Fisher Linear Discriminant

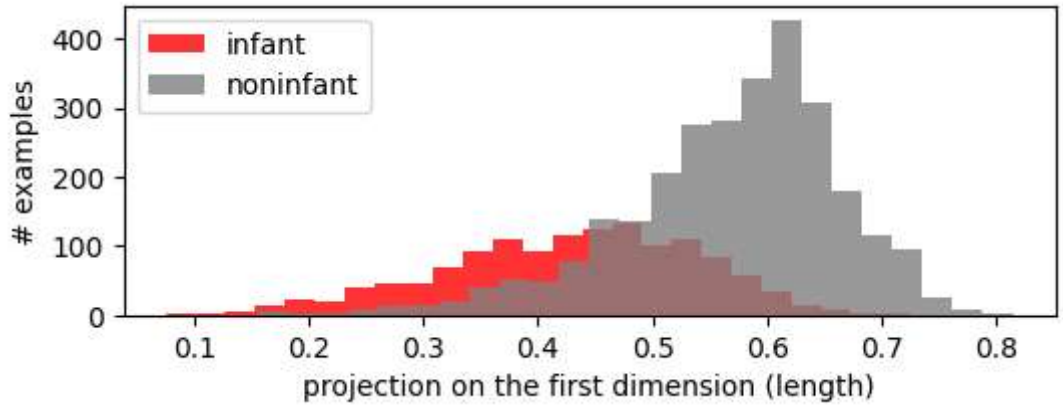
In this exercise, we apply Fisher Linear Discriminant as described in Chapter 3.8.2 of Duda et al. on the UCI Abalone dataset. A description of the dataset is given at the page https://archive.ics.uci.edu/ml/datasets/Abalone. The following two methods are provided for your convenience:

- `utils.Abalone.__init__(self)` reads the Abalone data and instantiates two data matrices corresponding to: *infant (I)*, *non-infant (N)*.
- `utils.Abalone.plot(self,w)` produces a histogram of the data when projected onto a vector `w`, and where each class is shown in a different color.

Sample code that makes use of these two methods is given below. It loads the data, looks at the shape of instantiated matrices, and plots the projection on the first dimension of the data representing the length of the abalone.

In [1]:
```
%matplotlib inline
import utils,numpy
from matplotlib import pyplot as plt

# Load the data
abalone = utils.Abalone()

# Print dataset size for each class
print(abalone.I.shape, abalone.N.shape)

# Project data on the first dimension
w1 = numpy.array([1,0,0,0,0,0,0])
abalone.plot(w1,'projection on the first dimension (length)')
```

```
(1342, 7) (2835, 7)
```



Implementation (10 + 5 + 5 = 20 P)

- **Create a function `w = fisher(X1,X2)` that takes as input the data for two classes and returns the Fisher linear discriminant.**
- **Create a function `objective(X1,X2,w)` that evaluates the objective defined in Equation 96 of Duda et al. for an arbitrary projection vector `w`.**
- **Create a function `z = phi(X)` that returns a quadratic expansion for each data point `x` in the dataset. Such expansion consists of the vector `x` itself, to which we concatenate the vector of all pairwise products between elements of `x`.** In other words, letting $x = (x_1, \ldots, x_d)$ denote the d-dimensional data point, the quadratic expansion for this data point is a $d \cdot (d+3)/2$ dimensional vector given by $\phi(x) = (x_i)_{1 \le i \le d} \cup (x_i x_j)_{1 \le i \le j \le d}$. For example, the quadratic expansion for $d = 2$ is $(x_1, x_2, x_1^2, x_2^2, x_1 x_2)$.

