Artificial Intelligence and Machine Learning - Unit II
16 September 2022

First and last name, Student ID: _____ Seat: _____

1. Answer the following questions by reporting also the procedure that leads you to the numerical values, not just the values themselves.

   (a) In Eq. (1) left, $\mathbf{X}$ is a design matrix where each row is a sample. What is the dimensionality of the samples in $\mathbf{X}$ and how many samples do we have? Complete the $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ to compute the empirical average and the covariance matrix associated with $\mathbf{X}$. [2]

$$\mathbf{X} = \begin{bmatrix} 2 & 5 \\ -10 & -7 \end{bmatrix} \quad \boldsymbol{\mu} = \begin{bmatrix} \quad & \quad \end{bmatrix} \quad \boldsymbol{\Sigma} = \begin{bmatrix} \quad & \quad \\ \quad & \quad \end{bmatrix} \tag{1}$$

Solution: The number of samples is 2 and the dimensionality is 2. $\boldsymbol{\Sigma}$ is $\frac{1}{N}(\mathbf{X}-\boldsymbol{\mu})^T(\mathbf{X}-\boldsymbol{\mu})$

$$\boldsymbol{\mu} = \begin{bmatrix} -4 & -1 \end{bmatrix} \quad \boldsymbol{\Sigma} = \begin{bmatrix} 36 & 36 \\ 36 & 36 \end{bmatrix} \tag{2}$$

   (b) Referring to transformation $\mathbf{T} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$, compute the area of the ellipsoid that gets generated once you map the unit circle through $\mathbf{T}$ as shown in Fig. 1 *(Hint: start from the area of the unit disk which is πr^2 and remember how to compute the change in volume induced by a transformation)*. Explain how your approach can work also in higher dimensions, not only in 2D. Given a generic transformation $\mathbf{T}$, when it may happen that the ellipsoid is squished down to a line? [3]

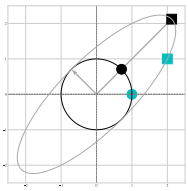

Figure 1: The black round point is sent to the black squared point. The gray round point to the gray squared point.

Solution: We can compute the area by expressing the output area of the ellipsoid as the source area of the unit circle which is π multiplied by the absolute value of the determinant of $\mathbf{T}$. In this case the determinant $|\det(\mathbf{T})|$ is $2 \cdot 2 - 1 \cdot 1 = 3$. Thus the ellipse area will be 3π. This will work in N dimension too since we can still compute the det. of a bigger matrix, provided we know the hyper-volume of the source shape to compute the final hyper-volume. The ellipsoid can be reduced to a line if the transformation matrix $\mathbf{T}$ is not full-rank, in this case at least an eigenvalue will be zero and thus the determinant will be also zero, resulting an a shape with zero hyper-volume in N dimensions.

   (c) You work as a computer scientist for a medical company and you interact with a doctor. He wants to visualize the most prominent variation in 3D scans of skulls. The doctor can give you a set of 3D point cloud of skulls. The point clouds are densely registered and aligned across all samples and given to you as matrix $\mathbf{S} \in \mathbb{R}^{3P \times N}$, where the P is the number of points in each cloud and N is the number of individuals. Describe the technique that helps the doctor in the visualization, describing even with math what you need to implement, how you find the most prominent variation and how you morph the data along the prominent variation. [2]

Solution: You can help the doctor by computing the Principal Components of the point clouds in $\mathbf{S}$. To do that you need to find eigenvectors (the principal components) of the centered covariance matrix of $\mathbf{S}$, ordered by the eigenvalues in descending order. The eigenvector with the highest eigenvalue will tell you how to morph the data to see the most prominent variations as:

$$\mathbf{S}' = \hat{\mathbf{S}} + \alpha \mathbf{U}_i$$

where $\hat{\mathbf{S}}$ is the average skull, α is scalar that you can tune to see the variation and $\mathbf{U}_i$ is the eigenvector associated to the largest eigenvalue.

Total for Question 1: 7

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 the same standard deviation across clusters, 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. [2]

Solution: This can be solved in multiple ways but the simpler is a variance minimizer objective function. The variance to be minimized is the spread of a cluster, for all the clusters. The problem in this loss is that we do not know the cluster assignment a priori. The objective function is minimizing the squared L2 norm of all the points respect to the k centroids as:

$$\min_{\mu,y} \sum_{i}^{N} \left\| \mathbf{x}_i - \boldsymbol{\mu}_{y_i} \right\|^2 \tag{3}$$

where y_i represents the assignment for the point i. To minimize it, we use a two step procedure similar to EM algorithm in which: 1) given centers, we compute the nearest center to a point (assignment step) 2) given the assignments, we recompute the centers.

