Artificial Intelligence and Machine Learning - Unit II
June 23, 2022

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

1. You have to implement some geometrical checks for a machine learning algorithm.

   (a) In Eq. (1) left, $\mathbf{X}$ is a design matrix where each column is a sample. Complete the $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ to compute the empirical average and the covariance matrix associated to $\mathbf{X}$. [½]

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

Solution: The number of samples is 5 and dimensionality is 2. $\boldsymbol{\Sigma}$ is $\frac{1}{N}(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^T$.

$$\boldsymbol{\mu} = \begin{bmatrix} \frac{5}{5} \\ \frac{10}{5} \end{bmatrix} \quad \boldsymbol{\Sigma} = \begin{bmatrix} \frac{134.}{5} & \frac{-122.}{5} \\ \frac{-122.}{5} & \frac{126.}{5} \end{bmatrix} \tag{2}$$

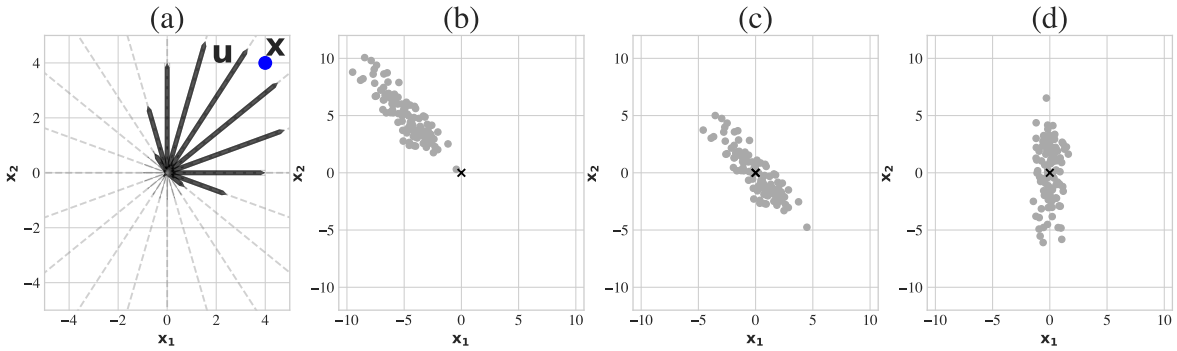


Figure 1: Projection and Point cloud

   (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 $\mathbf{u}$. [3]

Solution: The projection is $(\mathbf{x}^T\mathbf{u})\vec{\mathbf{u}}$. Its length is $\left\|(\mathbf{x}^T\mathbf{u})\mathbf{u}\right\| = \left\|\mathbf{x}\right\||\cos\theta|$ since $||\mathbf{u}||_2 = 1$ and θ is the angle between $\mathbf{x}$ and $\mathbf{v}$. We can maximize it when $\cos\theta = \pm 1$ which means $\mathbf{u}$ is in the same direction of $\mathbf{x}$ ($\theta = \{0,\ \pi\}$). When this happens, the maximum projection length is $\left\|\mathbf{x}\right\|$.

   (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? [1]

Solution: We centered by subtracting the empirical mean of $\mathbf{X}$. We can achieve that with `X = X - X.mean(axis=0)`. Numpy tries to perform broadcasting to match the shape, if the shapes of matrices are not the same.

   (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? [3]

Solution: We extract the rotation matrix $\mathbf{U}$ by solving eigendecomposition of centered covariance matrix and apply it to the point cloud as $\mathbf{U}^T\bar{\mathbf{X}}^T$. In particular, to "make it vertical" (longest principal component on the y axis), the $\mathbf{U}$ last column vector needs to be associated with the highest eigenvalue. After the transformation, the covariance matrix is decorrelated, and will be a diagonal matrix.

Total for Question 1: 7½

2. We have to build a **Gaussian Mixture Model (GMM)**, from a training set of data. Each data point lives in a space such as $x_1 \in \mathbb{R}^1$. The assignment z of the GMM are known and given to you already as $z \in \{0, 1, 2\}$, for each training point—see Tab. 1. Assume the estimate for GMM is maximum likelihood (MLE).

x_1	11	3	-1	10	-5	-6	-4	2	4	1	-2	-3
z	2	1	0	2	2	1	2	0	1	0	0	2

Table 1: Training set of a GMM with assignments.

(a) How many modes does the GMM described above have? Please, motivate your answer. [1/2]

Solution: The modes are three given that z takes values in the set $Z = \{0, 1, 2\}$; a data point can be associated to mode 0 or mode 1 or mode 2. Thus number of modes is $|Z| = 3$.

