**Artificial Intelligence and Machine Learning - Unit II**
30 January 2024

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 row is a sample. How many samples are present in the design matrix $\mathbf{X}$? Complete the $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ to compute the empirical average and the covariance matrix associated with $\mathbf{X}$. [1]

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

   (b) We have a point cloud $\{\mathbf{x}_i\}_{i=1}^{N}$ of data points that live in a 3 dimensional space. They are generated from three unknown multivariate normal distributions $\mathcal{N}_i \quad i \in 1, 2, 3$. In addition you also have a list of indexes z of length $N$. You can see z as a map in which you insert the index of a point and gives you the index of associated Gaussian. `z[10] = 2` means the point 10 is associated with second Gaussian. See Fig. 1 for an example of the 3D point cloud, each color indicates a specific Gaussian. [1]

   ◇ Assume that all the ellipsoid you have are axis-aligned, what can you say about the covariance matrix of all of them? Do you have to compute the full covariance matrix or can you compute something else quicker and still assemble the covariance matrix?

   ◇ Describe the mathematical procedure to make all the 3 ellipsoids become three unit spheres centered at the origin of the space.

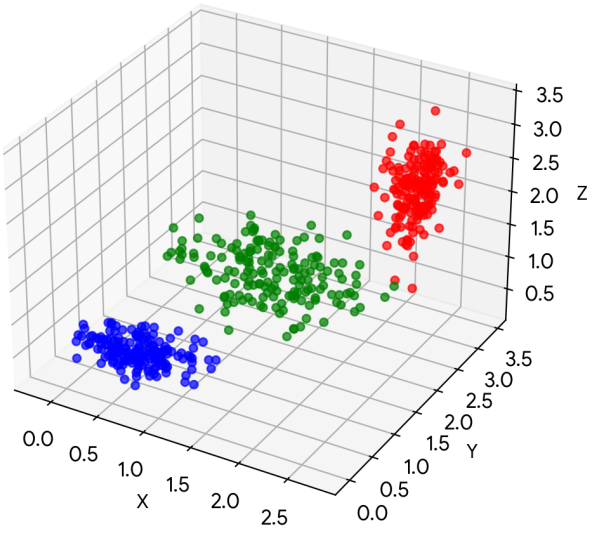


**Figure 1:** (a) A 3D point cloud generated by 3 Gaussian. Z allows you to know which point belong to which Gaussian.

   (c) Your friend gives you a matrix of the principal components $\mathbf{U} \in \mathbb{R}^{k \times D}$, where $D$ is the original dimension of the data. Along with a mean vector $\boldsymbol{\mu} \in \mathbb{R}^{1 \times D}$. You also notice that $k \ll D$. She also gives you another matrix of $N$ "compressed" data as $Y \in \mathbb{R}^{N \times k}$ and tells you "the original data lives in low-dimensional subspace". [1]

   ◇ Explain what is the role of each row of $\mathbf{U}$.

   ◇ Can you recover the *original, exact same, uncompressed* data? If you yes, explain how. if you replied no, can you then recover an approximation of the data? If so, describe how you can do it.

Total for Question 1: 3

2. We are in the $i$-th step of the Expectation-Maximization (EM) for learning the parameters of a GMM. Let us assume the **maximization** part just finished. The parameters of the GMM and the training points $x$ are given in Tab. 1. Assume the estimate for GMM is maximum likelihood.

| | $z_1$ | $z_2$ | $z_3$ |
|---|---|---|---|
| $\mu$ | 1.80 | 1.13 | 1.05 |
| $\sigma^2$ | 46.81 | 6.91 | 3.31 |
| $\pi$ | 0.4 | 0.2 | 0.4 |

**Table 1:** Training set of a GMM with parameters.

(a) Write the density of the GMM with equations using the parameters in Tab. 1, for a point $x$. [½]

(b) You are given a point cloud generated by a multi-variate Gaussian pdf, with covariance matrix $\mathbf{\Sigma}$ and center $\boldsymbol{\mu}$, that is $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$. You generate with random seed set as `random.seed(0)`. Then you have another point cloud sampled as $\mathbf{y}_i \sim \mathcal{N}(\boldsymbol{\mu} + \boldsymbol{\theta}, \mathbf{R\Sigma})$ where $\mathbf{R}$ is a rotation matrix and before sampling this new point cloud, you set the same seed as `random.seed(0)`. What can you say about the approximate volume of the two point clouds? [1]

(c) Assume you have the GMM in Tab. 1. You are also given a new point $\mathbf{x}$ and the responsibilities $\boldsymbol{\gamma}$ of this point $\mathbf{x}$ as $\boldsymbol{\gamma} = [0.2, 0.4, 0.4]$. Explain how to compute the probability of $\mathbf{x}$ under a given Gaussian, i.e. explain and then compute numerically $p(\mathbf{x}|z_k)$. $\forall k \in 1, \ldots, 3$, assuming that under the GMM $p(\mathbf{x}) = 45\%$. You may need only *some* parameters in Tab. 1. [2½]

Total for Question 2: 4

3. Considering the k-NN algorithm, answer the following questions.

(a) Fig. 2 shows the 2D feature space for two sets of datapoints belonging to the gray triangle label or to the black square label. Taking into consideration the query point star $\star$, list all possible values of $k$ that you can set in the k-NN algorithm so that the query star $\star$ will be classified as black square, using the $\ell_2$ norm as distance function. 1