(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 $\mathbf{x}_3$. [1]

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

Solution: The assignments are:

$$\left[\mathbf{x}_1 \mapsto 2; \quad \mathbf{x}_2 \mapsto 2; \quad \mathbf{x}_3 \mapsto 1; \quad \mathbf{x}_4 \mapsto 1; \quad \mathbf{x}_5 \mapsto 2; \quad \mathbf{x}_6 \mapsto 1;\right]. \tag{5}$$

For example, for point $\mathbf{x}_3 = [1.45, 423]$ we have to compute the L2 distance between the point and all the centers. Then select the id k' of the minimum distance center as:

$$k' = \arg\min_{k} \left\| \mathbf{x}_3 - \boldsymbol{\mu}_k \right\|^2$$

$d(\mathbf{x}_3, \boldsymbol{\mu}_1) \approx 1.23 \quad d(\mathbf{x}_3, \boldsymbol{\mu}_2) \approx 2.25$ thus point 3 belongs to cluster 1.

(c) Given the assignments you have found in Eq. (4), now compute the **update step** and explaining the procedure for your computation. [1]

Solution: We have to compute the mean just by "filtering" the points for a specific assignment. We have the new centers are:

$$\boldsymbol{\mu}_1 = [1.43, 4.18] \quad \boldsymbol{\mu}_2 = [2.28, 1.21]$$

(d) Different clustering methods that we reviewed in our course can handle different shapes of the data density. Fig. 2 offers four different configurations of the data density. Each color indicates a cluster. For each Case: (a) state which clustering algorithm is the more appropriate and (b) report if you need to model the covariance matrix (c) if so, how do you model it, specifying how many parameters you need in the covariance matrix, if you need off-diagonal values (d) if you need a specific covariance matrix for each blob. [3]

Solution: Note when you do clustering you do not have the labels, i.e. do not know the "colors". **Case 1:** GMM with a single parameter (the radius) to model the entire covariance matrix, repeated on the diagonal; different values for each blob. K-means may work as well. **Case 2:** GMM with a covariance matrix with a different value for each axis (the diagonal part) but no need to model off-diagonal components, those are zeros. The matrix can be shared across clusters; **Case 3:** K-means.

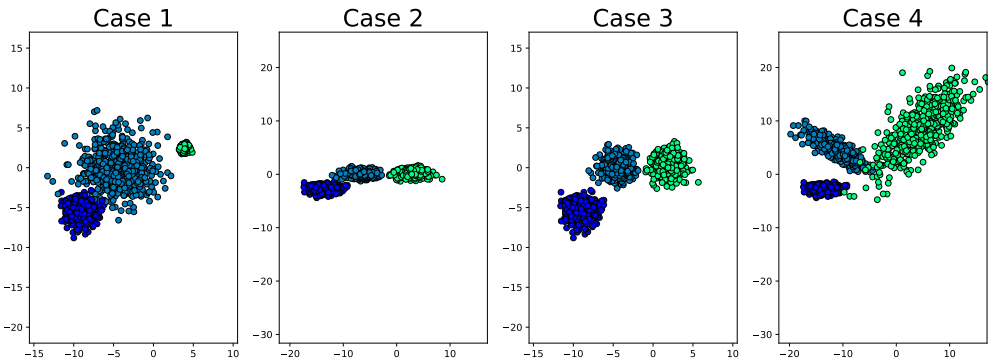

Figure 2: Different shapes of the data density.

It does not model covariances at all. **Case 4:** GMM with full covariance matrix—different values on diagonal components and off-diagonal values; we need a full matrix for each mode.

Total for Question 2: 7

3. We are given an already learned decision tree for binary classification shown in Fig. 3 below. Each square represents a training sample and each circle is a node. Negative points are white squares while positive points are gray squares.

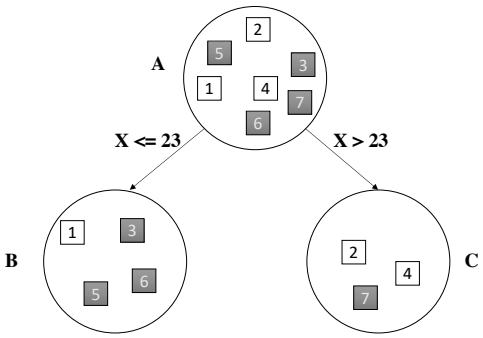

Figure 3: Decision Tree

(a) Compute the Impurity using the Entropy for nodes B and C. What is the Entropy of the Tree? What is the Entropy of the entire training set? [3]