In [4]:
```
def fisher(X1,X2):
    ##### Replace by your code
    mean1 = numpy.mean(X1, axis=0)
    mean2 = numpy.mean(X2, axis=0)
    mean_diff = mean1 - mean2

    # Here I get the term " * (X1.shape[0] - 1)" from Nico, but I don't get that.
    # Since in the slides I can find any clue about this scaling vector.
    # But after adding this term the results seem to be identical to the given ones.
    cov1 = numpy.cov(X1, rowvar=False, bias = True) * (X1.shape[0] - 1)
    cov2 = numpy.cov(X2, rowvar=False, bias = True) * (X2.shape[0] - 1)
    S_w = cov1 + cov2
```

```
    S_w_inv = numpy.linalg.inv(S_w)

    return S_w_inv@mean_diff
    #####

def objective(X1,X2,w):
    ##### Replace by your code
    mean1 = numpy.mean(X1, axis=0)
    mean2 = numpy.mean(X2, axis=0)
    mean_diff = mean1 - mean2

    S_B = numpy.outer(mean_diff, mean_diff)

    cov1 = numpy.cov(X1, rowvar=False, bias = True) * (X1.shape[0] - 1)
    cov2 = numpy.cov(X2, rowvar=False, bias = True) * (X2.shape[0] - 1)
    S_w = cov1 + cov2

    max_mean_class_diff = w.T @ S_B @ w
    min_inclass_var = w.T @ S_w @ w

    return (max_mean_class_diff / min_inclass_var)
    #####

def expand(X):
    # Get the number of samples (n) and dimensionality (d) of the input data
    n, d = X.shape

    # Calculate the number of quadratic terms: (d*(d+3))/2
    expanded_dim = (d * (d + 3)) // 2

    # Initialize an empty array to hold the expanded dataset
    Z = numpy.zeros((n, expanded_dim))

    # Iterate through each data point
    for i, x in enumerate(X):
        # Copy the original vector
        z = list(x)

        # Append all pairwise products (including squares) to the expansion
        for j in range(d):
            for k in range(j, d):
                z.append(x[j] * x[k])

        # Assign the expanded vector to the result array
```

```
        Z[i] = z

    return Z

def my_plot(Z1, Z2, w,name):
        plt.figure(figsize=(6,2))
        plt.xlabel(name)
        plt.ylabel('# examples')
        plt.hist(numpy.dot(Z1,w),bins=25, alpha=0.8,color='red',label='infant')
        plt.hist(numpy.dot(Z2,w),bins=25, alpha=0.8,color='gray',label='noninfant')
        plt.legend()
        plt.show()
```

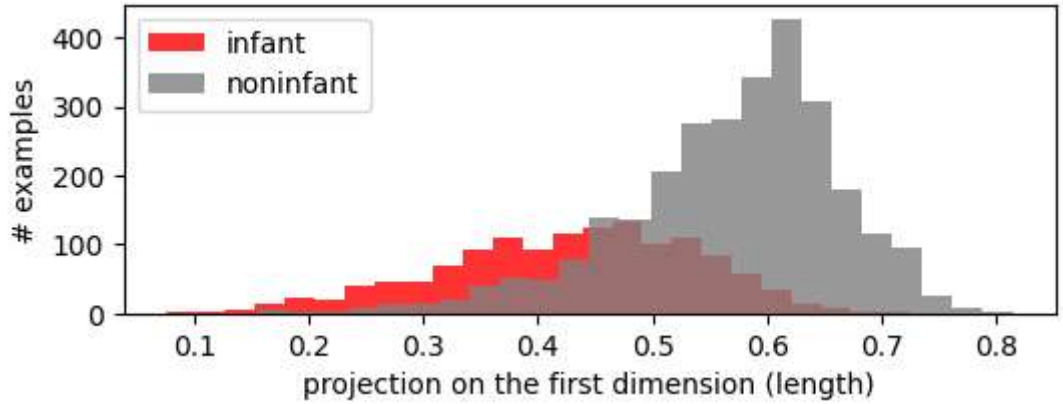
Analysis (5 + 5 = 10 P)

- **Print value of the objective function and the histogram for several values of `w`:**
  - `w` is a canonical coordinate vector for the first feature (length).
  - `w` is the difference between the mean vectors of the two classes.
  - `w` is the Fisher linear discriminant.
  - `w` is the Fisher linear discriminant (after quadratic expansion of the data).