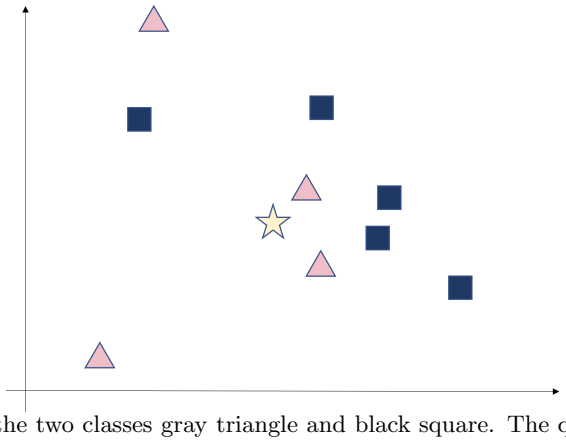

**Figure 2:** Feature vector for the two classes gray triangle and black square. The query point is indicated with the star $\star$.

(b) Assume that you have feature vectors where the number of dimension $D$ is very high, you also know that there is a number of features $D_{\text{ok}}$ that are very informative for the tasks of classification; the rest of the features $D_{\text{ko}}$ are not informative. Assuming $D = D_{\text{ok}} + D_{\text{ok}}$ and $D_{\text{ok}} \ll D_{\text{ko}}$. Describe if this can be a problem for a $k$-NN classifier and, if so, how to improve the $k$-NN classifier. 1

(c)
- Between a k-NN classifier and the perceptron, which can model more complex decision boundaries?
- If you notice that a classifier has very fragmented decision boundary, does the classifier suffer from underfitting or overfitting?
- Let's say that you want to make the decision boundary of a k-NN classifier less fragmented and more smooth: are you going to increase or decrease K? Explain your answer.

1

Total for Question 3: 3

4. We want to perform some evaluation of a binary classifier.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $y$ | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| s | 0.2 | 1 | 0.3 | 0.1 | -1 | -0.2 | 0 |

**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 correlates with $y$—compute the ROC curve for the values in Tab. 2 by showing the TPR and FPR in a table. [1½]

(b) Compute the Area Under the Curve (AUC) of the above ROC. [1]

(c) Let's assume that in Tab. 2 we multiply all the score by $-0.5$. Is the ROC going to change? Do we have to recompute it? Explain what happens to the ROC. [½]

(d) Bob works for `IseekU`, a biometric company using AI, and she is happy since she developed a "perfect" classifier: it achieves 99.9% AUC in the validation set over $100K$ samples. Alice says "it is ready to be employed in practice since it will never generate false alarm". What would you tell Alice? What Alice should measure if the company wants a quota "$X$" on the false alarms? [½]

Total for Question 4: 3½

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. You work in the condition for which $D \ll N$. We want to regress a value $\mathbf{y} \in \mathbb{R}^N$ by learning parameters $\boldsymbol{\theta}$. You also want to learn a bias term too. We thus want to find the "best" $\hat{\boldsymbol{\theta}}$ so that $\mathbf{y} \approx \hat{\mathbf{y}}$ where $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\theta}}$. "best" indicates a least square sense.

(a) 1½

- What is the dimensionality of $\hat{\boldsymbol{\theta}}$?
- What is the total number of parameters you have to learn?
- Explain how you can recover $\hat{\boldsymbol{\theta}}$ in closed form.
- After you recovered the $\hat{\boldsymbol{\theta}}$, let us say that you want to find out the training point that has the highest error in a least square sense. What do you do?

(b) 2

You work in a company that implements `autodiff` a software for automatic differentiation. You have to implement a new node that will be added to the suite usable in a graph which performs:

$$z = f(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{y} + \mathbf{x}^\top \mathbf{x}. \tag{3}$$

The node computes as the above equation does not learn any parameter. $\mathbf{x}$ is a vector with $D$ components, same as $\mathbf{y}$.

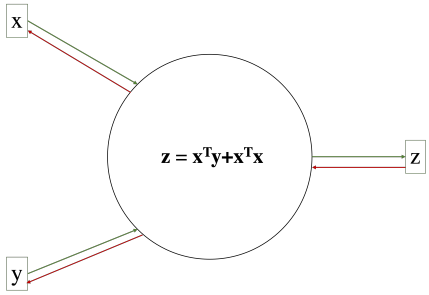


**Figure 3:** Node of the computational graph

- What is the type of derivative that you have to compute given $f(\mathbf{x}, \mathbf{y})$ as defined in Eq. (3) considering the first derivative over the input?
  - ◯ Hessian
  - ◯ Jacobian
  - ◯ Gradient
  - ◯ High-order tensor
- Write the derivation of $\frac{\partial \mathbf{z}}{\partial \mathbf{x}}$ and $\frac{\partial \mathbf{z}}{\partial \mathbf{y}}$.
- Now consider to implement this with python code in a class named `MyNode`, complete the forward pass and backward pass methods below.

```python
class MyNode:
    def forward(x, y):
        # complete the code
        return x.T@y+x.T@x

    def backward(dLdz):
        # complete the code and
        # explain what is dLdz.
        return # describe the items to return and how many
```

**Listing 1:** The node described above

Total for Question 5: 3½

You can use this space for writing. The summary of points is at the bottom.

| Question: | 1 | 2 | 3 | 4 | 5 | Total |
|---|---|---|---|---|---|---|
| Points: | 3 | 4 | 3 | $3\frac{1}{2}$ | $3\frac{1}{2}$ | 17 |
| Score: | | | | | | |