Classification Assessed Coursework

Student Name: Harleen Gulati , Student Number: 2101550

2024-04-05

Question 1 (a.i)

Rewrite

$$\mathbb{P}(G = k|X = z) = \frac{\pi_k f_k(z)}{\sum_{L=1}^{K} \pi_L f_L(z)} (*)$$

using Bayes

We are told in the question $X|G = k \sim Uniform(a_k, b_k)$ thus by the hint given in the question we deduce

$$f_k(z) = \frac{\mathbb{I}(a_k \leq z \leq b_k)}{\prod_{j=1}^{p}(b_{jk} - a_{jk})} = \prod_{j=1}^{p} \frac{\mathbb{I}(a_{jk} \leq z_j \leq b_{jk})}{b_{jk} - a_{jk}}$$

Thus we can write (*) as follows:

$$\mathbb{P}(G = k|X = z) = \frac{\pi_k \prod_{j=1}^{p} \frac{\mathbb{I}(a_{jk} \leq z_j \leq b_{jk})}{b_{jk} - a_{jk}}}{\sum_{L=1}^{K} \pi_L f_L(z)} (**)$$

Now maximizing (**) with respect to k is equivalent to maximizing $\pi_k \prod_{j=1}^{p} \frac{\mathbb{I}(a_{jk} \leq z_j \leq b_{jk})}{b_{jk} - a_{jk}}$ with respect to k because $\sum_{L=1}^{K} \pi_L f_L(z)$ is constant with respect to k.

Thus $\hat{G}(z) \in \underset{k}{\operatorname{argmax}}\, \mathbb{P}(G = k|X = z) \in \underset{k}{\operatorname{argmax}}\, \pi_k \prod_{j=1}^{p} \frac{\mathbb{I}(a_{jk} \leq z_j \leq b_{jk})}{b_{jk} - a_{jk}}$

Question 1 (a. ii)

$$L(\pi_1, a_1, b_1, \ldots, \pi_k, a_k, b_k | g_1, x_1, \ldots, g_n, x_n) \,\alpha\, f(g_1, x_1, \ldots, g_n, x_n | \pi_1, a_1, b_1, \ldots, \pi_k, a_k, b_k)$$
$$= \prod_{i=1}^{n} f(g_i, x_i | \pi_1, a_1, b_1, \ldots, \pi_k, a_k, b_k)$$

(Equation I)

(because $(g_1, x_1), \ldots, (g_n, x_n)$ are i.i.d observations)

Now consider $f(g_i, x_i | \pi_1, a_1, b_1, \ldots, \pi_k, a_k, b_k)$

We know by Bayes: $\mathbb{P}(G = k|X = z) = \frac{\pi_k f_k(z)}{\sum_{L=1}^{K} \pi_L f_L(z)} (*)$

We know by conditional probability: $\mathbb{P}(G = k|X = z) = \frac{f(g=k, x=z)}{\sum_{L=1}^{K} \pi_L f_L(z)} (**)$

Using $(*)\ and\ (**)$ we deduce
$f(g = k, x = z) = \mathbb{P}(G = k|X = z) \sum_{L=1}^{K} \pi_L f_L(z) = \frac{\pi_k f_k(z)}{\sum_{L=1}^{K} \pi_L f_L(z)} \sum_{L=1}^{K} \pi_L f_L(z) = \pi_k f_k(z)$

Hence $f(g_i, x_i | \pi_1, a_1, b_1, \ldots, \pi_k, a_k, b_k) = \sum_{k=1}^{K} \mathbb{I}(g_i = k) \pi_k f_k(x_i) (* * *)$

We showed in 1 (a.i) that $f_k(x_i) = \prod_{j=1}^{p} \frac{\mathbb{I}(a_{jk} \leq x_{ij} \leq b_{jk})}{b_{jk} - a_{jk}} (* * **)$

Thus using $(***)\ and\ (****)$ we deduce

$$L(\pi_1, a_1, b_1, \ldots, \pi_k, a_k, b_k | g_1, x_1, .., g_n, x_n)\ \alpha\ f(g_1, x_1, \ldots, g_n, x_n | \pi_1, a_1, b_1, \ldots, \pi_k, a_k, b_k)$$
$$= \prod_{i=1}^{n} [\sum_{k=1}^{K} \mathbb{I}(g_i = k)\pi_k \prod_{j=1}^{p} \frac{\mathbb{I}(a_{jk} \le x_{ij} \le b_{jk})}{b_{jk} - a_{jk}}]$$

(Equation II)

Note $n_1, \ldots, n_k$ represents the number of observations we have in class 1,…,number of observations we have in class k

So in (Equation II) we will be multiplying π_k to itself n_k times for all $k \in [1, .., K]$ (because for each $k \in [1, \ldots, K]$ $\mathbb{I}(g_i = k) = 1$) holds exactly n_k times and thus we can rewrite (Equation II) as follows

$$L(\pi_1, a_1, b_1, \ldots, \pi_k, a_k, b_k | g_1, x_1, .., g_n, x_n)\ \alpha\ \prod_{i=1}^{n} [\sum_{k=1}^{K} \mathbb{I}(g_i = k)\pi_k \prod_{j=1}^{p} \frac{\mathbb{I}(a_{jk} \le x_{ij} \le b_{jk})}{b_{jk} - a_{jk}}]$$
$$= (\prod_{k=1}^{K} \pi_k^{n_k})(\prod_{i=1}^{N} \sum_{k=1}^{K} \mathbb{I}(g_i = k) \prod_{j=1}^{p} \frac{\mathbb{I}(a_{jk} \le x_{ij} \le b_{jk})}{b_{jk} - a_{jk}})$$

