Assignment 4 - Data Science
Theoretical Questions

Mohammad Hossein Basouli

May 22, 2025

Question 1

(a)
1. **Sensors & Preprocessing**: Use sensors to receive signals, and apply preprocessing to improve data quality.
2. **Feature Extraction**: Extract useful features from raw feature vectors.
3. **Learning Algorithm**: The utilized algorithm to learn the pattern from the data.
4. **Teacher**: Tells the class that each data point belongs to.

(b)
1. **Teacher**: The teacher provides information about the hidden state of the data points, to the learning algorithm.
2. **Learning Algorithm**: Takes the data points (which have, **Preprocessing** & **Feature Extraction** applied on them already) along with the information provided by the teacher by the hidden state of the data points, to learn the pattern that underlies in the dataset.

(c) In the **Training Mode**, we first apply **Preprocessing** & **Feature Extraction/Selection** on the training samples, and then feed them to the **Learning Algorithm**, to try to learn the pattern that underlies in the dataset. But in **Classification Mode**, we will apply the utilized **Preprocessing** & **Feature Extraction/Selection** in the **Training Mode**, to the test samples and then feed it to the **Classification** model that has been obtained from the **Learning Algorithm**, in order to classify each test sample.

Question 2

(a) In **Classification**, we try to predict the class that the sample belongs to, which is a discerete number. But in **Regression**, we predict a continious number y, which we have learnt it's relationship with a set of feature vectors X before.

(b) I have already determined the type of output for each one. Now we get to showing one application for each of them:

- **Classification**: Optical character recognition
- **Regression**: Prediction of the amount of sales for a icecream shop in the next month.

Question 3

(a) It represents a vector of observations (measurements).

(b) It determines the class that the given sample belongs to, and could be used in order to learn the pattern between the feature vectors and the hidden states of some given samples.

Question 4

(a) It stores the feature vectors along with hidden states for all of the training samples, and then given a new sample X, it will lookup the stored training sample X_i such that $X_i = X$ and then it will output the hidden state of the sample X_i as the predicted hidden state for X.

(b) Some images might not be recognized.

Question 5

(a) The area/region that all of the points in it, belong to the sample class, form a **Decision Region**.

(b) The boundary between different **Deicision Regions** is called **Decision Boundary**. Illustration is given in Figure 1

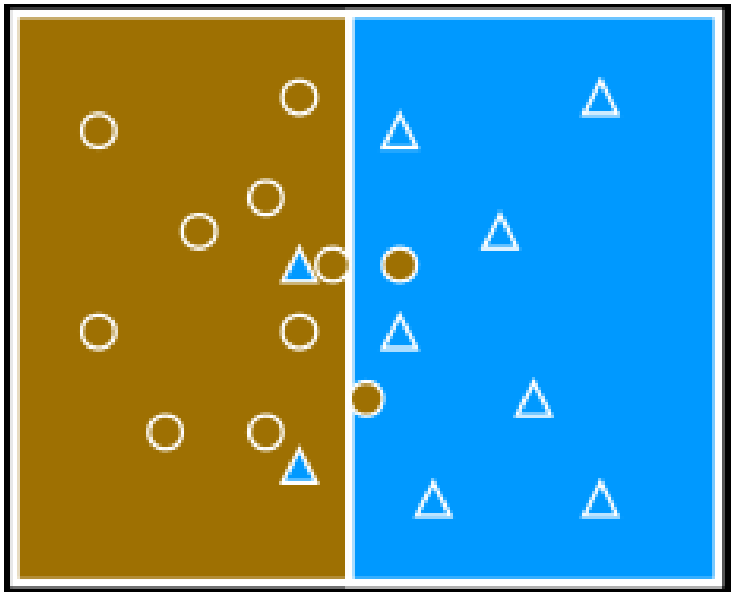

Figure 1: Illustration of The Decision Boundary. The white line on the middle, that separates the two classes, illustrates the decision boundary in this problem.

Question 6

note: I have used the GPT to write out formula for ***Bayesian Theorem*** *and explain the terms in it, but i'm fully aware of what is going on and what's the logic behind all these.*

(a) Bayes' Theorem in the Context of Classification:

$$P(C_k \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid C_k) \cdot P(C_k)}{P(\mathbf{x})}$$

Where:

- C_k is the k-th class.
- $\mathbf{x}$ is the feature vector (observation).
- $P(C_k \mid \mathbf{x})$ is the posterior probability of class C_k given observation $\mathbf{x}$.
- $P(\mathbf{x} \mid C_k)$ is the likelihood of observing $\mathbf{x}$ given class C_k.
- $P(C_k)$ is the prior probability of class C_k.