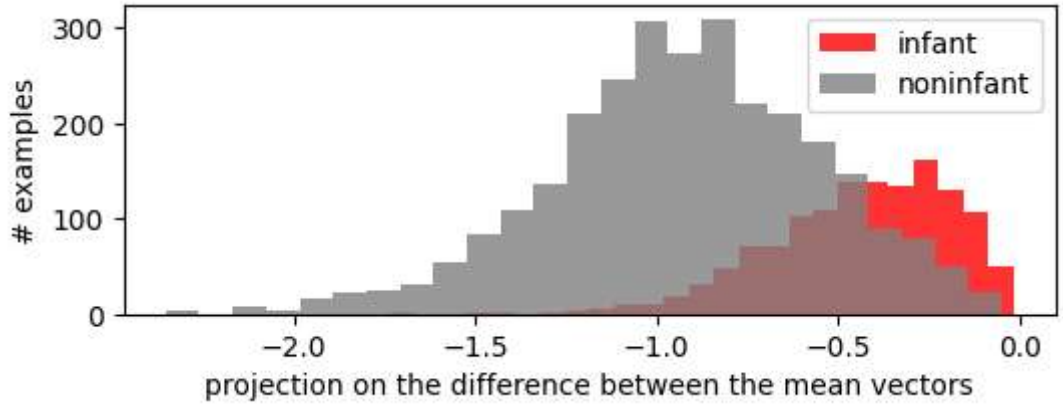
In [6]:
```
##### REPLACE BY YOUR CODE
%matplotlib inline

w1 = numpy.array([1,0,0,0,0,0,0])
print('First dimension:')
print(objective(abalone.I,abalone.N,w1))
abalone.plot(w1,'projection on the first dimension (length)')

mean1 = numpy.mean(abalone.I, axis=0)
mean2 = numpy.mean(abalone.N, axis=0)
mean_diff = mean1 - mean2
w2 = mean_diff
print('Difference:')
print(objective(abalone.I,abalone.N,w2))
abalone.plot(w2,'projection on the difference between the mean vectors')


w3 = fisher(abalone.I,abalone.N)
```

```
print('Fisher linear discriminant:')
print(objective(abalone.I,abalone.N,w3))
abalone.plot(w3,'projection on the Fisher linear discriminant')


Z1 = expand(abalone.I)
Z2 = expand(abalone.N)
w4 = fisher(Z1, Z2)
print('Fisher linear discriminant(after expand):')
print(objective(Z1, Z2, w4))
my_plot(Z1, Z2, w4,'projection on the first dimension (length)')
#####
```