(Equation III)

Let $k \in [1, .., K]$ and $(g_k, x_{1k}), \ldots, (g_k, x_{n_k k})$ be the observations where g = k (i.e. where $\mathbb{I}(g_i = k) = 1$)

There are n_k of such observations and thus since $b_{jk} - a_{jk}$ is independent of the observation and only dependent on k, $b_{jk} - a_{jk}$ is observed in (Equation III) n_k times (because each time $\mathbb{I}(g_i = k) = 1$, we observe a $b_{jk} - a_{jk}$) and hence we deduce that $b_{jk} - a_{jk}$ must be multiplied to itself n_k times in (Equation III)

Since k was chosen arbitrary, this explanation holds for all values of k allowing us to rewrite (Equation III) as follows:

$$L(\pi_1, a_1, b_1, \ldots, \pi_k, a_k, b_k | g_1, x_1, .., g_n, x_n)\ \alpha\ (\prod_{k=1}^{K} \pi_k^{n_k})(\prod_{k=1}^{K} \prod_{j=1}^{p} \frac{1}{(b_{jk} - a_{jk})^{n_k}} \prod_{i=1}^{N} \sum_{k=1}^{K} \mathbb{I}(g_i = k) \prod_{j=1}^{p} \mathbb{I}(a_{jk} \le x_{ij} \le b_{jk}))$$

(Equation IV)

Let $k \in [1, .., k]$ and without loss of generality let $(g_k, x_1), \ldots, (g_k, x_{n_k})$ be the n_k observations with g=k.

Then by the indicator function in (Equation IV) it must hold that for all $1 \le j \le p$

$a_{jk} \le x_{j1}, a_{jk} \le x_{j2}, \ldots, a_{jk} \le x_{jn_k}$ (i.e. for any $i \in [1, .., n_k]$ the jth component of x_i is $\ge$ the jth component of a_k)

This is equivalent to us saying $a_{jk} \le min(x_{ji} : g_i = k, 1 \le i \le n)$ (i.e. the jth component of a_k is smaller than the smallest jth component of x_i for any $i \in [1, .., n_k]$)

Similarly, by the indicator function in (Equation IV) it must also hold that for all $1 \le j \le p$

$b_{jk} \ge x_{j1}, b_{jk} \ge x_{j2}, \ldots, b_{jk} \ge x_{jn_k}$ (i.e. for any $i \in [1, .., n_k]$ the jth component of x_i is $\le$ the jth component of b_k)

This is equivalent to us saying $b_{jk} \ge max(x_{ji} : g_i = k, 1 \le i \le n)$ (i.e. the jth component of b_k is greater than the largest jth component of x_i for any $i \in [1, .., n_k]$)

Thus the restrictions in the indicator function in (Equation IV) is equivalent to the following restrictions:

$a_{jk} \le min(x_{ji} : g_i = k, 1 \le i \le n) = \hat{a}_{jk}$ and $b_{jk} \ge max(x_{ji} : g_i = k, 1 \le i \le n) = \hat{b}_{jk}$

Thus $\prod_{n=1}^{N_k} \prod_{j=1}^{p} \mathbb{I}(a_{jk} \le x_i \le b_{jk})$ is equivalent to $\prod_{j=1}^{p} \mathbb{I}(a_{jk} \le \hat{a}_{jk}, b_{jk} \ge \hat{b}_{jk})$

Since we chose k arbitrary, this holds for all k.

Thus we can rewrite (Equation IV) as follows:

$$L(\pi_1, a_1, b_1, \ldots, \pi_k, a_k, b_k | g_1, x_1, .., g_n, x_n) \; \alpha \; (\prod_{k=1}^{K} \pi_k^{n_k})(\prod_{k=1}^{K} \prod_{j=1}^{p} \frac{1}{(b_{jk} - a_{jk})^{n_k}} \prod_{i=1}^{N} \sum_{k=1}^{K} \mathbb{I}(g_i = k) \prod_{j=1}^{p} \mathbb{I}(a_{jk} \leq x_{ij} \leq b_{jk}))$$

$$= (\prod_{k=1}^{K} \pi_k^{n_k})(\prod_{k=1}^{K} \prod_{j=1}^{p} \frac{\mathbb{I}(a_{jk} \leq \hat{a}_{jk}, b_{jk} \geq \hat{b}_{jk})}{(b_{jk} - a_{jk})^{n_k}})$$

as required.

Question 1 (a. iii)

Let $k \in G$.

Consider the maximum likelihood estimator of a_k, which is obtained from maximizing the likelihood in (a. ii) with respect to a_k

$$L(\pi_1, a_1, b_1, \ldots, \pi_k, a_k, b_k | g_1, x_1, .., g_n, x_n) = (\prod_{k=1}^{K} \pi_k^{n_k})(\prod_{k=1}^{K} \prod_{j=1}^{p} \frac{\mathbb{I}(a_{jk} \leq \hat{a}_{jk}, b_{jk} \geq \hat{b}_{jk})}{(b_{jk} - a_{jk})^{n_k}})$$

Since π_k, n_k are non-negative and $\frac{\mathbb{I}(a_{jk} \leq \hat{a}_{jk}, b_{jk} \geq \hat{b}_{jk})}{(b_{jk} - a_{jk})^{n_k}}$ strictly positive (as given to us in the hint) we can maximize the likelihood with respect to a_k by focusing on maximizing only the terms in the likelihood dependent on a_k.

