Part (a): Derivation of the Classifier for Maximum Classification Accuracy(a) 部分：最大分类精度的分类器的推导

To derive the classifier that achieves the highest classification accuracy, we can apply **Bayes' Theorem** and use the **maximum a posteriori (MAP)** decision rule. The objective of the classifier is to assign a new sample $\mathbf{x}$ to the class ω_i that maximizes the posterior probability $p(\omega_i|\mathbf{x})$. According to Bayes' theorem, the posterior probability can be written as:

为了导出实现最高分类精度的分类器，我们可以应用**贝叶斯定理**并使用**最大后验（MAP）**决策规则。分类器的目标是分配一个新样本 $\mathbf{x}$ 到班级 ω_i 最大化后验概率 $p(\omega_i|\mathbf{x})$ 。根据贝叶斯定理，后验概率可以写为：

$$p(\omega_i|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_i)p(\omega_i)}{p(\mathbf{x})}$$

Since $p(\mathbf{x})$ is constant across all classes, it does not affect the decision-making process. Thus, we can simplify the MAP decision rule to:

自从 $p(\mathbf{x})$ 在所有类别中都是恒定的，它不会影响决策过程。因此，我们可以将 MAP 决策规则简化为：

$$\omega_{\text{MAP}} = \arg\max_i p(\mathbf{x}|\omega_i)p(\omega_i)$$

This means we classify $\mathbf{x}$ into the class ω_i that maximizes the product of the class-conditional probability $p(\mathbf{x}|\omega_i)$ and the prior probability $p(\omega_i)$.

这意味着我们分类 $\mathbf{x}$ 进入班级 ω_i 最大化类条件概率的乘积 $p(\mathbf{x}|\omega_i)$ 和先验概率 $p(\omega_i)$ 。

Gaussian Class-Conditional Probability高斯类条件概率

The class-conditional probability density function (PDF) $p(\mathbf{x}|\omega_i)$ is given as a multivariate Gaussian distribution:类条件概率密度函数 (PDF) $p(\mathbf{x}|\omega_i)$ 给出为多元高斯分布：

$$p(\mathbf{x}|\omega_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mu_i)^T\Sigma_i^{-1}(\mathbf{x}-\mu_i)\right]$$

where:

- μ_i is the mean vector of class ω_i,μ_i 是类的平均向量 ω_i ,
- Σ_i is the covariance matrix of class ω_i,Σ_i 是类的协方差矩阵 ω_i ,
- $|\Sigma_i|$ is the determinant of the covariance matrix,$|\Sigma_i|$ 是协方差矩阵的行列式,
- d is the dimensionality of the feature space (i.e., the number of features in $\mathbf{x}$).

  d 是特征空间的维数（即特征空间的数量） $\mathbf{x}$ ）。

We are interested in maximizing $p(\mathbf{x}|\omega_i)p(\omega_i)$. Taking the natural logarithm of this expression (because the logarithm is a monotonically increasing function, it preserves the ordering), we obtain the discriminant function:

我们感兴趣的是最大化 $p(\mathbf{x}|\omega_i)p(\omega_i)$ 。对该表达式取自然对数（因为对数是单调递增函数，所以它保持顺序），我们得到判别函数：

$$g_i(\mathbf{x}) = \ln p(\mathbf{x}|\omega_i) + \ln p(\omega_i)$$

Logarithm of the Class-Conditional PDF类条件 PDF 的对数

The logarithm of the Gaussian class-conditional PDF is:高斯类条件 PDF 的对数为：

$$\ln p(\mathbf{x}|\omega_i) = -\frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma_i| - \frac{1}{2}(\mathbf{x}-\mu_i)^T\Sigma_i^{-1}(\mathbf{x}-\mu_i)$$

The term $-\frac{d}{2}\ln(2\pi)$ is constant for all classes and does not affect the decision, so it can be ignored. Thus, the discriminant function becomes:

期限 $-\frac{d}{2}\ln(2\pi)$ 对于所有类都是常数，不影响决策，因此可以忽略。因此，判别函数变为：

$$g_i(\mathbf{x}) = -\frac{1}{2}\ln|\Sigma_i| - \frac{1}{2}(\mathbf{x}-\mu_i)^T\Sigma_i^{-1}(\mathbf{x}-\mu_i) + \ln p(\omega_i)$$

This is the general form of the discriminant function for a Gaussian classifier.