Solution: Entropy of entire training set with no split is:

$$H(\mathcal{S}) = -\sum_{k=\{B,W\}} p_k \log_2(p_k) = -\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7} \approx 0.985$$

which is also that of A. $H(B) = -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4} \approx 0.811$ while $H(C) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} \approx 0.918$. The entropy of the tree is:

$$\frac{4}{7}H(B) + \frac{3}{7}H(C) \approx 0.856$$

(b) Define the Gini Impurity function over a set $\mathcal{S}$ for a generic k-class classification problem and compute the Gini Impurity for the entire training set in Fig. 3. [2]

Solution: For K classification problem, the Gini impurity of a set $\mathcal{S}$ [of labels] is defined as: $H(\mathcal{S}) = \sum_{k=1}^{K} p_k(1-p_k)$ where $p_k = \frac{\sum_i^N \mathbf{1}(s_i==k)}{|\mathcal{S}|}$ and $p_b = \frac{4}{7}$ and $p_w = \frac{3}{7}$ thus $H(\mathcal{S}) = \frac{3}{7}\frac{4}{7}\cdot 2 = 24/49$.

(c) Given the splitting attribute of the decision tree in Fig. 3, assume we have to classify a test sample which feature $\mathbf{x}$ is 25. What is the y' label predicted by the above decision tree, given $\mathbf{x}'$? How much is the probability returned for that prediction, $p(y'|\mathbf{x}')$? [1]

Solution: Looking at the tree in Fig. 3, the test point $\mathbf{x}'$ will be classified as negative (white) with $\frac{2}{3}$ probability since we have 2 white sample over 3 samples in the set C. We arrive at C by the decision attribute $\mathbf{x} > 23$.

Total for Question 3: 6

4. We want to perform some evaluation of a binary classifier that has positive labels +1 and negative labels −1 and the scores are reported as s below the labels in Tab. 2.

y	+1	-1	-1	-1	-1	+1	-1
s	100	-100	-99.5	-99	99	0	-1

Table 2: Labels and unnormalized scores for a binary classifier.

(a) Give a definition of True Positive Rate (TPR) and False Positive Rate (FPR). Given a binary classifier with unnormalized scores s—the higher the score, the more is likely that $y = +1$—compute the ROC curve for the values in Tab. 2 by showing the TPR and FPR in a table. 2

Solution: The True Positive Rate is the ratio between the positive samples that are correctly classified (TP) divided by the number of ground-truth positive we have in the population (P_{gt}). The False Positive Rate is the ratio between the negative samples that are incorrectly classified as positive FP divided by the total number of ground-truth negative in the population N_{gt}.

$$TPR = \frac{TP}{P_{gt}} \quad FPR = \frac{FP}{N_{gt}}.$$

In Tab. 2 N_{gt} is 5 and P_{gt} is 2, so the denominator in the ROC computation will be always the same. We just have to sort the score in descending order and compute the TPR, FPR for each score. We start from maximum score plus 1 to make sure the first point in in the ROC is TPR=0@FPR=0.

thrs	101	100	99	0	-1	-99	-99.5	-100
FPR	$\frac{0}{5}$	$\frac{0}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{4}{5}$	$\frac{5}{5}$
TPR	$\frac{0}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$	$\frac{2}{2}$

(b) Compute the Area Under the Curve (AUC) of the above ROC. 2

Solution: We have all zero in both TPR and FPR until score=100. Then we have to sum $\frac{1}{5} \cdot \frac{1}{2}$ (scores 100 and 99); then we have the same TPR for a FPR that goes from $\frac{5}{5} - \frac{1}{5}$ so add $\frac{4}{5} \cdot \frac{2}{2}$ for a total of 90% AUC.

(c) Let's assume that in Tab. 2, we replace each score as $s' = \text{sign}(s)\sqrt{|s|}$. Will the ROC change? What if we use $s' = \cos(s)$? Motivate why will/will not change for both cases. 2

Solution: $\text{sign}(s)\sqrt{|\cdot|}$ is a strictly monotonic function thus does not change the order of the score and neither ROC will change. Cosine is not monotonic and the order of scores will change as well as the ROC.