Thus we focus on maximizing $\prod_{j=1}^{p} \frac{\mathbb{I}(a_{jk} \leq \hat{a}_{jk}, b_{jk} \geq \hat{b}_{jk})}{(b_{jk} - a_{jk})^{n_k}}$ with respect to a_k. Since $\frac{\mathbb{I}(a_{jk} \leq \hat{a}_{jk}, b_{jk} \geq \hat{b}_{jk})}{(b_{jk} - a_{jk})^{n_k}}$ is strictly positive, maximizing $\prod_{j=1}^{p} \frac{\mathbb{I}(a_{jk} \leq \hat{a}_{jk}, b_{jk} \geq \hat{b}_{jk})}{(b_{jk} - a_{jk})^{n_k}}$ with respect to a_k is equivalent to maximizing $\frac{\mathbb{I}(a_{jk} \leq \hat{a}_{jk}, b_{jk} \geq \hat{b}_{jk})}{(b_{jk} - a_{jk})^{n_k}}$ with respect to a_k for each $j \in [1, .., p]$

So we focus on choosing a_k maximizing $\frac{\mathbb{I}(a_{jk} \leq \hat{a}_{jk}, b_{jk} \geq \hat{b}_{jk})}{(b_{jk} - a_{jk})^{n_k}}$ for each $j \in [1, .., p]$

Note as a_{jk} increases, $(b_{jk} - a_{jk})$ decreases and thus $(b_{jk} - a_{jk})^{n_k}$ decreases (because n_k is a non-negative value), hence $\frac{1}{(b_{jk} - a_{jk})^{n_k}}$ increases.

So, to maximize $\frac{\mathbb{I}(a_{jk} \leq \hat{a}_{jk}, b_{jk} \geq \hat{b}_{jk})}{(b_{jk} - a_{jk})^{n_k}}$ with respect to a_k for each $j \in [1, .., p]$ we choose a_{jk} as large as possible however we require $a_{jk} \leq \hat{a}_{jk}$ (otherwise $\frac{\mathbb{I}(a_{jk} \leq \hat{a}_{jk}, b_{jk} \geq \hat{b}_{jk})}{(b_{jk} - a_{jk})^{n_k}}$ = 0 and as a result the likelihood becomes 0). Hence the largest a_{jk} can be is $\hat{a}_{jk}$.

Thus, we choose $a_{jk} = \hat{a}_{jk}$ for each $j \in [1, .., p]$. k was chosen arbitrary, thus we deduce for all k, the maximum likelihood estimate of $a_k = \hat{a}_k = (\hat{a}_{1k}, \ldots, \hat{a}_{p,k})$

Let $k \in G$.

Consider now the maximum likelihood estimator of b_k, which is obtained from maximizing the likelihood in (a. ii) with respect to b_k

$$L(\pi_1, a_1, b_1, \ldots, \pi_k, a_k, b_k | g_1, x_1, .., g_n, x_n) = (\prod_{k=1}^{K} \pi_k^{n_k})(\prod_{k=1}^{K} \prod_{j=1}^{p} \frac{\mathbb{I}(a_{jk} \leq \hat{a}_{jk}, b_{jk} \geq \hat{b}_{jk})}{(b_{jk} - a_{jk})^{n_k}})$$

As discussed previously, Since π_k, n_k are non-negative and $\frac{\mathbb{I}(a_{jk} \leq \hat{a}_{jk}, b_{jk} \geq \hat{b}_{jk})}{(b_{jk} - a_{jk})^{n_k}}$ strictly positive (as given to us in the hint) we can maximize the likelihood with respect to b_k by focusing on maximizing only the terms in the likelihood dependent on b_k.

Thus we focus on maximizing $\prod_{j=1}^{p} \frac{\mathbb{I}(a_{jk} \leq \hat{a}_{jk}, b_{jk} \geq \hat{b}_{jk})}{(b_{jk} - a_{jk})^{n_k}}$ with respect to b_k. Since $\frac{\mathbb{I}(a_{jk} \leq \hat{a}_{jk}, b_{jk} \geq \hat{b}_{jk})}{(b_{jk} - a_{jk})^{n_k}}$ is strictly positive, maximizing $\prod_{j=1}^{p} \frac{\mathbb{I}(a_{jk} \leq \hat{a}_{jk}, b_{jk} \geq \hat{b}_{jk})}{(b_{jk} - a_{jk})^{n_k}}$ with respect to b_k is equivalent to maximizing $\frac{\mathbb{I}(a_{jk} \leq \hat{a}_{jk}, b_{jk} \geq \hat{b}_{jk})}{(b_{jk} - a_{jk})^{n_k}}$ with respect to b_k for each $j \in [1, .., p]$

So we focus on choosing b_k maximizing $\frac{\mathbb{I}(a_{jk} \leq \hat{a}_{jk}, b_{jk} \geq \hat{b}_{jk})}{(b_{jk}-a_{jk})^{n_k}}$ for each $j \in [1, .., p]$

Note as b_{jk} decreases, $(b_{jk} - a_{jk})$ decreases and thus $(b_{jk} - a_{jk})^{n_k}$ decreases (because n_k is a non-negative value), hence $\frac{1}{(b_{jk}-a_{jk})^{n_k}}$ increases.

So, to maximize $\frac{\mathbb{I}(a_{jk} \leq \hat{a}_{jk}, b_{jk} \geq \hat{b}_{jk})}{(b_{jk}-a_{jk})^{n_k}}$ with respect to b_k for each $j \in [1, .., p]$ we choose b_{jk} as small as possible however we require $b_{jk} \geq \hat{b}_{jk}$ (otherwise $\frac{\mathbb{I}(a_{jk} \leq \hat{a}_{jk}, b_{jk} \geq \hat{b}_{jk})}{(b_{jk}-a_{jk})^{n_k}}$ = 0 and as a result the likelihood becomes 0). Hence the smallest b_{jk} can be is $\hat{b}_{jk}$.

