DDA3020 Assignment 4 Student ID: 121090429 Ou Ziyi 欧子裔

Written Problems

1. Gaussian Mixtrue Model: $p(x) = \sum_{i=1}^{n} \pi_i \cdot \frac{1}{\sqrt{(2\pi)^d \cdot |\Sigma_i|}} \cdot \exp(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i))$

Let $\theta = \{\pi, \mu, \Sigma\}$, $\pi = \{\pi_1, \cdots, \pi_n\}$, $\mu = \{\mu_1, \cdots, \mu_n\}$, $\Sigma = \{\Sigma_1, \cdots, \Sigma_n\}$

The log-likelihood function is:

$$l(\theta) = \ln p(X|\theta) = \ln\left[\prod_{n=1}^{N} p(x^{(n)}|\theta)\right] = \sum_{n=1}^{N} \ln p(x^{(n)}|\theta)$$

Before M step, we compute the posterior probability of latent variable $\gamma_k^{(n)}$ and get the objective function in the E step

objective function: $l(\theta) = \sum_{n=1}^{N} E_{q_n(z^{(n)})}\left[\log p(z^{(n)}, x^{(n)}|\theta)\right]$, s.t. $\sum_{k=1}^{K} \pi_k = 1$

M step: maximize objective function

Given current parameters $\theta^{old} = \{\pi^{old}, \mu^{old}, \Sigma^{old}\}_{k=1}^{K}$

$$\begin{aligned}
\theta^{new} &= \arg\max_{\theta} l(\theta) \\
&= \arg\max_{\theta} \sum_{n=1}^{N} E_{q_n(z^{(n)})}\left[\log p(z^{(n)}, x^{(n)}|\theta)\right] \\
&= \arg\max_{\theta} \left[\sum_{n=1}^{N}\sum_{k=1}^{K} \gamma_k^{(n)} \log(\pi_k) + \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma_k^{(n)} \log(N(x^{(n)}|\mu_k, \Sigma_k))\right]
\end{aligned}$$

Use KKT conditions:

Define Lagrangian function $L(\theta, \lambda) = -l(\theta) + \lambda(1 - \sum_{k=1}^{K} \pi_k)$

Solve equations: $\forall k$, $\begin{cases} \frac{\partial L(\theta,\lambda)}{\partial \mu_k} = 0 \\ \frac{\partial L(\theta,\lambda)}{\partial \Sigma_k} = 0 \\ \frac{\partial L(\theta,\lambda)}{\partial \pi_k} = 0 \\ 1 - \sum_{k=1}^{K} \pi_k = 0 \end{cases}$

We get the updates θ^{new}, where:

$$\forall k, \begin{cases} \mu_k = \frac{1}{N_k}\sum_{n=1}^{N} \gamma_k^{(n)} x^{(n)} \\ \Sigma_k = \frac{1}{N_k}\sum_{n=1}^{N} \gamma_k^{(n)} (x^{(n)} - \mu_k)(x^{(n)} - \mu_k)^T \\ \pi_k = \frac{N_k}{N} \text{ with } N_k = \sum_{n=1}^{N} \gamma_k^{(n)} \end{cases}$$

We continue such iteration until $l(\theta)$ converges to the maximum.

2. 2.1

(1) Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positives.

$$\text{precision} = \frac{TP}{TP+FP}$$, TP: true positive; FP: false positive

(2) Recall

Recall is the ratio of correctly predicted positive observations to all the actual positives.

$$\text{recall} = \frac{TP}{TP+FN}$$, TP: true positive; FN: false negative

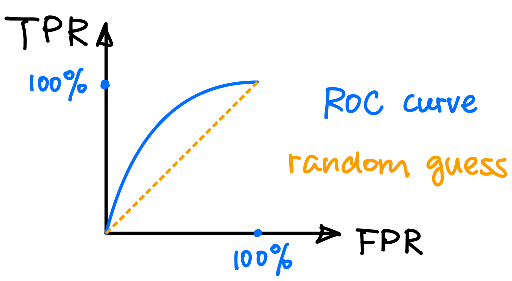
(3) ROC (Receiver Operating Characteristic Curve)