(d) Setting the classifier threshold to -0.5, compute the confusion matrix for the values in Tab. 2. For each cell of the confusion matrix, indicate what is the metric computed. 2

Solution: Setting the threshold as -0.5 the scores get binarized as:

$$\begin{bmatrix} y & +1 & -1 & -1 & -1 & -1 & +1 & -1 \\ s & +1 & -1 & -1 & -1 & +1 & +1 & -1 \end{bmatrix} \text{ Confusion matrix } = \begin{bmatrix} TP = \frac{2}{2} & FP = \frac{1}{5} \\ FN = \frac{0}{2} & TN = \frac{4}{5} \end{bmatrix}$$

Total for Question 4: 8

5. We have to solve a linear regression problem, given a design matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$ where N is the number of samples and D is the number of features. We want to regress a value $\mathbf{y} \in \mathbb{R}^N$ by learning parameters $\boldsymbol{\theta}$. Assume no bias. We thus want to solve the following optimization problem.

$$\arg\min_{\boldsymbol{\theta}} \frac{1}{2}\|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 = \arg\min_{\boldsymbol{\theta}} \frac{1}{2}(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})^T(\mathbf{X}\boldsymbol{\theta} - \mathbf{y}) \tag{6}$$

(a) Describe with words a generic problem in which you need to use **regression** instead of **classification**. What is the difference between the two? Referring to Eq. (6), how many parameters do we have to learn? (*Hint: use dimensionality check to see if you answer is correct for the #params.*) [1]

Solution: Regression is needed all the time the output value that you predict is a continuous value, unlike classification that is discrete. An example is that you want to perform some prediction of the temperature for tomorrow where temperature is not quantize but is a continuous value to regress; this is different for example to classify if tomorrow is sunny or cloudy. Another example is classifying if a person is tall or small (classification) versus predicting his height (regression). We have to learn $\boldsymbol{\theta}$ that has the dimension of $D \times 1$ thus is a vector $\boldsymbol{\theta} \in \mathbb{R}^D$.

(b) Derive the closed form solution to minimize the objective function Eq. (6). Show all the steps of how you derive the solution and justify all the steps. [3]

Solution: We have to find the gradient set it to zero and solve for $\boldsymbol{\theta}$.

$$\nabla_{\boldsymbol{\theta}} \frac{1}{2}(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})^T(\mathbf{X}\boldsymbol{\theta} - \mathbf{y}) = 0.$$

$$(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})^T(\mathbf{X}\boldsymbol{\theta} - \mathbf{y}) = (\underbrace{(\mathbf{X}\boldsymbol{\theta})^T}_{\boldsymbol{\theta}^T\mathbf{X}^T}\mathbf{X}\boldsymbol{\theta}) - \underbrace{(\mathbf{X}\boldsymbol{\theta})^T\mathbf{y}}_{\boldsymbol{\theta}^T\mathbf{X}^T\mathbf{y}} - \mathbf{y}(\mathbf{X}\boldsymbol{\theta}) - \mathbf{y}^T\mathbf{y} = \boldsymbol{\theta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} - 2\boldsymbol{\theta}^T\mathbf{X}^T\mathbf{y} + \mathbf{y}^T\mathbf{y}.$$

$$\frac{1}{2}\nabla_{\boldsymbol{\theta}} \underbrace{\boldsymbol{\theta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\theta}}_{\text{quadratic form}} - 2\boldsymbol{\theta}^T\mathbf{X}^T\mathbf{y} + \mathbf{y}^T\mathbf{y} = 2\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} - 2\mathbf{X}^T\mathbf{y} = 0$$

The unknown $\boldsymbol{\theta}$ is what we have to learn. For which we can find $\boldsymbol{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$.

(c) Now assume that it is forbidden to use the closed form solution to minimize Eq. (6). Is there any way to find an approximate solution using another method? If so, explain the algorithm you are going to use from a mathematical and computer science perspective. [3]

Solution: If we cannot use the pseudo-inverse we can still find an approximate solution using iterative Gradient Descent algorithm. This implies optimizing $\boldsymbol{\theta}$ to take small steps (ϵ) in the negative direction of the gradient of the loss respect to $\boldsymbol{\theta}$ by starting from a zero value of $\boldsymbol{\theta}$ and iterating until convergence. In this case the loss function has a ball shape and we are guaranteed to reach near a global minima.

$$\boldsymbol{\theta}' = \boldsymbol{\theta} - \epsilon\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}; \mathbf{X}; \mathbf{y}).$$

So we have to iterate as:

$$\boldsymbol{\theta}' = \boldsymbol{\theta} - \epsilon\mathbf{X}^T(\mathbf{X}\boldsymbol{\theta} - \mathbf{y}).$$

So from a computer science perspective where we have to iterate to find a solution and wrap the code in some sort of loop; while in closed form we just have to invert a matrix using some library (without going into details on the computation of inverting a matrix).

Total for Question 5: 7

You can use this space for writing. Summary for points is at the bottom.

Question:	1	2	3	4	5	Total
Points:	7	7	6	8	7	35
Score:						