Thus, we choose $b_{jk} = \hat{b}_{jk}$ for each $j \in [1, .., p]$. k was chosen arbitrary, thus we deduce for all k, the maximum likelihood estimate of $b_k = \hat{b}_k = (\hat{b}_{1k}, \ldots, \hat{b}_{p,k})$.

Let $k \in G$.

Consider the maximum likelihood estimator of π_k, which is obtained from maximizing the likelihood in (a. ii) with respect to π_k

Since π_k, n_k are non-negative and $\frac{\mathbb{I}(a_{jk} \leq \hat{a}_{jk}, b_{jk} \geq \hat{b}_{jk})}{(b_{jk}-a_{jk})^{n_k}}$ strictly positive (as given to us in the hint) we can maximize the likelihood with respect to π_k by focusing on maximizing only the terms in the likelihood dependent on π_k.

Thus we focus on maximizing $\prod_{k=1}^{K} \pi_k^{n_k}$ [1]. As given in the hint, maximizing this product is equivalent to maximizing its logarithm. As given in the hint:

$log(\prod_{k=1}^{K} \pi_k^{n_k}) = \sum_{k=1}^{K} -1 n_k log \pi_k + n_K log(1 - \sum_{k=1}^{K} -1 \pi_k)$ [2]

To maximize the logarithm, we differentiate [2] with respect to π_k. For $k \neq K$ we get:

$\frac{d}{d\pi_k} log(\prod_{k=1}^{K} \pi_k^{n_k}) = \frac{n_k}{\pi_k} + \frac{n_K}{1-(\sum_{k=1}^{K} -1\pi_k)(-1)}$

We observe $\pi_1 + \ldots + \pi_K = 1 \implies 1 - (\sum_{k=1}^{K} -1 \pi_k) = \pi_K$ hence giving us:

$\frac{d}{d\pi_k} log(\prod_{k=1}^{K} \pi_k^{n_k}) = \frac{n_k}{\pi_k} - \frac{n_K}{\pi_K}$

Setting the derivative to 0 then gives:

$$\begin{aligned}
\frac{d}{d\pi_k} log(\prod_{k=1}^{K} \pi_k^{n_k}) = 0 &\implies \frac{n_k}{\pi_k} - \frac{n_K}{\pi_K} = 0 \\
&\implies \frac{n_k}{\pi_k} = \frac{n_K}{\pi_K} \\
&\implies \pi_k = \frac{n_k \pi_K}{n_K}
\end{aligned}$$

We now use the constraint $\sum_{j=1}^{K} \pi_k = 1 \implies \sum_{j=1}^{K} \frac{n_k \pi_K}{n_K} = 1 \implies \frac{\pi_K}{n_K} \sum_{j=1}^{K} n_k = 1 \implies \frac{\pi_K}{n_K} n = 1 \implies n = \frac{n_K}{\pi_K}$

Thus, since $\pi_k = \frac{n_k \pi_K}{n_K}$ and $n = \frac{n_K}{\pi_K}$ we deduce $\pi_k = \frac{n_k}{n}$

Hence, the maximum likelihood estimate for π_k for $k \neq K$ is $\pi_k = \frac{n_k}{n}$ and thus the maximum likelihood estimate for $pi_K = 1 - \sum_{j=1}^{K} -1 \pi_k = 1 - \frac{n_k}{n} = \frac{n-n_k}{n} = \frac{n_K}{n}$

Thus, for all $k \in G$, the maximum likelihood estimate for π_k is $\frac{n_k}{n}$

Hence we have shown for all $k \in G$

Maximum likelihood estimate of $a_k = \hat{a}_k = (\hat{a}_{1k}, \ldots, \hat{a}_{p,k})$

Maximum likelihood estimate of $b_k = \hat{b}_k = (\hat{b}_{1k}, \ldots, \hat{b}_{p,k})$

Maximum likelihood estimate of π_k is $\frac{n_k}{n}$

Question 1 (b. i)

Loading the data

The code below gives us:

$\hat{\pi}_1 = 0.324$, $\hat{\pi}_2 = 0.345$, $\hat{\pi}_3 = 0.331$

$\hat{a}_1 = (-0.9825579, -0.9925699)$ $\hat{a}_2 = (-0.2496723, -0.2447942)$ $\hat{a}_3 = (-2.471201, -2.495919)$