(b) Give a definition and mathematically describe what is the probability density function used in GMMs. Write down the name of the distribution if your recall it. [2]

Solution: The density function is a linear combination of Gaussian distributions, which is called Mixture of Gaussians (MoG). The linear combination coefficients—mixing coefficients—indicates also a categorical distribution $\boldsymbol{\pi}$ so if the modes are K, we have that $\sum_{k=1}^{K} \boldsymbol{\pi}_k = 1$ and $\pi_k \geq 0$. Each Gaussian distribution is specified by its parameters $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. Thus the final pdf is:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \boldsymbol{\pi}_k \cdot \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{3}$$

(c) Compute the density function given the training data in Tab. 1 *(Hint: to compute it, you need just to specify which distribution you have in GMM, and compute the parameters of those distribution given the training set. It is OK to say: it distributes as distribution A with parameters indicated by the B and C and offer numerical values for B and C.)* [2 1/2]

Solution: The density is defined by the MoG. Once you know $\boldsymbol{\pi}$ and each param of the Gaussian we are done. $\boldsymbol{\pi}$ is simply $\frac{\sum_i^N \mathbf{1}(z==k)}{\sum_i^N 1}$ in our case $\boldsymbol{\pi} = [4/12, 3/12, 5/12]$. For each Gaussian component, we can estimate it with MLE restricting the points to select just x_1 given $z == k$ and then computing 1D mean and variance.

z	μ	σ^2
0	0.0	2.5
1	0.333	20.222
2	1.8	50.96

(d) Each mode of the GMM models $p(\mathbf{x}|z)$. Given $\mathbf{x}'$ as new unseen input, how could you compute the probability $p(z = 0|\mathbf{x}')$? [2]

Solution: We indicate $p(z = 0|\mathbf{x}')$ for notation clarity as $p(z_0|\mathbf{x}')$. If we have a new $\mathbf{x}'$, the GMM models $p(\mathbf{x}|z_0) \sim \mathcal{N}(\mu_0, \Sigma_0)$ where the parameters μ_0, Σ_0 are those associated to the mode $z = 0$. We want to compute $p(z_0|\mathbf{x}')$, so we can use Bayes Theorem as:

$$p(z_0|\mathbf{x}') = \frac{p(\mathbf{x}'|z_0)p(z_0)}{\sum_{k\in\{0,1,2\}} p(\mathbf{x}'|z_k)p(z_k)}$$

(e) After you have fit the GMM, let's say that you want to sample a new point from the generative model behind the GMM. What are the steps necessary to sample from your estimated GMM? *(Describe how sampling works step by step, which distributions you sample from using which technique.)* [1]

Solution: First we sample the index of the Gaussian from the prior $\hat{k} \sim \text{Cat}(\boldsymbol{\pi}_k)$, we can use inverse transform sampling to do that. After that we have the $\hat{k}$ of the Gaussian, we sample $\mathbf{x} \sim \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\hat{k}}, \boldsymbol{\Sigma}_{\hat{k}})$.

Total for Question 2: 8

3. Given the training points below for $y \in \{0, 1\}$ binary classification:

$$(x_1 = 1/4; y_1 = 1) \quad (x_2 = 1/2; y_2 = 0) \quad (x_3 = 3/4; y_3 = 1)$$

(a) Determine the output of a **K Nearest Neighbour (K-NN)** classifier for all points on the interval $0 \leq x \leq 1$ using: • 1-NN and • 3-NN. *(You have to write explicitly how the [0,1] interval is classified though you can augment your answer with drawing if you want)* [3]

Solution:

1NN:

$$\begin{array}{ll} 0 \leq x < \frac{3}{8} & \rightarrow y = 1 \\ \frac{3}{8} < x \leq \frac{5}{8} & \rightarrow y = 0 \\ \frac{5}{8} < x \leq 1 & \rightarrow y = 1 \\ x = \frac{3}{8}, x = \frac{5}{8} & \rightarrow \text{equally likely} \end{array}$$

3NN:

$$0 \leq x \leq 1 \rightarrow y = 1$$

since it is majority vote of all points.

(b) Assume you want to regress continuous values—thus $y \in \mathbb{R}$. The regressed output is the mean of the **K Nearest Neighbour (K-NN)** of a test point. Determine the ouput on the interval $0 \leq x \leq 1$ using the same training data above for $K = 2$. [1 ½]

Solution: $0 \leq x \leq 1 \rightarrow y = \frac{1}{2}$ since on the left half, the mean will be between 0 and 1; on the right half, the mean will be between 1 and 0.