ROC is the plot of the true positive rate (TPR) against the false positive rate (FPR),

$$TPR = \frac{TP}{TP+FN}, \quad FPR = \frac{FP}{FP+TN}$$

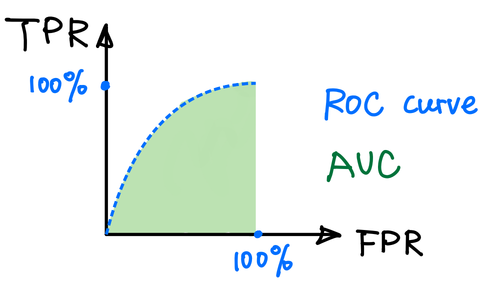
TP: true positive; FP: false positive;

FN: false negative; TN: true negative



The higher the RoC (bending to the top-left corner), the better is the classification accuracy.

(4) AUC (Area Under the RoC)



$0 \leq AUC \leq 1$

The closer AUC is to 1, the better is the classification accuracy.

Let $g(x)$ be a predictor,

$$e_{ij} = g(x_i^+) - g(x_j^-)$$

$$u(e) = \begin{cases} 1, & \text{if } e > 0 \\ 0.5, & \text{if } e = 0 \\ 0, & \text{if } e < 0 \end{cases}$$

$$AUC = \frac{1}{m^+ m^-} \sum_{i=1}^{m^+} \sum_{j=1}^{m^-} u(e_{ij})$$

2.2 Denote $\frac{m^+}{m^+ + m^-} = \alpha$

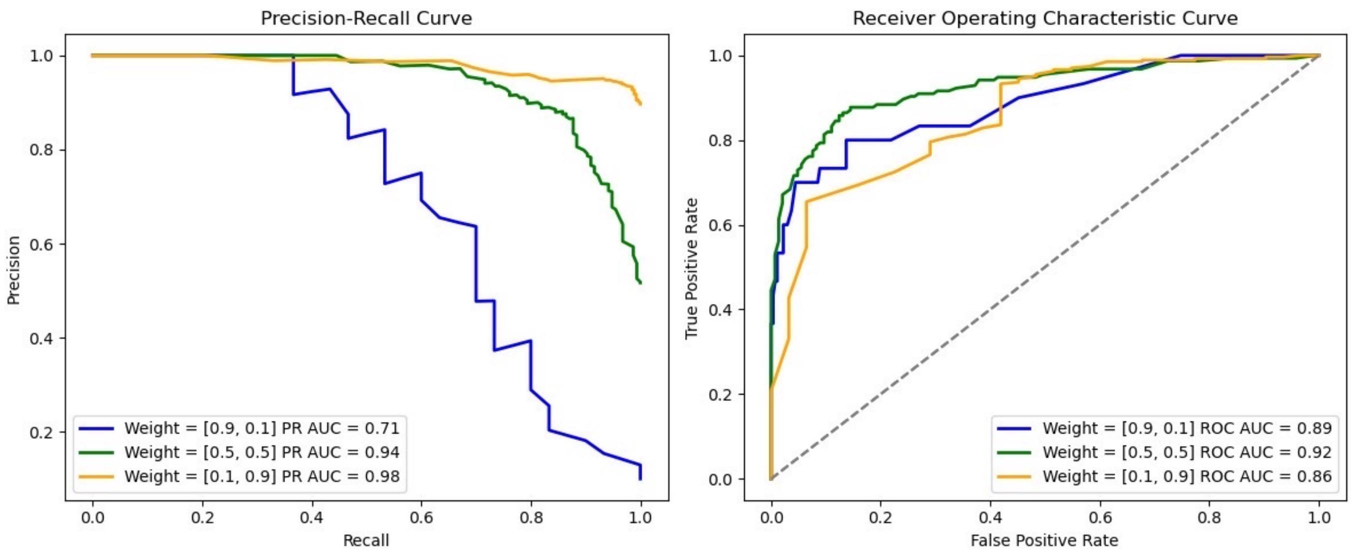
$$\text{recall} = TPR = \frac{TP}{TP+FN} = \frac{TP}{m^+} = 1 - FNR$$

$$FPR = \frac{FP}{FP+TN} = \frac{FP}{m^-} = 1 - TNR$$

$$\text{precision} = \frac{TP}{TP+FP} = \frac{m^+ \cdot TPR}{m^+ \cdot TPR + m^- \cdot FPR} = \frac{\alpha \cdot TPR}{\alpha \cdot TPR + (1-\alpha) FPR}$$