$\hat{b}_1 = (0.9926797, 0.9975142)$ $\hat{b}_2 = (2.499499, 2.490871)$ $\hat{b}_3 = (0.2489489, 0.2435660)$

```
N = 1000 # there are 1000 observations
K = 3 # there are 3 classes
P = 2 # there are two dimensions
num_1 = 0 # number of observations in class 1
num_2 = 0 # number of observations in class 2
num_3 = 0 # number of observations in class 3
for (i in 1:N) { # for each observed class, find which class this was and increment num_k acc
ordingly
  if (g[i] == 1) {num_1 = num_1 + 1}
  else if (g[i] == 2) {num_2 = num_2 + 1}
  else {num_3 = num_3 + 1}
}
# mle of pi_1, pi_2 and pi_3
pi_1_MLE = num_1 / N
pi_2_MLE = num_2 / N
pi_3_MLE = num_3 / N
print(pi_1_MLE)
```

```
## [1] 0.324
```

```
print(pi_2_MLE)
```

```
## [1] 0.345
```

```
print(pi_3_MLE)
```

```
## [1] 0.331
```

```
pis = c(pi_1_MLE, pi_2_MLE, pi_3_MLE)

# now onto the mle of a_k
mat_as = matrix(c(1,1,1,1,1,1),nrow = 2, ncol = 3,byrow = TRUE) # stores the a_ks in a matrix
where each column refers to the class and each row refers to the dimension (e.g., column 1 an
d row 2 refers to class 1 and dimension 2)
mat_bs = matrix(c(1,1,1,1,1,1),nrow = 2, ncol = 3,byrow = TRUE) # stores the b_ks in a matrix
in the same format as the a_ks

for (k in 1:K) { # for each class
  for (p in 1:P) {
    # for each dimension in the given class
    min = 100000000 # initially sets minimum to be a very large number
    max = -10000000 # initially sets maximum to be a very small number
    for (i in 1:N) { # going through every observation
      if (g[i] == k) { # if the observation belongs to class k
        if (min > X[, i][p]) { # we see if the corresponding x value in the pth dimension is
smaller than the current minimum
          min = X[, i][p] # if so we update the minimum
        }
        if (max < X[, i][p]) { # we see if the corresponding x value in the pth dimension is
larger than the current maximum
          max = X[, i][p] # if so we update the maximum
        }
      }
    }
    mat_as[, k][p] = min # we store the final minimum as the value of a_pk once all observati
ons have been tested
    mat_bs[, k][p] = max # we store the final maximum as the value of b_pk once all observati
ons have been tested
  }
}
print(mat_as)
```

```
##            [,1]       [,2]      [,3]
## [1,] -0.9825579 -0.2496723 -2.471201
## [2,] -0.9925699 -0.2447942 -2.495919
```

```
print(mat_bs)
```

```
##           [,1]     [,2]      [,3]
## [1,] 0.9926797 2.499499 0.2489489
## [2,] 0.9975142 2.490871 0.2435660
```

Question 1 (b.ii)

We firstly begin by creating a function which will return $\hat{G}(x) = \underset{k \in G}{\operatorname{argmax}} \, \pi_k \prod_{j=1}^{p} \frac{\mathbb{1}(a_{j,k} \leq x_j \leq b_{j,k})}{b_{j,k} - a_{j,k}}$

```
classification_func <- function(pis, as, bs, x) { # pass into this function the MLE of pi1, p
i2, pi3, a1, a2, a3 and b1, b2, b3 and the observation to be estimated
  K = 3 # number of classes
  P = 2 # number of dimensions
  max_k = -1 # max_k stores the class which we choose for our observation x, currently this s
tores -1
  max = -1000000 # initially, maximum is a large negative quantity
  for (k in 1:K) {
    product = 1 # stores the product (for which we choose the k maximum) -> for each new clas
s being considered, we reset the product to be 1
    pik = pis[k]  # gets the corresponding MLE estimate for pi_k
    ak = as[, k] # gets the corresponding MLE estimate for a_k
    bk = bs[, k] # gets the corresponding MLE estimate for b_k
    for (p in 1:P) {
      xp = x[p] # gets the pth dimensional value of x
      ap = ak[p] # gets the pth dimensional value of a_k
      bp = bk[p] # gets the pth dimensional value of b_k
      if (ap <= xp) { # checks the identity function and multiplies the product accordingly
        if (xp <= bp) {
          diff = bp - ap
          product = product * (1/diff)
          }
        else {product = 0}
      }
      else {product = 0}
    }
    product = product * pik # once all dimensions covered, multiply product by pi_k
    if (product > max) { # if product exceeds the current maximum value
      max = product # set it to be the new maximum value
      max_k = k # set the maximum k to be the value of k responsible for this new maximum
    }
  }
  return (max_k) # once all classes have been considered, return the maximum k
}
```

We then find an estimate of the probability of classification error under $\hat{G}(x)$ by doing as follows:

1. For each observation x_i we pass this observation into $\hat{G}(x_i)$
2. We then check if $\hat{G}(x_i) = g_i$ and if not, we increment the total number of misclassifications we observe
3. We then estimate the probability of classification error as follows: $\frac{total\ number\ of\ misclassifications}{total\ number\ of\ observations}$

This is done in R as follows, giving us the probability of classification error under $\hat{G}(x)$ being: 0.14

```
N = 1000 # total number of observations
misclassifications = 0 # total number of misclassifications
for (i in 1:N) {
  true_class = g[i] # class of observation x_i
  x = X[, i] # observation x_i
  estimated_class = classification_func(pis, mat_as, mat_bs, x) # estimate of the class obser
vation x comes from using the classification function
  if (true_class != estimated_class) { # if the estimate is incorrect, increment the total nu
mber of misclassifications
    misclassifications = misclassifications + 1
  }
}
prob_error = misclassifications / N # once all observations have been tested, find the probab
ility of classification error
print(prob_error)
```

```
## [1] 0.14
```

Question 2 (a.i)

From lectures we know the optimal values of α and β for linearly separable observations to be a solution to the following constraint optimization problem:

Minimize $\frac{||\beta||^2}{2}$

Subject to $y_i(\alpha + \beta^T x_i) \geq 1$ for $1 \leq i \leq n$

[1]

In R, the solve.QP function solves the following quadratic optimization problem:

Minimize $\frac{1}{2}x^T Ax - b^T x$

Subject to $C^T x \geq d$

[2]

Thus we want to find A, b, C and d such that the quadratic optimization problem in [2] is equivalent to the constraint optimization problem in [1].