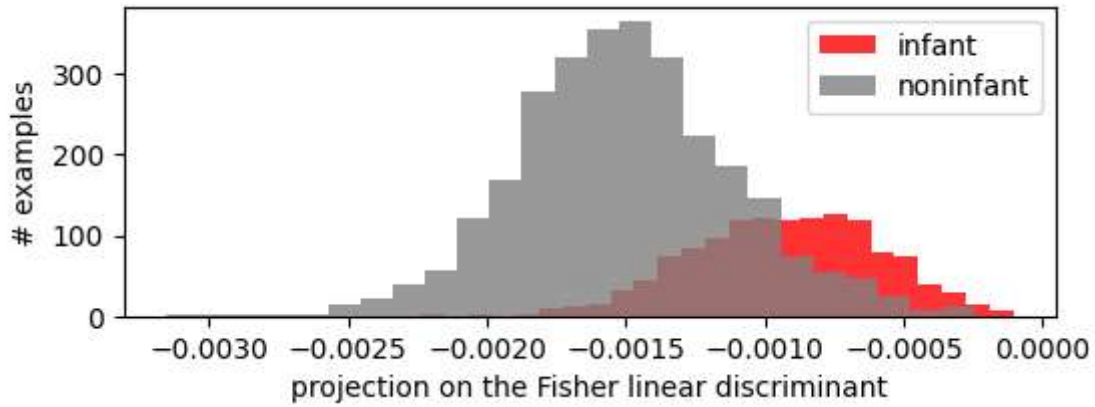
First dimension:
0.00048003649875237174



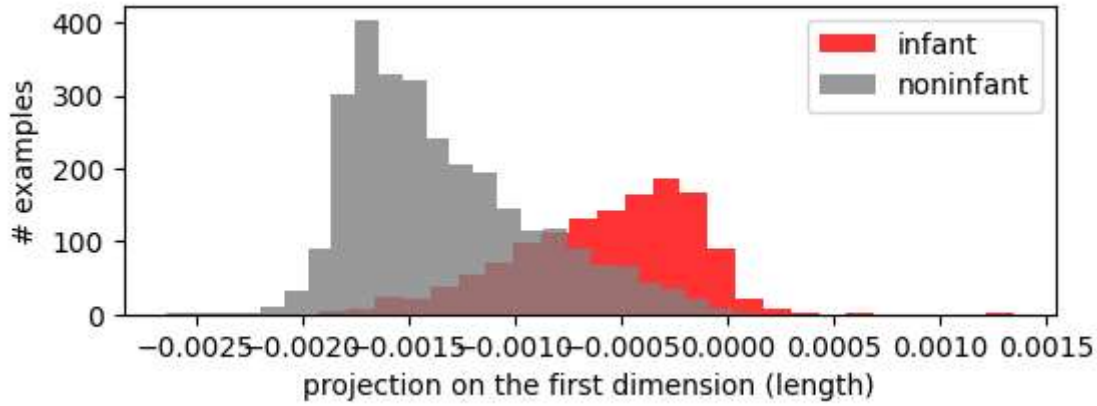
Difference:
0.0004997853617622609



```
Fisher linear discriminant:
0.0005694412070089378
```




```
Fisher linear discriminant(after expand):
0.0007663654723881523
```



Exercises for the course
Machine Learning 1
Winter semester 2024/25

Fachgebiet Maschinelles Lernen
Institut für Softwaretechnik und theoretische Informatik
Fakultät IV, Technische Universität Berlin
Prof. Dr. Klaus-Robert Müller
Email: klaus-robert.mueller@tu-berlin.de

Exercise Sheet 3

Exercise 1: Fisher Discriminant (10 + 10 + 10 P)

The objective function to find the Fisher Discriminant has the form

$$\max_{\boldsymbol{w}} \frac{\boldsymbol{w}^\top \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^\top \boldsymbol{S}_W \boldsymbol{w}}$$

where $\boldsymbol{S}_B = (\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^\top$ is the between-class scatter matrix and $\boldsymbol{S}_W$ is within-class scatter matrix, assumed to be positive definite. Because there are infinitely many solutions (multiplying $\boldsymbol{w}$ by a scalar doesn't change the objective), we can extend the objective with a constraint, e.g. that enforces $\boldsymbol{w}^\top \boldsymbol{S}_W \boldsymbol{w} = 1$.

(a) *Reformulate* the problem above as an optimization problem with a quadratic objective and a quadratic constraint.

(b) *Show* using the method of Lagrange multipliers that the solution of the reformulated problem is also a solution of the generalized eigenvalue problem:

$$\boldsymbol{S}_B \boldsymbol{w} = \lambda \boldsymbol{S}_W \boldsymbol{w}$$

(c) Show that the solution of this optimization problem is equivalent (up to a scaling factor) to

$$\boldsymbol{w}^\star = \boldsymbol{S}_W^{-1}(\boldsymbol{m}_2 - \boldsymbol{m}_1)$$

Exercise 2: Bounding the Error (10 + 10 P)

The direction learned by the Fisher discriminant is equivalent to that of an optimal classifier when the class-conditioned data densities are Gaussian with same covariance. In this particular setting, we can derive a bound on the classification error which gives us insight into the effect of the mean and covariance parameters on the error.

Consider two data generating distributions $P(\boldsymbol{x} \mid \omega_1) = \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ and $P(\boldsymbol{x} \mid \omega_2) = \mathcal{N}(-\boldsymbol{\mu}, \Sigma)$ with $\boldsymbol{x} \in \mathbb{R}^d$. Recall that the Bayes error rate is given by:

$$P(\text{error}) = \int_{\boldsymbol{x}} P(\text{error} \mid \boldsymbol{x})\, p(\boldsymbol{x})\, d\boldsymbol{x}$$

(a) Show that the conditional error can be upper-bounded as:

$$P(\text{error} \mid \boldsymbol{x}) \leq \sqrt{P(\omega_1 \mid \boldsymbol{x}) P(\omega_2 \mid \boldsymbol{x})}$$

(b) Show that the Bayes error rate can then be upper-bounded by:

$$P(\text{error}) \leq \sqrt{P(\omega_1)P(\omega_2)} \cdot \exp\Big(-\frac{1}{2}\boldsymbol{\mu}^\top \Sigma^{-1} \boldsymbol{\mu}\Big)$$

Exercise 3: Fisher Discriminant (10 + 10 P)