- $P(\mathbf{x})$ is the evidence or marginal likelihood.

(b) Decision Rule for Classification: Assign observation $\mathbf{x}$ to the class with the highest posterior probability:

$$\hat{C} = \arg\max_{C_k} P(C_k \mid \mathbf{x})$$

Using Bayes' theorem, this becomes:

$$\hat{C} = \arg\max_{C_k} [P(\mathbf{x} \mid C_k) \cdot P(C_k)]$$

Since $P(\mathbf{x})$ is constant for all classes, it can be omitted in the maximization.

(c) Explanation of Terms in Bayes' Theorem:

- **Prior** ($P(C_k)$): The prior probability represents our initial belief about the probability of class C_k before observing any data. It reflects how common or likely a class is in general.
- **Likelihood** ($P(\mathbf{x} \mid C_k)$): The likelihood is the probability of observing the data $\mathbf{x}$ given that the class is C_k. It models how probable the observed features are under the assumption that the data comes from class C_k.
- **Posterior** ($P(C_k \mid \mathbf{x})$): The posterior probability is the updated probability of class C_k after observing the data $\mathbf{x}$. It combines the prior and the likelihood to give a more informed estimate.
- **Evidence** ($P(\mathbf{x})$): The evidence (also called the marginal likelihood) is the total probability of observing the data $\mathbf{x}$ across all possible classes. It serves as a normalization constant to ensure the posterior probabilities sum to one:

$$P(\mathbf{x}) = \sum_j P(\mathbf{x} \mid C_j) \cdot P(C_j)$$

Question 7

(a) Because when we want to predict posterior probability $P(C_k|x)$ for class k, we need to calculate $likelihood = P(x|C_k)$ for each class k, and that requires us to be able to estimate PDF.

(b) Parametric estimation is when the density function is known but it's parameters are unknown. Non-parametric is when both are unknown. Examples for each:

- **Parametric**: Maximum Likelihood Estimation.
- **Non-parametric**: Parzen Windows, K-NN, etc.

Question 8

(a) Each of the colors, grey and white, form a separate decision region, and the boundary between these two regions is decision boundary.

(b) There will be multiple cases:

- **Overfitted Model**: If the utilized model is overfitted on the training data, the decision regions will be smalled and the decision boundary will have a jagged, irregular pattern.
- **Moderate/Underfitted Model**: If the model is Moderate or Underfitted on the training data, the decision boundaries would be smoother.

Question 9

(a) It finds the first nearest neighbor of a given test sample, and then it will the test sample, to the class of that neighbor.

(b) It doesn't require any training, it just memorizes the training samples, and then it will find the most similar training sample, to a given unseen test sample, and assign the class of that similar training sample to it.

(c) The choice of metric could significantly affect the classifier's accuracy. e.g. if we take **Euclidian Distance** between the two signals given in the Figure 2, we will possibly get a really high number, meaning that these two signals are very disimilar, but if fact these two signals might convey a similar behavior in our point of view; e.g. we might just care about whether these two signals go up and down together or not. Thus using a distance metric such as **Correlation** would be better for this case.

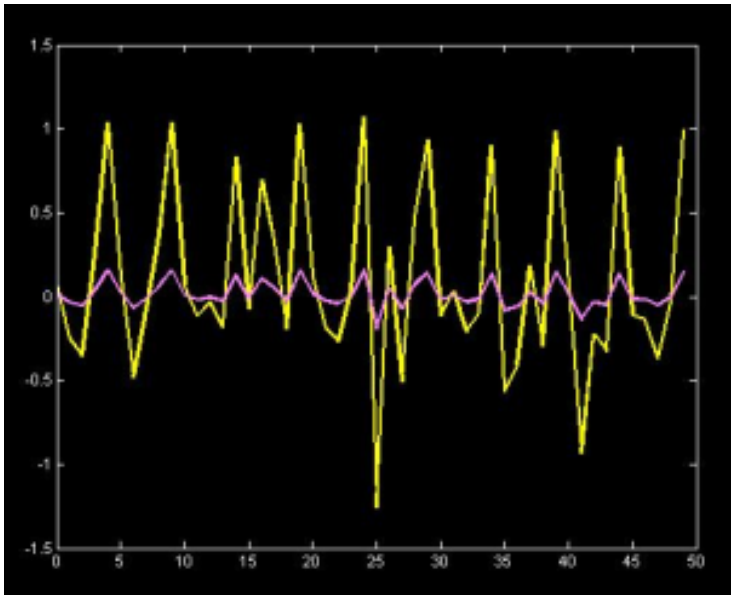

Figure 2: Two signals that their **Euclidian Distance** is so high but their **Correlation** about 1.