(c) Your mate tells you that he/she got perfect (100%) training accuracy by classifying the training data with K-NN when $K = 1$. Also, adds that increasing K is a bad idea because training accuracy drops. What is happening and how would you reply? What is the effect of increasing K? *(Explain clearly all details as much as possible)* [1 ½]

Solution: If you apply k-NN with $k = 1$ on the training set itself, you are comparing each point to itself, so training accuracy will be always 100% no matter how points distribute. The method overfits with $k = 1$ and increasing $k \gg 1$ may prevent overfitting since induce **smoothness** in the response. Evaluating the performance of a ML algorithm on the training set is flawed and performance can be observed on a validation set. More in general, the following holds for k-NN:

$k = 1$	overfit/variance problem
$k = \#\text{samples}$	underfit/bias problem

always predicts the most frequent class in the training

Total for Question 3: 6

4. We are given a training set, where each attribute $\mathbf{x}$ describes if a patient 1) had cough 2) had soar throat 3) had fever; the $y \in \{\text{C, F, H}\}$ indicates having contracted COVID-19 (C); normal flu (F) or being healthy (H).

Patient	Coughing	Sore Troath	Fever	$y = \{\text{Covid, Flu, Healthy}\}$
$\mathbf{x}_1$	Y	Y	Y	C
$\mathbf{x}_2$	Y	N	Y	F
$\mathbf{x}_3$	Y	N	N	H
$\mathbf{x}_4$	Y	Y	N	H
$\mathbf{x}_5$	N	Y	Y	C
$\mathbf{x}_6$	N	N	Y	F

Table 2: Training set for disease classification.

(a) Using the training data above, construct a **decision tree** for the 3-class classification problem above. Use the Information Gain (IG) with **entropy** as impurity function, as the decision criterion to select which attribute to split on. Show your calculations for the IG for all possible attributes for just the first split. Draw the configuration of the resulting tree. [3]

Solution: Entropy of entire set with no split is:

$$H(\mathcal{S}) = -\sum_{k=\{C,F,H\}} p_k \log_2(p_k) = -\log_2 \frac{2}{6} \approx 1.58$$

I developed only for splitting on Fever the others imply the same process:

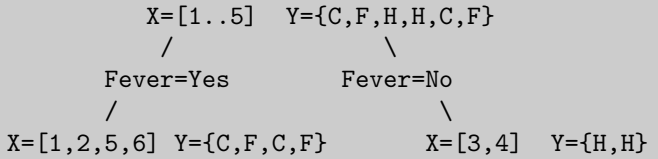
$$H(\mathcal{S}|\text{Fever}) = \frac{4}{6} H(\mathcal{S}|\text{Fever=Yes}) + \frac{2}{6} H(\mathcal{S}|\text{Fever=No}).$$

If we split of Fever=No, then $\mathcal{S} = \{H, H\}$ and Fever=Yes, then $\mathcal{S} = \{C, F, C, F\}$ thus:

$$H(\mathcal{S}|\text{Fever=Yes}) = -2 \cdot \frac{2}{4} \log \frac{2}{4} \quad H(\mathcal{S}|\text{Fever=No}) = -\frac{2}{2} \log \frac{2}{2} = 0$$

$$IG(\text{Fever}) = H(\mathcal{S}) - H(\mathcal{S}|\text{Fever}) \approx 0.91$$
$$IG(\text{Cough}) = H(\mathcal{S}) - H(\mathcal{S}|\text{Cough}) \approx 0.2$$
$$IG(\text{Sore}) = H(\mathcal{S}) - H(\mathcal{S}|\text{Sore}) \approx 0.67$$

The highest IG is when we split on **Fever**.



(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 trainng set in Tab. 2. [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 $p_c = p_h = p_f = \frac{2}{6}$ thus $H(\mathcal{S}) = \frac{2}{6}\frac{4}{6} \cdot 3 = 4/6$

(c) Let us assume that you have pairs of points as $\{x_i, y_i\}_{i=1}^N$ where y is a continuous value in $\mathbb{R}$ from a unknown function $y = f(x)$. Is it possible to learn the function $f(\cdot)$ with a tree? Can you tell how the tree can approximate f? Which loss function are you going to minimize? [1]

Solution: Yes, it is possible and the tree will approximate the function $f(\cdot)$ by learning a piece-wise constant function where the constant value correspond to the average $\bar{y}$ of the set of points considered at that level of the tree. The impurity function changes to the the variance computed as $\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}(y_i - \bar{y})^2$. The more we grow the tree the better we fit the datapoints yet we may not approximate $f(\cdot)$ well "outside" of the datapoints (overfit).

Total for Question 4: 6

5. You work as a data scientist for `VisionGrad` an hot startup working with automatic differentiation. Your job is to analyze the computational graph shown in Fig. 2.