Only precision is related to the proportion of the positive and negative samples, so PR curve will be more affected.

We test our result on a logistic regression with threshold 0.5. blue line is the case where $m^- \gg m^+$, orange is the case where $m^- \ll m^+$, green line is the case where $m^- = m^+$.



3.

```python
import numpy as np

X = np.array([
    [1, 0, 2, -3, -2],
    [0, 1, -3, -2, -3],
    [1, 2, 1, 3, -2],
    [-1, 1, 2, 3, -1],
    [1, 0, 1, -1, 1],
    [2, 3, -1, 1, -2],
    [-2, 3, -3, 2, 3],
    [-2, -2, 2, 3, -2],
    [-2, -2, 1, -3, -3],
    [-3, 2, 0, -1, -2]
])


mu = np.mean(X, axis=0)
cov_matrix = np.cov(X-mu, rowvar=False, bias = True) # devided by n instead of (n-1)

eigenvalues, eigenvectors = np.linalg.eig(cov_matrix)

# Sort eigenvalues and corresponding eigenvectors
sorted_indices = np.argsort(eigenvalues)[::-1]
eigenvalues_sorted = eigenvalues[sorted_indices]
eigenvectors_sorted = eigenvectors[:, sorted_indices]

# Select the top two eigenvalues and corresponding eigenvectors
top_eigenvalues = eigenvalues_sorted[:2]
top_eigenvectors = eigenvectors_sorted[:, :2]

print(eigenvalues)
print("Sorted Eigenvalues:")
print(top_eigenvalues)
print("Sorted Eigenvectors:")
U = top_eigenvectors
print(U)

projection = np.dot(U.T,(X-mu).T)
print(projection)
print(projection.shape)
```

```
[0.77040329 1.98945767 2.97959483 4.90948779 7.09105643]
Sorted Eigenvalues:
[7.09105643 4.90948779]
Sorted Eigenvectors:
[[ 0.00947335  0.12816499]
 [ 0.46377479  0.38603495]
 [-0.27613549 -0.67533319]
 [ 0.71046349 -0.60066664]
 [ 0.45145766  0.13294897]]
```

eigenvalues: $\lambda_1 = 7.09105643$, $\lambda_2 = 4.90948779$, $\lambda_3 = 2.97959483$,
$\lambda_4 = 1.98945767$, $\lambda_5 = 0.77040329$

Choose λ_1 and λ_2 with its corresponding unit-length eigenvectors:

$$U = [q_1 \ q_2] = \begin{bmatrix} 0.00947335 & 0.12816499 \\ 0.46377479 & 0.38603495 \\ -0.27613549 & -0.67533319 \\ 0.71046349 & -0.60066664 \\ 0.45145766 & 0.13294897 \end{bmatrix}.$$

mean colum vector of X: $\mu = \begin{bmatrix} -0.5 \\ 0.8 \\ 0.2 \\ 0.2 \\ -1.3 \end{bmatrix}$

$$\bar{X} = \mu \cdot [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$$

Projection of each data:

$$P = U^T(X - \bar{X})$$

$$= \begin{bmatrix} -3.44335723 & -1.34937249 & 2.0231088 & 1.71570947 & -0.39192178 & 1.62770094 & 5.10983029 & -0.1365459 & -4.57464902 & -0.58050309 \\ 0.49688877 & -3.39780904 & 1.65970801 & 2.84445715 & -0.36973558 & -1.40649158 & -2.30857621 & 4.26367594 & 0.11729188 & -0.90563179 \end{bmatrix}$$