Problem 9.1

Suppose you have a two-class classification problem, with D-dimensional observations given by

\[ \vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_D \end{bmatrix} \]

The prior probabilities are given by the known parameter \( \pi_0 \):

\[ p_Y(0) = \pi_0, \quad p_Y(1) = 1 - \pi_0 \]

The likelihoods are given by the known parameters \( \vec{\mu}_0 \) and \( \vec{\mu}_1 \), and by a shared covariance matrix \( \Sigma \) that is the same between the two classes:

\[
p_X|Y(\vec{x}|0) = \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(\vec{x} - \vec{\mu}_0)^T \Sigma^{-1}(\vec{x} - \vec{\mu}_0)} \]
\[
p_X|Y(\vec{x}|1) = \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(\vec{x} - \vec{\mu}_1)^T \Sigma^{-1}(\vec{x} - \vec{\mu}_1)} \]

Demonstrate that the Bayesian classifier, in this case, is a linear classifier, \( h(\vec{x}) = u(\vec{w}^T \vec{x} + b) \). Find the weight vector \( \vec{w} \) and the offset \( b \).

Problem 9.2

Suppose you have a two-class classification problem, with D-dimensional observations given by

\[ \vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_D \end{bmatrix} \]

The prior probabilities are given by the known parameter \( \pi_0 \):

\[ p_Y(0) = \pi_0, \quad p_Y(1) = 1 - \pi_0 \]

The likelihoods are given by the known parameters \( \vec{\mu}_0 \) and \( \vec{\mu}_1 \), and by DIFFERENT known covariance matrices \( \Sigma_0 \) and \( \Sigma_1 \):

\[
p_X|Y(\vec{x}|0) = \frac{1}{(2\pi)^{D/2} |\Sigma_0|^{1/2}} e^{-\frac{1}{2}(\vec{x} - \vec{\mu}_0)^T \Sigma_0^{-1}(\vec{x} - \vec{\mu}_0)} \]
\[
p_X|Y(\vec{x}|1) = \frac{1}{(2\pi)^{D/2} |\Sigma_1|^{1/2}} e^{-\frac{1}{2}(\vec{x} - \vec{\mu}_1)^T \Sigma_1^{-1}(\vec{x} - \vec{\mu}_1)} \]

Demonstrate that the Bayesian classifier, in this case, is a QUADRATIC classifier, that checks whether \( \vec{x} \) is closer to \( \vec{\mu}_1 \) or \( \vec{\mu}_0 \), and classifies accordingly... except that “closer to” is defined using the class-dependent Mahalanobis distances,

\[ h(\vec{x}) = u \left( d_0(\vec{x}, \vec{\mu}_0)^2 - d_1(\vec{x}, \vec{\mu}_1)^2 + b \right) \]
\(d_1\) is a Mahalanobis distance with covariance matrix \(\Sigma_1\), \(d_0\) is a Mahalanobis distance with covariance matrix \(\Sigma_0\), and \(b\) is a constant. Find \(b\).

**Problem 9.3**

Suppose you have a training dataset, \(\mathcal{D}\), that contains \(N\) vectors,

\[
\mathcal{D} = \{\vec{x}_1, \ldots, \vec{x}_N\}, \quad \vec{x}_n = \begin{bmatrix} x_{1n} \\ \vdots \\ x_{Dn} \end{bmatrix}
\]

All drawn from a \(D\)-dimensional Gaussian distribution with mean \(\vec{\mu}\) and covariance matrix \(\Sigma\):

\[
p_{\mathcal{X}}(\vec{x}) = \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu})}
\]

Suppose that you know \(\Sigma\), but you don’t know \(\vec{\mu}\). Your goal is to find a good estimate of \(\vec{\mu}\).

Suppose that the training vectors are i.i.d., so that the likelihood of the training dataset is

\[
p(\mathcal{D}) = \prod_{n=1}^{N} p_{\mathcal{X}}(\vec{x}_n)
\]

Define the maximum-likelihood estimator of \(\vec{\mu}\) to be

\[
\hat{\vec{\mu}}_{ML} = \arg \max_{\vec{\mu}} p(\mathcal{D})
\]

Find \(\hat{\vec{\mu}}_{ML}\).