We do this as follows:

For positive integer r, let 0_r be the r dimensional zero vector and e_r be the r dimensional vector with all elements 1.

Let

$$
\begin{aligned}
x &= \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \\
A &= diag(0, e_p) \\
b &= \begin{bmatrix} 0 \\ 0_p \end{bmatrix}
\end{aligned}
$$

Note A denotes the (p+1) x (p+1) matrix where the first diagonal element is 0 and the other diagonal elements are 1

We have then:

$$
\begin{aligned}
\frac{1}{2}x^T Ax + b^T x &= \frac{1}{2}\begin{bmatrix} \alpha \\ \beta \end{bmatrix}^T diag(0, e_p)\begin{bmatrix} \alpha \\ \beta \end{bmatrix} + [0, 0_p]\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \\
&= \frac{1}{2}\begin{bmatrix} \alpha \\ \beta \end{bmatrix}^T \begin{bmatrix} 0 \\ \beta \end{bmatrix}^T + 0 \\
&= \frac{1}{2}[\alpha, \beta^T]\begin{bmatrix} 0 \\ \beta \end{bmatrix} + 0 \\
&= \frac{1}{2}\beta^T\beta \\
&= \frac{1}{2}||\beta||^2
\end{aligned}
$$

[3]

Let Y, X, C and d be the matrices and vertices defined by

$$Y = diag(y_1, \ldots, y_n), X = [x_1, \ldots, x_n], C = \begin{bmatrix} e_n^T \\ X \end{bmatrix} Y, d = e_n$$

Now consider

$$
\begin{aligned}
C^T x - d &= Y[e_n, X^T] \begin{bmatrix} \alpha \\ \beta \end{bmatrix} - e_n \\
&= diag(y_1, \ldots, y_n) \begin{pmatrix} \alpha + x_1^T \beta \\ \vdots \\ \alpha + x_n^T \beta \end{pmatrix} - e_n \\
&= \begin{pmatrix} y_1(\alpha + x_1^T \beta) \\ \vdots \\ y_n(\alpha + x_n^T \beta) \end{pmatrix} - e_n \\
&= \begin{pmatrix} y_1(\alpha + x_1^T \beta) - 1 \\ \vdots \\ y_n(\alpha + x_n^T \beta) - 1 \end{pmatrix}
\end{aligned}
$$

[4]

Using [3] and [4] we deduce that constraint optimization problem [1] can be written as quadratic optimization problem [2].

We now use the code below and the R function solve.QP, with our values of A, b, C and d to find optimal values of α and β

We find the optimal value of α to be 0.0210844

The optimal value of β to be [-0.5717356 , -0.5299126]

```
library(quadprog) # loads the library
load("ClassificationQ2a.RData") # loads the data

n <- 250
p <- 2

A <- matrix(0 , nrow = p +1 , ncol = p +1)
A <- A + 10^-4 * diag(p+1) # A is positive semi-definite, so we replace it as told in the rem
ark
A[2:( p +1) ,2:( p +1) ] <- diag(p)
b <- numeric(p +1)
Y <- diag(y)
Z <- matrix(0 , nrow = p +1 , ncol = n)
Z[1 ,] <- rep(1 , n)
Z[2:( p +1) ,] <- X
C <- Z %*% Y
d <- rep (1 , n )
W <- solve.QP(A ,b ,C ,d , meq = 0)
alpha <- W$solution [1]
beta <- W$solution [2:( p +1) ]
print(alpha)
```

```
## [1] 0.0210844
```

```
print(beta)
```

```
## [1] -0.5717356 -0.5299126
```

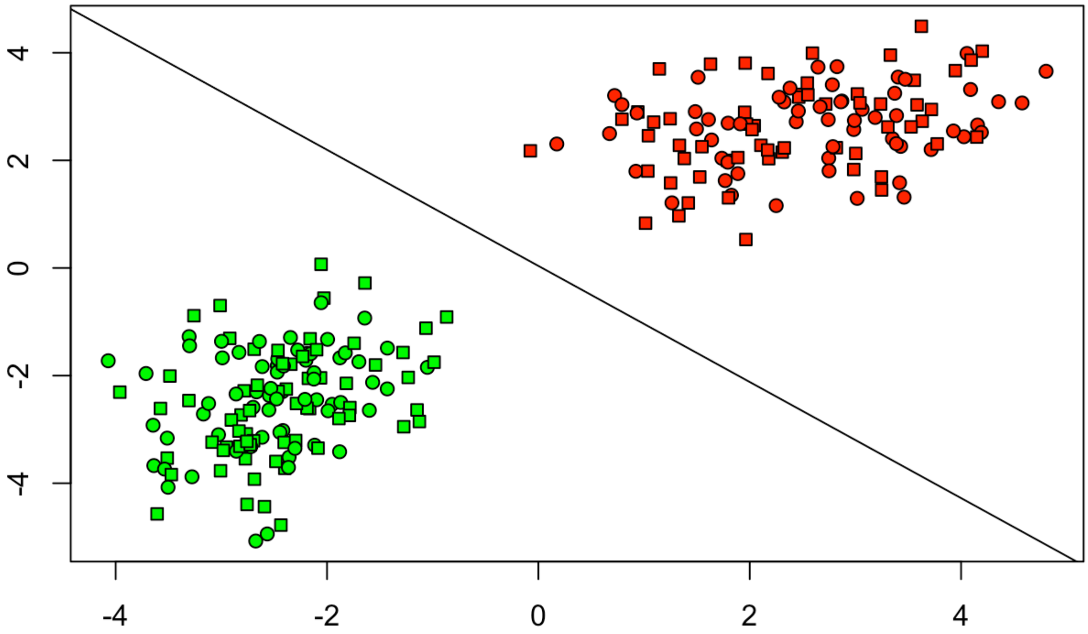
Question 2 (a .ii)

```
g = rep(0, n)
for (i in 1:n) {
  if (y[i] == 1) {g[i] = 2}
  else {g[i] = 1}
} # g = 1 if y = -1 and g = 2 if y = 1

plot (t(X) , bg = c ("red" ,"green")[g] , pch = c (21 ,22), xlab = " " , ylab = " ") # plot o
bservations, using g to define the plots with y=-1 as red and the plots with y=1 as green

boundary.line <- function ( alpha , beta ) {
abline(-alpha /beta[2] , -beta [1]/beta [2])}

boundary.line (alpha , beta) # plots the hyperplane
```