Question 10

*note: I have used the GPT to write out the steps performed in **Parzen Windows** method as well as it's formula, but i'm fully aware of why it works like this or what's the logic behind the fomrula for estimating the PDF.*

(a) Describe the **Parzen Window** method for estimating a PDF: **How it works**:

- Place a *kernel/window function* (e.g., mostly **Gaussian**) centered at each data point.
- Each kernel has a fixed **bandwidth parameter** h, which controls its spread.
- The estimated PDF at any point x is the sum of all kernel values at x, normalized by the number of data points.

Mathematically:

$$\hat{p}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

where:

- K is the kernel function,
- h is the window width (bandwidth),
- d is the number of dimensions,
- n is the number of samples.

(b) How does k-NN PDF estimation differ from Parzen? In **k-NN PDF estimation**, the volume around a point is *adjusted*, rather than fixing the window size. **Key differences**:

- **Parzen:** Fixed window size h, variable number of points inside.
- **k-NN:** Fixed number of neighbors k, variable volume enclosing those neighbors.

Mathematically:

$$\hat{p}(x) = \frac{k}{nV_k(x)}$$

where:

- $V_k(x)$ is the volume containing the k nearest neighbors of x,
- n is the number of samples.

(c) What are the challenges in selecting parameters h or k?

- If we choose a very large h we might lose the **Local Density Estimation** property, which essentially says that the points closer to each center should have a much higher impact than those which are far away from it. (or maybe we should neglect the impact of those which are far apart from the center.)
- If we choose a very small h we might lose a significant amount of information, provided by the points which are even a bit relatively further away from the center.
- Chosing a very large k is leads to some issue similar to chosing a very large h, which has been explain already.
- Chosing a very small k is leads to some issue similar to chosing a very small h, which has been explain already.

Question 11

(a) Is a function that provides us with a way to make a distinction between the samples of multiple classes.

(b) Discriminant Function for Multivariate Gaussian Classes:

Given a multivariate normal distribution for class ω_i with mean vector $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$, the class-conditional density is:

$$p(\boldsymbol{x} \mid \omega_i) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_i|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right)$$

The discriminant function $g_i(\boldsymbol{x})$ used in Bayesian classification is:

$$g_i(\boldsymbol{x}) = -\frac{1}{2}\ln|\boldsymbol{\Sigma}_i| - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i) + \ln P(\omega_i)$$

Where:

- $\boldsymbol{x} \in \mathbb{R}^d$ is the feature vector
- $\boldsymbol{\mu}_i$ is the mean vector of class ω_i
- $\boldsymbol{\Sigma}_i$ is the covariance matrix of class ω_i
- $|\boldsymbol{\Sigma}_i|$ is the determinant of the covariance matrix
- $P(\omega_i)$ is the prior probability of class ω_i

Question 12

(a) Explaining how the EM algorithm is used to estimate parameters in a Gaussian mixture: The **Expectation-Maximization (EM)** algorithm is used to estimate parameters of a **Gaussian Mixture Model (GMM)** — a probabilistic model that represents the data as a mixture of several Gaussian distributions.

Let the parameters of the model be:

$$\theta = \{\pi_k, \mu_k, \Sigma_k\} \quad \text{for } k = 1, 2, \ldots, K$$

where:

- π_k is the mixing coefficient of component k,
- μ_k is the mean of component k,

- Σ_k is the covariance matrix of component k,
- $\mathcal{N}(x_i \mid \mu_k, \Sigma_k)$ is the Gaussian density function.

EM Algorithm Steps:

(a) **Initialization:** Choose initial values for π_k, μ_k, Σ_k.

(b) **E-step (Expectation):** Compute the *responsibilities*:

$$\gamma_{ik} = \frac{\pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i \mid \mu_j, \Sigma_j)}$$

(c) **M-step (Maximization):** Update the parameters:

$$\pi_k^{\text{new}} = \frac{1}{n} \sum_{i=1}^{n} \gamma_{ik}$$

$$\mu_k^{\text{new}} = \frac{\sum_{i=1}^{n} \gamma_{ik} x_i}{\sum_{i=1}^{n} \gamma_{ik}}$$

$$\Sigma_k^{\text{new}} = \frac{\sum_{i=1}^{n} \gamma_{ik} (x_i - \mu_k^{\text{new}})(x_i - \mu_k^{\text{new}})^T}{\sum_{i=1}^{n} \gamma_{ik}}$$

(d) **Repeat** E-step and M-step until convergence.

(b) Why might ensemble methods outperform a single classifier?

- **Robustness:** Ensembles are less sensitive to noise and outliers.
- **Improved Accuracy:** By leveraging diversity among base learners, ensembles often achieve higher predictive accuracy.