这是高斯分类器判别函数的一般形式。

Special Cases:

- **Case 1**: If the covariance matrices are equal for all classes (i.e., $\Sigma_1 = \Sigma_2 = \cdots = \Sigma_c = \Sigma$), the classifier becomes a **linear discriminant classifier** (LDA), and the discriminant function simplifies to:

  情况 1 ：如果所有类的协方差矩阵都相等（即 $\Sigma_1 = \Sigma_2 = \cdots = \Sigma_c = \Sigma$ ），分类器变为**线性判别分类器**（LDA），判别函数简化为：

  $$g_i(\mathbf{x}) = \mathbf{x}^T\Sigma^{-1}\mu_i - \frac{1}{2}\mu_i^T\Sigma^{-1}\mu_i + \ln p(\omega_i)$$

- **Case 2**: If the covariance matrices are different for each class, the classifier is known as a **quadratic discriminant classifier** (QDA).

  情况 2 ：如果每个类的协方差矩阵不同，则该分类器称为**二次判别分类器**(QDA)。

Thus, the final classifier can be either linear or quadratic, depending on the assumptions regarding the covariance matrices. The decision rule is to classify $\mathbf{x}$ into the class ω_i that maximizes $g_i(\mathbf{x})$.

因此，最终的分类器可以是线性的或二次的，具体取决于关于协方差矩阵的假设。决策规则是分类 $\mathbf{x}$ 进入班级 ω_i 最大化 $g_i(\mathbf{x})$ 。

Answer the question (b)回答问题（b）

Part (b): Design the Derived Classifier Using the Training Samples (b) 部分：使用训练样本设计派生分类器

Training Data训练数据

We are given the following training samples:我们得到以下训练样本：

- Class 1 samples:

  $$\mathbf{x}_1 = (1,1)^T,\ \mathbf{x}_2 = (1,2)^T,\ \mathbf{x}_3 = (2,1)^T$$

- Class 2 samples:

  $$\mathbf{x}_4 = (2,2)^T,\ \mathbf{x}_5 = (2,3)^T,\ \mathbf{x}_6 = (3,2)^T$$

These samples are from two classes, ω_1 (Class 1) and ω_2 (Class 2).

这些样本来自两个类别， ω_1 （1 类）和 ω_2 （第 2 类）。

Step 1: Calculate the Means (μ_1 and μ_2)第 1 步：计算平均值 (μ_1 和 μ_2)

The mean vector for each class can be calculated as the average of the samples from that class.每个类别的平均向量可以计算为该类别的样本的平均值。

- **Mean of Class 1** (μ_1):

  $$\mu_1 = \frac{1}{3}\left[(1,1)^T + (1,2)^T + (2,1)^T\right] = \frac{1}{3}\left[(1+1+2, 1+2+1)\right] = \frac{1}{3}(4,4) = (1.33, 1.33)^T$$

- **Mean of Class 2** (μ_2):

  $$\mu_2 = \frac{1}{3}\left[(2,2)^T + (2,3)^T + (3,2)^T\right] = \frac{1}{3}\left[(2+2+3, 2+3+2)\right] = \frac{1}{3}(7,7) = (2.33, 2.33)^T$$

Step 2: Calculate the Covariance Matrices (Σ_1 and Σ_2)步骤 2：计算协方差矩阵 (Σ_1 和 Σ_2)

The covariance matrix for each class is calculated as:每个类别的协方差矩阵计算如下：

$$\Sigma_i = \frac{1}{n_i}\sum_{j=1}^{n_i}(\mathbf{x}_j-\mu_i)(\mathbf{x}_j-\mu_i)^T$$

where n_i is the number of samples in class i.在哪里 n_i 是类中的样本数 i 。

- **Covariance matrix for Class 1** (Σ_1)**:第 1 类的协方差矩阵**(Σ_1)：

Let's compute the deviations from the mean for each sample:

让我们计算每个样本与平均值的偏差：

$$\mathbf{x}_1-\mu_1 = (1,1)^T - (1.33,1.33)^T = (-0.33,-0.33)^T$$
$$\mathbf{x}_2-\mu_1 = (1,2)^T - (1.33,1.33)^T = (-0.33,0.67)^T$$
$$\mathbf{x}_3-\mu_1 = (2,1)^T - (1.33,1.33)^T = (0.67,-0.33)^T$$

Now, compute the covariance matrix as the average of the outer products of the deviations:

现在，计算协方差矩阵作为偏差外积的平均值：

$$\Sigma_1 = \frac{1}{3}\left[(-0.33,-0.33)^T(-0.33,-0.33) + (-0.33,0.67)^T(-0.33,0.67) + (0.67,-0.33)^T(0.67,-0.33)\right]$$

After performing the matrix multiplications and summing them up:

执行矩阵乘法并对它们求和后：

$$\Sigma_1 = \begin{pmatrix} 0.333 & 0.0 \\ 0.0 & 0.333 \end{pmatrix}$$

- **Covariance matrix for Class 2** (Σ_2)**:第 2 类的协方差矩阵**(Σ_2)：

Let's compute the deviations from the mean for each sample:

让我们计算每个样本与平均值的偏差：

$$\mathbf{x}_4-\mu_2 = (2,2)^T - (2.33,2.33)^T = (-0.33,-0.33)^T$$
$$\mathbf{x}_5-\mu_2 = (2,3)^T - (2.33,2.33)^T = (-0.33,0.67)^T$$
$$\mathbf{x}_6-\mu_2 = (3,2)^T - (2.33,2.33)^T = (0.67,-0.33)^T$$

Now, compute the covariance matrix:现在，计算协方差矩阵：

$$\Sigma_2 = \frac{1}{3}\left[(-0.33,-0.33)^T(-0.33,-0.33) + (-0.33,0.67)^T(-0.33,0.67) + (0.67,-0.33)^T(0.67,-0.33)\right]$$

After performing the matrix multiplications and summing them up:

执行矩阵乘法并对它们求和后：

$$\Sigma_2 = \begin{pmatrix} 0.333 & 0.0 \\ 0.0 & 0.333 \end{pmatrix}$$

Step 3: Construct the Classifier第 3 步：构建分类器

Since both Σ_1 and Σ_2 are equal (i.e., $\Sigma_1 = \Sigma_2 = \Sigma$), we can use a **linear discriminant classifier (LDA)**.

既然两者 Σ_1 和 Σ_2 是相等的（即 $\Sigma_1 = \Sigma_2 = \Sigma$ ），我们可以使用**线性判别分类器（LDA）**。

The discriminant function for LDA is:LDA的判别函数为：

$$g_i(\mathbf{x}) = \mathbf{x}^T\Sigma^{-1}\mu_i - \frac{1}{2}\mu_i^T\Sigma^{-1}\mu_i + \ln p(\omega_i)$$

Given that $\Sigma^{-1} = \Sigma_1^{-1} = \Sigma_2^{-1} = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}$, the classifier can now be applied to classify new points based on which $g_i(\mathbf{x})$ is larger.

鉴于 $\Sigma^{-1} = \Sigma_1^{-1} = \Sigma_2^{-1} = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}$ ，现在可以应用分类器对新点进行分类 $g_i(\mathbf{x})$ 更大。

Thus, the classifier is defined as:因此，分类器定义为：

$$g_1(\mathbf{x}) = 3\mathbf{x}^T\mu_1 - \frac{3}{2}\mu_1^T\mu_1 + \ln p(\omega_1)$$
$$g_2(\mathbf{x}) = 3\mathbf{x}^T\mu_2 - \frac{3}{2}\mu_2^T\mu_2 + \ln p(\omega_2)$$

The decision rule is to classify $\mathbf{x}$ into the class for which the discriminant function is higher.