Question 2 (b. i)

As discussed in the lectures, optimal values of α and β for non linearly seperable observations are solutions to the following constraint optimisation problem:

Minimize $\frac{||\beta||^2}{2} + \lambda \sum_{i=1}^{n} \epsilon_i$

Subject to $y_i(\alpha + \beta^T x_i) + \epsilon_i \geq 1, \epsilon_i \geq 0$ for $1 \leq i \leq n$

[1]

where $\epsilon_i = max(1 - y_i(\alpha + \beta^T x_i), 0)$

In R, the solve.QP function solves the following quadratic optimization problem:

Minimize $\frac{1}{2} x^T A x - b^T x$

Subject to $C^T x \geq d$

[2]

Thus we want to find A, b, C and d such that the quadratic optimization problem in [2] is equivalent to the constraint optimization problem in [1].

We do this as follows:

For positive integers r, s let $0_r, 0_{r,s} and \mathbb{I}_r$ be the r dimensional 0 vector, rxs 0 matrix and rxr unit matrix. Let e_r be the r dimensional vector whose elements are all 1. Let $\epsilon = (\epsilon_1, \dots, \epsilon_n)$

Let Let

$$x = \begin{bmatrix} \alpha \\ \beta \\ \epsilon \end{bmatrix}$$
$$A = diag(0, e_p, 0_n)$$
$$b = \begin{bmatrix} 0 \\ 0_p \\ -\lambda e_n \end{bmatrix}$$

A denotes the (p+n+1) x (p+n+1) diagonal matrix whose first diagonal element and last n diagonal elements are 0 and whose other diagonal entries are 1.

We have then:

$$\frac{1}{2}x^T Ax - b^T x = \frac{1}{2}\begin{bmatrix} \alpha \\ \beta \\ \epsilon \end{bmatrix}^T diag(0, e_p, 0_n)\begin{bmatrix} \alpha \\ \beta \\ \epsilon \end{bmatrix} - \begin{bmatrix} 0 \\ 0_p \\ -\lambda e_n \end{bmatrix}^T \begin{bmatrix} \alpha \\ \beta \\ \epsilon \end{bmatrix}$$
$$= \frac{1}{2}\begin{bmatrix} \alpha \\ \beta \\ \epsilon \end{bmatrix}^T \begin{bmatrix} 0 \\ \beta \\ 0 \end{bmatrix}^T - \begin{bmatrix} 0 \\ 0_p \\ -\lambda e_n \end{bmatrix}^T \begin{bmatrix} \alpha \\ \beta \\ \epsilon \end{bmatrix}$$
$$= \frac{1}{2}[\alpha, \beta^T, \epsilon^T]\begin{bmatrix} 0 \\ \beta \\ \epsilon \end{bmatrix} - \begin{bmatrix} 0 \\ 0_p \\ -\lambda e_n \end{bmatrix}^T \begin{bmatrix} \alpha \\ \beta \\ \epsilon \end{bmatrix}$$
$$= \frac{1}{2}\beta^T \beta + \begin{bmatrix} 0 \\ 0_p \\ -\lambda e_n \end{bmatrix}^T \begin{bmatrix} \alpha \\ \beta \\ \epsilon \end{bmatrix}$$
$$= \frac{1}{2}||\beta||^2 - \begin{bmatrix} 0 \\ 0_p \\ -\lambda e_n \end{bmatrix}^T \begin{bmatrix} \alpha \\ \beta \\ \epsilon \end{bmatrix}$$
$$= \frac{1}{2}||\beta||^2 - [0, 0_p^T, -\lambda e_n^T]\begin{bmatrix} \alpha \\ \beta \\ \epsilon \end{bmatrix}$$
$$= \frac{1}{2}||\beta||^2 - (-\lambda e_n^T \epsilon)$$
$$= \frac{1}{2}||\beta||^2 + \lambda \sum_{i=1}^{n} \epsilon_i$$

[3]

Let

$$Y = diag(y_1, \ldots, y_n), X = [x_1, \ldots, x_n], B = \begin{bmatrix} e_n^T \\ X \end{bmatrix} Y, C = \begin{bmatrix} B & 0_{p+1,n} \\ I_n & I_n \end{bmatrix}, d = \begin{bmatrix} e_n \\ 0_n \end{bmatrix}$$

Consequently we get

$$\begin{aligned} B^T \begin{bmatrix} \alpha \\ \beta \end{bmatrix} &= Y[e_n, X^T] \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \\ &= Y \begin{bmatrix} \alpha + x_1^T \beta \\ \vdots \\ \alpha + x_n^T \beta \end{bmatrix} \\ &= diag(y_1, .., y_n) \begin{bmatrix} \alpha + x_1^T \beta \\ \vdots \\ \alpha + x_n^T \beta \end{bmatrix} \\ &= \begin{bmatrix} y_1 \ + \alpha + x_1^T \beta \\ \vdots \\ y_n \ + \alpha + x_n^T \beta \end{bmatrix} \end{aligned}$$