Consider the case of two classes ω_1 and ω_2 with associated data generating probabilities

$$p(\boldsymbol{x} \mid \omega_1) = \mathcal{N}\left(\begin{bmatrix}-1\\-1\end{bmatrix}, \begin{bmatrix}2 & 0\\0 & 1\end{bmatrix}\right) \qquad \text{and} \qquad p(\boldsymbol{x} \mid \omega_2) = \mathcal{N}\left(\begin{bmatrix}+1\\+1\end{bmatrix}, \begin{bmatrix}2 & 0\\0 & 1\end{bmatrix}\right)$$

(a) Find for this dataset the Fisher discriminant $\boldsymbol{w}$ (i.e. the projection $y = \boldsymbol{w}^\top \boldsymbol{x}$ under which the ratio between inter-class and intra-class variability is maximized).

(b) Find a projection for which the ratio is minimized.

Exercise 4: Programming (30 P)

Download the programming files on ISIS and follow the instructions.

Exercise 1: Fisher Discriminant (10 + 10 + 10 P)

The objective function to find the Fisher Discriminant has the form

$$\max_{\boldsymbol{w}} \frac{\boldsymbol{w}^\top \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^\top \boldsymbol{S}_W \boldsymbol{w}}$$

where $\boldsymbol{S}_B = (\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^\top$ is the between-class scatter matrix and $\boldsymbol{S}_W$ is within-class scatter matrix, assumed to be positive definite. Because there are infinitely many solutions (multiplying $\boldsymbol{w}$ by a scalar doesn't change the objective), we can extend the objective with a constraint, e.g. that enforces $\boldsymbol{w}^\top \boldsymbol{S}_W \boldsymbol{w} = 1$.

(a) *Reformulate* the problem above as an optimization problem with a quadratic objective and a quadratic constraint.

Solution:

The problem can be formulated as

$$\max_w \; w^T S_B w \quad \text{s.t.} \quad w^T S_w w = 1 \quad \Longrightarrow \quad \max_w \; w^T S_B w \quad \text{s.t.} \quad 1 - w^T S_w w = 0$$

(b) *Show* using the method of Lagrange multipliers that the solution of the reformulated problem is also a solution of the generalized eigenvalue problem:

$$\boldsymbol{S}_B \boldsymbol{w} = \lambda \boldsymbol{S}_W \boldsymbol{w}$$

Solution.

We can formulate the Lagrange function:

$$\mathcal{L}(\lambda) = w^T S_B w + \lambda (1 - w^T S_w w)$$

$$\frac{\partial \mathcal{L}}{\partial w} = S_B w - \lambda S_w w$$

$$\Downarrow \text{ let } \frac{\partial \mathcal{L}}{\partial w} = 0$$

$$\therefore \quad S_B w = \lambda S_w w \longrightarrow \text{generalized Eigenvalue problem.}$$

(c) Show that the solution of this optimization problem is equivalent (up to a scaling factor) to

$$\boldsymbol{w}^\star = \boldsymbol{S}_W^{-1}(\boldsymbol{m}_2 - \boldsymbol{m}_1)$$

Solution:

From above we have:

$$S_B w = \lambda S_w w$$

$$\Downarrow \text{ left multiplied by } S_w^{-1}$$

$$\therefore \quad S_w^{-1} S_B w = \lambda S_w^{-1} S_w w$$

$$\Downarrow$$

$$S_w^{-1} S_B w = \lambda I w$$

$$\Downarrow S_B = (m_2 - m_1)(m_2 - m_1)^T$$

$$S_w^{-1} (m_2 - m_1) \underbrace{(m_2 - m_1)^T w}_{\text{scaler}} = \lambda w$$

$$\Downarrow$$

$$\beta$$

$$\Downarrow$$

$$S_w^{-1} (m_2 - m_1) \cdot \beta = \lambda w$$

$\therefore$ suppose the scaling factor is $\lambda^* = \frac{\lambda}{\beta}$

$$\Downarrow$$

$$w^* = \frac{1}{\lambda^*} S_w^{-1} (m_2 - m_1)$$

$$\Downarrow$$

α (scaling factor)

$$\therefore \quad w^* = \alpha S_w^{-1} (m_2 - m_1)$$

$$\Downarrow$$

$$w^* = S_w^{-1} (m_2 - m_1)$$

Exercise 2: Bounding the Error (10 + 10 P)

The direction learned by the Fisher discriminant is equivalent to that of an optimal classifier when the class-conditioned data densities are Gaussian with same covariance. In this particular setting, we can derive a bound on the classification error which gives us insight into the effect of the mean and covariance parameters on the error.

