  $Q$  and diagonal  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$  with  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ , then

$$\Theta = Q\sqrt{\Lambda_3}, \text{ where } \sqrt{\Lambda_3} = \text{diag}(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \sqrt{\lambda_3}, \dots, 0, 0).$$

The computational efficiency of the approach derives from the fact that it relies on eigen-decomposition of a small  $k \times k$  matrix, requiring only  $O(k^3)$  additional operations.

In this task, we will write a function that finds the optimal 3D chromatin representation for the given Hi-C matrix. We do that by optimizing the functional from task 1.

- (5 pts) If you recall, the lemma from task 1 requires  $H$  to have orthogonal columns. Write a code to verify whether it is true. If it is not, propose and compare the ways to orthogonalize it.
- (10 pts) You are given below a template for function `low_rank_approximation`. Complete the code for minimizing  $\tilde{\ell}(\Theta)$  using the proposed low rank approach. Remember that we model chromatin as 3D curve so the rank of our decomposition should be 3. However, compare it for different ranks: plot the loss value against the ranks (small, medium, large) and plot the reconstructed matrix. Which rank would be a good fit?

```
1 def are_columns_orthogonal(matrix, tol=1e-10):
2     dot_products = matrix.T @ matrix
3     off_diagonal_sum = np.sum(np.abs(dot_products - np.diag(np.diagonal(dot_products))))
4     return off_diagonal_sum < tol
5
6 is_orthogonal = are_columns_orthogonal(H)
7
8 if is_orthogonal:
9     print("The columns of H are orthogonal.")
10 else:
11     print("The columns of H are not orthogonal.")
12
13 # If H is not orthogonal, propose and compare ways to orthogonalize it
14 if not is_orthogonal:
15     # Propose and apply a method to orthogonalize H, e.g., Gram-Schmidt process
16     orthogonalized_H = np.linalg.qr(H)[0]
17
18     # Check if the orthogonalized matrix is indeed orthogonal
19     is_orthogonalized = are_columns_orthogonal(orthogonalized_H)
20
21     if is_orthogonalized:
```

```

22     print("After orthogonalization, the columns of H are now orthogonal.")
23     else:
24         print("Orthogonalization did not succeed in making the columns of H orthogonal.")
25 H = orthogonalized_H
26 H

```

↗ The columns of H are not orthogonal.

After orthogonalization, the columns of H are now orthogonal.

```

array([[ -6.51379967e-01, -3.05360877e-01,  1.72197890e-01, ...,
         1.16882104e-13,  7.18124381e-14,  2.84444414e-14],
       [-5.09548635e-01, -1.01455598e-01,  3.49084713e-02, ...,
         1.73073469e-14,  1.06346542e-14,  4.19930513e-15],
       [-3.89962149e-01,  5.87941602e-02, -6.12222214e-02, ...,
        -4.88145501e-14, -2.99917877e-14, -1.18709328e-14],
       ...,
       [ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00, ...,
        -2.11106269e-02,  2.73769068e-01, -2.93544967e-01],
       [ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00, ...,
        -5.59415870e-03,  1.61305786e-01, -4.95429256e-01],
       [ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00, ...,
         0.00000000e+00,  0.00000000e+00, -7.49053288e-01]])

```

We can use the Housholder transformation (QR decomposition) to get a good approximation for orthogonalisation since it is a more stable algorithm. The numpy algorithm actually utilises this. Other methods of orthogonalisation are:

Gram–Schmidt process, which uses projection

Givens rotation

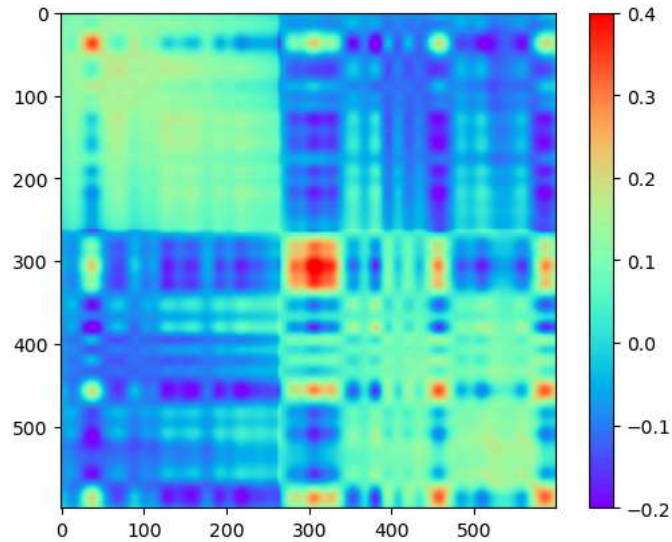
Symmetric orthogonalization, which uses the Singular value decomposition

```

1
2 loss_ = lambda X, Z: np.mean((Z - X @ X.T)**2)
3
4 def low_rank_approximation(Z, H, rank=3):
5     '''
6     Solves the minimization problem from task 1 using diagonalization
7
8     - Z: normalized Hi-C counts matrix
9     - H: orthogonal matrix
10    - rank: the rank of the eigendecomposition
11
12    returns:
13    - Theta: optimal Theta
14    - loss: calculated using `loss_`
15    - adj_rank: the rank adjusted for numerical stability. Usage: adj_rank = np.sum(eigen_values > 1e-12)
16    '''
17
18    # Eigenvalue decomposition of H
19    eigenvalues, eigenvectors = np.linalg.eigh(H.T @ Z @ H)
20
21    # Select the top-rank eigenvectors
22    top_eigenvectors = eigenvectors[:, -rank:]
23
24    # Diagonal matrix of eigenvalues
25    diagonal_matrix = np.diag(np.sqrt(eigenvalues[-rank:]))
26
27    # Optimal Theta based on diagonalization
28    Theta = top_eigenvectors @ diagonal_matrix
29
30    # Reconstruction of X
31    X = H @ Theta
32
33    adj_rank = np.sum(eigenvalues > 1e-12)
34
35    return {'Theta': Theta,
36            'X': X,
37            'loss': loss_(X, Z),
38            'adj_rank': adj_rank}
39
40 Theta, X, loss, rank = low_rank_approximation(Z, H, rank=3).values()
41 Z_hat = X @ X.T
42 print('The optimal loss value is', loss)
43
44 im = plt.imshow(Z_hat, cmap=cm.rainbow, norm=colors.Normalize(vmin=-0.2, vmax=0.4))
45 plt.colorbar(im)
46 plt.show()

```

The optimal loss value is 0.024434653807743647



1 Theta @ Theta.T

```
array([[ 0.54030422, -0.69087154, -0.8341042 , ..., -0.29862481,  
        0.2102539 , -0.06647736],  
       [-0.69087154,  0.88443115,  1.06338257, ...,  0.4308615 ,  
       -0.29373957,  0.09960432],
```