决策规则是分类 $\mathbf{x}$ 进入判别函数较高的类别。

Let's go through each part of the question.让我们逐一讨论问题的每个部分。

Part (a): Compute the one-dimensional representations a_i and b_i

(a) 部分：计算一维表示 a_i 和 b_i

To obtain the one-dimensional representations of the samples along each eigenvector, we project each sample $\mathbf{x}_i$ onto each eigenvector.

为了获得样本沿每个特征向量的一维表示，我们投影每个样本 $\mathbf{x}_i$ 到每个特征向量上。

1. **Compute** $a_i = \mathbf{x}_i \cdot \phi_1$:

$$a_i = \mathbf{x}_i \cdot \phi_1 = \mathbf{x}_i \cdot \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right)^T$$

2. **Compute** $b_i = \mathbf{x}_i \cdot \phi_2$:

$$b_i = \mathbf{x}_i \cdot \phi_2 = \mathbf{x}_i \cdot \left(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right)^T$$

Let's calculate each projection for the given samples:让我们计算给定样本的每个投影：

- $\mathbf{x}_1 = (1, 6)^T$
- $\mathbf{x}_2 = (4, 7)^T$
- $\mathbf{x}_3 = (2, 9)^T$
- $\mathbf{x}_4 = (5, 10)^T$

Calculation:

For $\phi_1 = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right)^T$:

$$a_1 = \frac{1}{\sqrt{2}}(1) + \frac{1}{\sqrt{2}}(6) = \frac{1+6}{\sqrt{2}} = \frac{7}{\sqrt{2}} = \frac{7\sqrt{2}}{2}$$

Similarly, we calculate a_2, a_3, and a_4.同样，我们计算 a_2 , a_3 ， 和 a_4 。

For $\phi_2 = \left(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right)^T$:

$$b_1 = -\frac{1}{\sqrt{2}}(1) + \frac{1}{\sqrt{2}}(6) = \frac{-1+6}{\sqrt{2}} = \frac{5}{\sqrt{2}} = \frac{5\sqrt{2}}{2}$$

Similarly, we calculate b_2, b_3, and b_4.同样，我们计算 b_2 , b_3 ， 和 b_4 。

After calculating, we obtain all a_i and b_i values.经过计算，我们得到所有 a_i 和 b_i 价值观。

Part (b): Compute the two eigenvalues(b) 部分：计算两个特征值

The eigenvalues can be computed from the covariance matrix of the data. First, we compute the mean vector $\bar{\mathbf{x}}$, then the covariance matrix Σ, and finally its eigenvalues.

特征值可以根据数据的协方差矩阵计算。首先，我们计算均值向量 $\bar{\mathbf{x}}$ ，然后协方差矩阵 Σ ，最后是它的特征值。

Given the covariance matrix has eigenvectors ϕ_1 and ϕ_2, the eigenvalues correspond to the variances along these directions.

给定协方差矩阵具有特征向量 ϕ_1 和 ϕ_2 ，特征值对应于沿这些方向的方差。

Part (c): Compute the two-dimensional samples $\hat{\mathbf{x}}_i$ from single one-dimensional representations a_i or b_i

(c) 部分：计算二维样本 $\hat{\mathbf{x}}_i$ 来自单个一维表示 a_i 或者 b_i

To reconstruct the samples from one-dimensional representations, we approximate $\mathbf{x}_i$ as follows:为了从一维表示重建样本，我们近似 $\mathbf{x}_i$ 如下：

- Using only a_i: $\hat{\mathbf{x}}_i \approx a_i \phi_1$
- Using only b_i: $\hat{\mathbf{x}}_i \approx b_i \phi_2$

For each sample, we calculate $\hat{\mathbf{x}}_i$ using either $a_i \phi_1$ or $b_i \phi_2$ and find the approximation that minimizes the reconstruction error.

对于每个样本，我们计算 $\hat{\mathbf{x}}_i$ 使用任一 $a_i \phi_1$ 或者 $b_i \phi_2$ 并找到使重建误差最小化的近似值。

Let me know if you need detailed calculations for each part!

如果您需要每个部分的详细计算，请告诉我！