Consider two data generating distributions $P(\boldsymbol{x} \mid \omega_1) = \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ and $P(\boldsymbol{x} \mid \omega_2) = \mathcal{N}(-\boldsymbol{\mu}, \Sigma)$ with $\boldsymbol{x} \in \mathbb{R}^d$. Recall that the Bayes error rate is given by:

$$P(\text{error}) = \int_{\boldsymbol{x}} P(\text{error} \mid \boldsymbol{x})\, p(\boldsymbol{x})\, d\boldsymbol{x}$$

(a) Show that the conditional error can be upper-bounded as:

$$P(\text{error} \mid \boldsymbol{x}) \leq \sqrt{P(\omega_1 \mid \boldsymbol{x}) P(\omega_2 \mid \boldsymbol{x})}$$

Solution:

From sheet 1 we have the definition:

$$P(\text{error}|x) = \min\left(P(\omega_1|x), P(\omega_2|x)\right)$$

And we know that there is an inequality called generalized mean inequality:

$$\min(x_1, \cdots, x_n) \leq M_p(x_1, \cdots, x_n) \leq \max(x_1, \cdots, x_n)$$

where $M_p(x_1, \cdots, x_n) = \left(\frac{1}{n}\sum_{i=1}^{n} x_i^p\right)^{\frac{1}{p}} \quad p > 0$

$$M_0(x_1, \cdots, x_n) = \left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}}$$

⇒ equal to geometric mean

∴ we can derive:

$$\min\left(P(\omega_1|x), P(\omega_2|x)\right) \leq M_0\left(P(\omega_1|x), P(\omega_2|x)\right)$$

$$\Downarrow$$

$$\min\left(P(\omega_1|x), P(\omega_2|x)\right) \leq \left(P(\omega_1|x) P(\omega_2|x)\right)^{\frac{1}{2}}$$

$$\Downarrow$$

$$\min\left(P(\omega_1|x), P(\omega_2|x)\right) \leq \sqrt{P(\omega_1|x) P(\omega_2|x)}$$

$$\Downarrow$$

$$P(\text{error}|x) \leq \sqrt{P(\omega_1|x) P(\omega_2|x)}$$

(b) Show that the Bayes error rate can then be upper-bounded by:

$$P(\text{error}) \leq \sqrt{P(\omega_1)P(\omega_2)} \cdot \exp\left(-\frac{1}{2}\boldsymbol{\mu}^T\Sigma^{-1}\boldsymbol{\mu}\right)$$

Solution:

$$P(\text{error}) \leq \int_x P(\text{error}|x)\,P(x)\,dx$$

$$P(\text{error}) \leq \int_x \sqrt{P(\omega_1|x)P(\omega_2|x)}\,P(x)\,dx$$

$$= \int_x \sqrt{\frac{P(x|\omega_1)P(\omega_1)}{P(x)} \cdot \frac{P(x|\omega_2)P(\omega_2)}{P(x)}}\,P(x)\,dx$$

$$= \int_x \sqrt{P(x|\omega_1)P(\omega_1)\,P(x|\omega_2)P(\omega_2)}\,dx$$

$$= \sqrt{P(\omega_1)P(\omega_2)} \int_x \sqrt{P(x|\omega_1)P(x|\omega_2)}\,dx$$

Since $P(x|\omega_1)$ and $P(x|\omega_2)$ have opposite mean and same variance Gaussian.