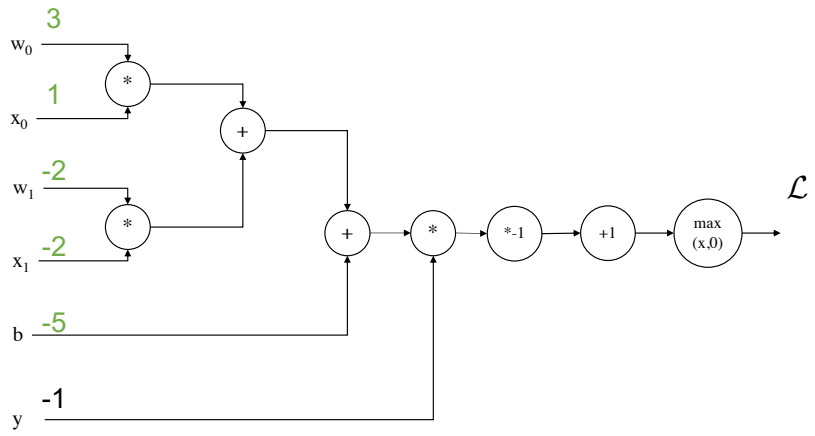


Figure 2: Computational Graph

(a) Given the graph, write down the function in a vectorized form, that implements the graph. Assume $y \in \{1, -1\}$. In which algorithm we have encountered a loss function similar to one in the graph above? Explain what the loss does with as much as details as possible for each single step. [1]

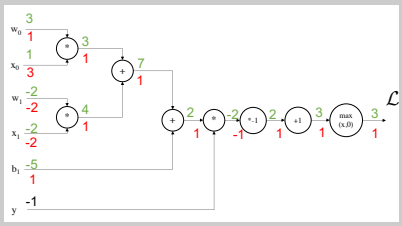
Solution: We assume $y \in \{-1, +1\}$. The loss function is:

$$\max\big(-y(\mathbf{w}^T\mathbf{x} + b) + 1, 0\big)$$

and is the Hinge Loss that can be used for training Support Vector Machines (SVMs). The loss firstly computes the distance to the hyper-plane $\mathbf{w}, b$. The objective is to have $y(\mathbf{w}^T\mathbf{x} + b) \geq 1$ for all samples. If $y(\mathbf{w}^T\mathbf{x} + b) \gg 1$, then $\mathbf{x}$ already satisfies the constraint, it will not incur any loss since the $\max(z, 0)$ will clip the loss to zero; otherwise it will incur a loss of $1 - y(\mathbf{w}^T\mathbf{x} + b)$.

(b) Fill in the graph to compute the forward pass and backward pass: compute the derivatives over all the inputs (e.g. $\frac{\partial \mathcal{L}}{\partial w_0}, \frac{\partial \mathcal{L}}{\partial w_1}, \frac{\partial \mathcal{L}}{\partial x_0}$, etc.), showing also the intermediate values of those. Write the forward pass value $f(x)$ above each gate, write $\frac{\partial f(x)}{\partial x}$ below each gate. Skip computation on y. [3]

Solution:



(c) Given the partial derivatives you computed on the inputs, let's assume that you can perturb only a single input with $+1$ or -1. Which input do you perturb to decrease the loss the most? Please, specify if you are adding or subtracting one. Given your perturbation, can you compute the value of the new loss after your perturbation yet **without forwarding the new input in the graph**? If yes, motivate how you can do it and write the new value of the loss; if no, explain why. [3½]

Solution: We see that the derivative with the largest magnitude is $\frac{\partial \mathcal{L}}{\partial x_0} = 3$, so we have to select x_0. If we add a unit to $x_0 \leftarrow x_0 + 1$ this will make the loss change by $+3$ as $\mathcal{L} \leftarrow \mathcal{L} + \frac{\partial \mathcal{L}}{\partial x_0} = \mathcal{L} + 3 = 3 + 3 = 6$. This holds only when we perturb a single input from the definition of derivative $\frac{\partial \mathcal{L}}{\partial x_0} = \lim_{\epsilon \to 0} \frac{\mathcal{L}(w_0; w_1; \mathbf{x_0} + \epsilon; x_1) - \mathcal{L}(w_0; w_1; \mathbf{x_0}; x_1)}{\epsilon}$. In our case $\epsilon = \pm 1$ thus

$$\mathcal{L}(w_0; w_1; \mathbf{x_0} + \epsilon; x_1) = \mathcal{L}(w_0; w_1; \mathbf{x_0}; x_1) + \epsilon \frac{\partial \mathcal{L}}{\partial x_0}.$$

The two term on the right are computed already, and we want to **decrease** the loss, we have to go in the other direction of x_0 and add -1 to x_0, which yields the new $x_0 = 0$. Without using the graph $\mathcal{L} \leftarrow \mathcal{L} - \frac{\partial \mathcal{L}}{\partial x_0} = 3 - 3 = 0$.

Total for Question 5: 7½

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

Question:	1	2	3	4	5	Total
Points:	7½	8	6	6	7½	35
Score:						