Artificial Intelligence and Machine Learning - Unit II
Self-Assessment - April 11, 2024

First and last name, Student ID: ______________________________

1. Answer the following questions by reporting the mathematical procedure. If you have to compute the actual value, please write the procedure that leads you to the numerical values. **Read well the text before proceeding!**

   (a) In Eq. (1) left, $\mathbf{X}$ is a design matrix where each column is a sample. How many samples are present in the design matrix $\mathbf{X}$? What is the dimensionality of a sample? Compute the $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ to compute the empirical average and the covariance matrix associated to $\mathbf{X}$. $\boxed{1}$

$$\mathbf{X} = \begin{bmatrix} 1 & -5 & 10 & 2 & -3 \\ 2 & 10 & -5 & -1 & 4 \end{bmatrix} \quad \boldsymbol{\mu} = \begin{bmatrix} \quad \end{bmatrix} \quad \boldsymbol{\Sigma} = \begin{bmatrix} \quad \end{bmatrix} \tag{1}$$

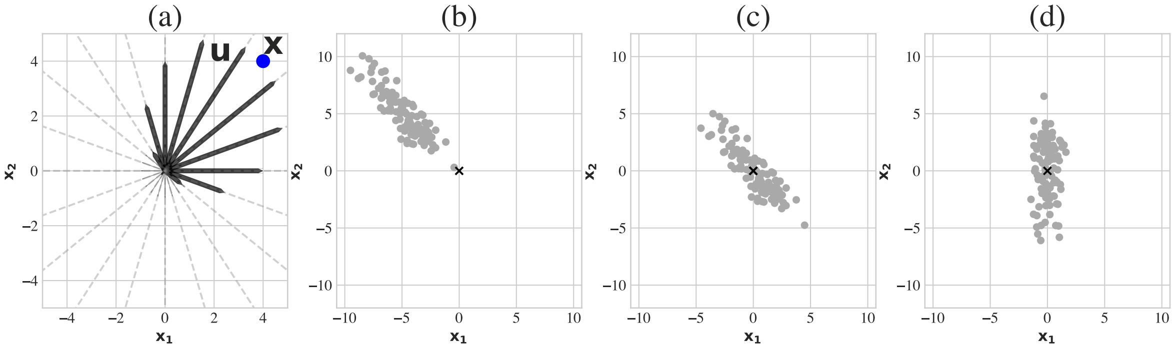

Figure 1: Projection and Point cloud

   (b) Referring to Fig. 1 (a), assume you have a training point: $\mathbf{x} \in \mathbb{R}^2$ and a unit vector $\mathbf{u}$—thus $||\mathbf{u}||_2 = 1$—that functions as a direction passing through the origin. Define with linear algebra the projection of $\mathbf{x}$ over $\mathbf{u}$. Now $\mathbf{x}$ is fixed and you can rotate $\mathbf{u}$: how can you set $\mathbf{u}$ to maximize the projection length? What is the maximum value of the projection length? Black segments in Fig. 1 (a) indicate the projection length over varying directions $\mathbf{u}$. $\boxed{3}$

   (c) A 2D point cloud $\mathbf{X} \doteq \{\mathbf{x}_i\}_{i=1}^N$ is shown in Fig. 1 (b). Fig. 1 (c) shows the same but centered $\bar{\mathbf{X}}$. How do we center the point cloud $\mathbf{X}$ to $\bar{\mathbf{X}}$? Assuming $\mathbf{X} \in \mathbb{R}^{N \times 2}$, which means is given to you as a matrix of N rows and 2 columns, write the one liner `numpy` code to perform the centering. What does `numpy` try to do when shape of matrices do not match? *(Numpy syntax is not important: write mathematically how to center, on which axis to reduce the data)* $\boxed{1}$

   (d) Given the centered point cloud $\bar{\mathbf{X}}$ in (c), which transformation you apply to make it as Fig. 1 (d)? How do you compute this transformation? After the transformation, what happens to the covariance matrix? $\boxed{3}$

Total for Question 1: 8

2. We are given a set of points $\mathbf{X}$ in 2D with no associated labels shown in Tab. 1. We wish to find the main 2 clusters identified by the centers $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2\}$. We hypothesize that the clusters distribute as Gaussian blobs with <u>the same standard deviation across clusters</u>, that is, clusters more or less will distribute as spheres all of the same size.

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$
[2.44, 0.96]	[2.28, 1.06]	[1.45, 4.23]	[1.91, 3.82]	[2.13, 1.62]	[0.92, 4.51]

Table 1: Training set

(a) Define the objective function that can solve the clustering problem mentioned above and describe the necessary steps to minimize the function. $\boxed{3}$

(b) Assume that the two starting cluster centers are $\boldsymbol{\mu}_1 = [1.4, 3.0]$ and $\boldsymbol{\mu}_2 = [1.8, 2.0]$. Given that now we know the cluster centers, compute the **assignment step**. Fill in the blanks alongside each point to indicate the assignment for that point. Show the procedure that was used to get the assignment just for point $\mathbf{x}_3$. $\boxed{1\frac{1}{2}}$

$$\left[\mathbf{x}_1 \ \_\_\_; \quad \mathbf{x}_2 \ \_\_\_; \quad \mathbf{x}_3 \ \_\_\_; \quad \mathbf{x}_4 \ \_\_\_; \quad \mathbf{x}_5 \ \_\_\_; \quad \mathbf{x}_6 \ \_\_\_;\right]. \tag{4}$$

(c) Given the assignments you have found in Eq. (4), now compute the **update step** and explaining the procedure for your computation. $\boxed{1\frac{1}{2}}$

(d) Let's assume that you have an image $\mathbf{x}$ of dimension H×W. Each pixel is expressed as a linear combination of red, green and blue, where each color component is quantized with 8 bits. $\boxed{3}$

◇ How many bits you have to transmit in total if you want to send this image to a friend over the internet? See Fig. 2 for a sketch.

Now assume that you cluster all the pixels of $\mathbf{x}$ in the RGB color space with k-means and you impose 4 clusters in total.

◇ Given the clustering result, describe what you need to send to your friend so that you can save bandwidth and he is able to approximately reconstruct the image.

◇ Describe the procedure your friend use to reconstruct the image. Is he going to get the exact same image you sent?

◇ How many bits do you have to send now if you use clustering?

◇ Is it possible to find the value of r=H×W for which you start to save bandwidth if you use k-means instead of sending the entire image?

Assume your friend knows a) the dimension H×W b) the order that you are using to send the pixels—row by row top to bottom, for each row left to right c) he knows how to interpret additional information of the clustering.

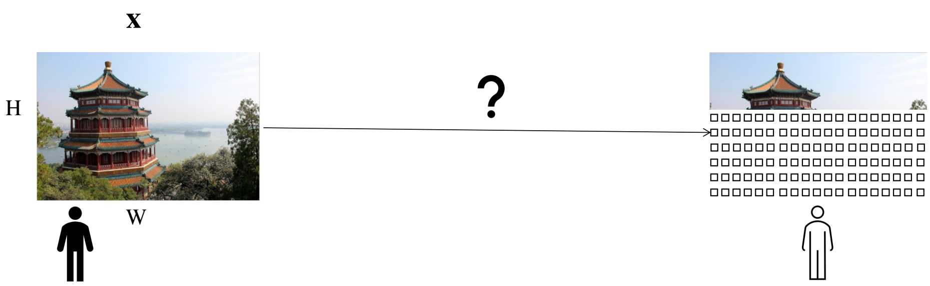

Figure 2: Image compression in colorspace with clustering.

Total for Question 2: 9