$$\therefore \int_x \sqrt{P(x|\omega_1)P(x|\omega_2)}\,dx$$

$$= \int_x \sqrt{2d\frac{1}{\sqrt{2\pi}(\det(\Sigma))^2}\exp\left(-\tfrac{1}{2}\left((x-\mu)^T\Sigma^{-1}(x-\mu) + (x+\mu)^T\Sigma^{-1}(x+\mu)\right)\right)}\,dx$$

$$= \int_x \frac{1}{d\sqrt{2\pi}\,\det(\Sigma)} \sqrt{\exp\left(-\tfrac{1}{2}\left(2x^T\Sigma^{-1}x + 2\mu^T\Sigma^{-1}\mu\right)\right)}\,dx$$

$$= \int_x \frac{1}{d\sqrt{2\pi}\,\det(\Sigma)} \sqrt{\exp(-x^T\Sigma^{-1}x)\cdot\exp(-\mu^T\Sigma^{-1}\mu)}\,dx$$

$$= \exp\left(-\tfrac{1}{2}\mu^T\Sigma^{-1}\mu\right) \cdot \underbrace{\int_x \frac{1}{d\sqrt{2\pi}\,\det(\Sigma)} \exp\left(-\tfrac{1}{2}x^T\Sigma^{-1}x\right)dx}_{\mathcal{N}(0,\Sigma)}$$

爽爲的 天

$\therefore = \exp(-\frac{1}{2}\mu^T\Sigma^{-1}\mu) \cdot \underbrace{1}_{=} \quad$ (integrate the pdf on the whole interval)

$\therefore p(\text{error}) \leq \sqrt{P(\omega_1)P(\omega_2)}\ \exp(-\frac{1}{2}\mu^T\Sigma^{-1}\mu)$

$\therefore$ proofed!

Exercise 3: Fisher Discriminant (10 + 10 P)

Consider the case of two classes ω_1 and ω_2 with associated data generating probabilities

$$p(\boldsymbol{x} \mid \omega_1) = \mathcal{N}\left(\begin{bmatrix}-1\\-1\end{bmatrix}, \begin{bmatrix}2 & 0\\0 & 1\end{bmatrix}\right) \quad \text{and} \quad p(\boldsymbol{x} \mid \omega_2) = \mathcal{N}\left(\begin{bmatrix}+1\\+1\end{bmatrix}, \begin{bmatrix}2 & 0\\0 & 1\end{bmatrix}\right)$$

(a) Find for this dataset the Fisher discriminant $\boldsymbol{w}$ (i.e. the projection $y = \boldsymbol{w}^\top \boldsymbol{x}$ under which the ratio between inter-class and intra-class variability is maximized).

Solution:

$$w^* = S_w^{-1}(m_2 - m_1)$$

$$= (\Sigma_1 + \Sigma_2)^{-1}(m_2 - m_1)$$

$$= \left(\begin{bmatrix}2 & 0\\0 & 1\end{bmatrix} + \begin{bmatrix}2 & 0\\0 & 1\end{bmatrix}\right)^{-1} \cdot \begin{bmatrix}2\\2\end{bmatrix}$$

$$= \begin{bmatrix}4 & 0\\0 & 2\end{bmatrix}^{-1} \cdot \begin{bmatrix}2\\2\end{bmatrix}$$

$$= \begin{bmatrix}\frac{1}{4} & 0\\0 & \frac{1}{2}\end{bmatrix} \cdot \begin{bmatrix}2\\2\end{bmatrix} = \begin{bmatrix}\frac{1}{2}\\1\end{bmatrix}$$ ✓

$\therefore$ w^* could be $\begin{bmatrix}\frac{1}{2}\\1\end{bmatrix}$

(b) Find a projection for which the ratio is minimized.

Solution:

$$S_w^{-1} S_B w = \lambda w$$

$\therefore$ $S_w^{-1} = \begin{bmatrix}\frac{1}{4} & 0\\0 & \frac{1}{2}\end{bmatrix}$, $S_B = (\mu_2 - \mu_1)(\mu_2 - \mu_1)^\top = \begin{bmatrix}4 & 4\\4 & 4\end{bmatrix}$

$\therefore$ $S_w^{-1} S_B = \begin{bmatrix}1 & 1\\2 & 2\end{bmatrix} = A$

$\therefore$ $Aw = \lambda w$

$\therefore$ $|\lambda I - A| = (\lambda - 1)(\lambda - 2) - 2 = \lambda^2 - 3\lambda = 0$

$\therefore$ ① $\lambda_1 = 0 \rightarrow w$ could be $\begin{bmatrix}2\\-2\end{bmatrix}$ ✓

② $\lambda_2 = 3 \rightarrow w$ could be $\begin{bmatrix}1\\2\end{bmatrix}$ or $\begin{bmatrix}2\\4\end{bmatrix}$ ....