Now consider

$$\begin{aligned} C^T x - d &= \begin{bmatrix} B^T & I_n \\ 0_{p+1,n} & I_n \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \epsilon \end{bmatrix} - \begin{bmatrix} e_n \\ 0_n \end{bmatrix} \\ &= \begin{bmatrix} B^T \begin{bmatrix} \alpha \\ \beta \end{bmatrix} + \epsilon \\ 0 + \epsilon \end{bmatrix} - \begin{bmatrix} e_n \\ 0_n \end{bmatrix} \\ &= \begin{bmatrix} B^T \begin{bmatrix} \alpha \\ \beta \end{bmatrix} + \epsilon - \epsilon_n \\ \epsilon - 0_n \end{bmatrix} \\ &= \begin{bmatrix} y_1 \ + \alpha + \beta^T x_1 \\ \vdots \\ y_n \ + \alpha + \beta^T x_n \\ \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix} \end{aligned}$$

[4]

Using [3] and [4] we deduce that constraint optimization problem [1] can be written as quadratic optimization problem [2].

We now use the code below and the R function solve.QP, with our values of A, b, C and d to find optimal values of α and β.

We find the optimal value of α to be 0.2127734

The optimal value of β to be [-0.6585156 , -0.5304993]

```r
library(quadprog) # loads the library
load("ClassificationQ2b.RData") # loads the data
p <- 2
n <- 250
lambda <- 3

A <- matrix(0 , nrow = p +1+n , ncol = p+1+n)
A[2:(p+1), 2:(p+1)] <- diag(p)
A <- A + 10^-4 * diag(p+1+n)


b <- rep(0, p+1+n)
b[(p+2):(n+p+1)] <- -lambda
Y <- diag(y)
Z <- matrix(0 , nrow = p +1 , ncol = n)
Z[1 ,] <- rep(1 , n)
Z[2:( p +1) ,] <- X
B <- Z %*% Y
C <- matrix(0 , nrow = p+1+n , ncol=2*n)
C[1:(p+1) , 1:n] <- B
C[(p+2):(p+1+n) , 1:n]<- diag(n)
C[(p+2):(p+1+n), (n+1):(2*n)] <- diag(n)
d <- rep(0, 2*n)
d[1:n] <- 1

W <- solve.QP(A ,b ,C ,d , meq = 0)
alpha <- W$solution [1]
beta <- W$solution [2:( p +1) ]
print(alpha)
```

```
## [1] 0.2127734
```

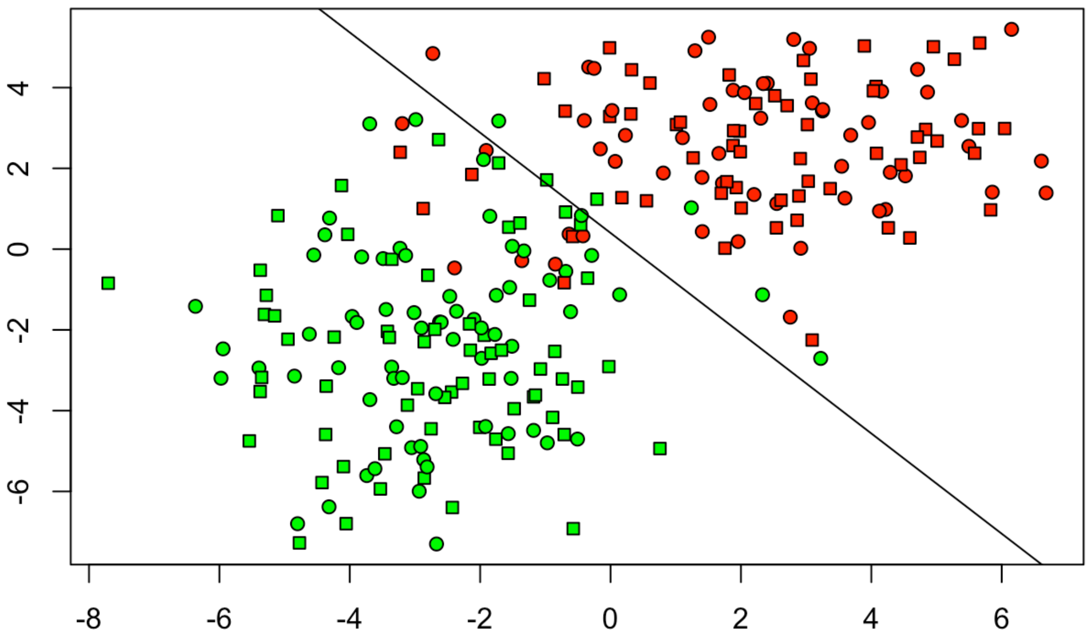
```r
print(beta)
```

```
## [1] -0.6585156 -0.5304993
```

Question (2 b.ii)

```r
g = rep(0, n)
for (i in 1:n) {
  if (y[i] == 1) {g[i] = 2}
  else {g[i] = 1}
} # g = 1 if y = -1 and g = 2 if y = 1

plot(t(X) , bg = c ("red" ,"green")[g] , pch = c (21 ,22), xlab = " " , ylab = " ") # plot ob
servations, using g to define the plots with y=-1 as red and the plots with y=1 as green
abline(-alpha/beta[2] , -beta [1]/beta [2])
```



4
2
0
-2
-4
-6
-8
-6
-4
-2
0
2
4
6