CMPE 544: Pattern Recognition (Fall 2020)

Homework1 — Yaşar Alim Türkmen - 2019700123

1. (a) The average probability of error can be calculated by

$$P(error) = \int P(error|x)p(x)\,dx.$$

We know that $P(error|x) = \min\{P(\omega_1|x), P(\omega_2|x)\}$. If we decide ω_1 for $x > k$; the conditional error becomes $P(\omega_1|x)$ for $x < k$, and $P(\omega_2|x)$ for $x > k$. Then the average probability of error becomes

$$\begin{cases} \int P(\omega_1|x)p(x)\,dx & x < k, \\ \int P(\omega_2|x)p(x)\,dx & x > k \end{cases}$$

$$= \int_{-\infty}^{k} P(\omega_1|x)p(x)\,dx + \int_{k}^{\infty} P(\omega_2|x)p(x)\,dx$$

If we substitute $P(\omega_1|x)p(x)$ for $P(\omega_1)p(x)p(x|\omega_1)$ from Bayes' rule, we get

$$\int_{-\infty}^{k} P(\omega_1)p(x|\omega_1)\,dx + \int_{k}^{\infty} P(\omega_2)p(x|\omega_2)\,dx$$

$$= P(\omega_1)\int_{-\infty}^{k} p(x|\omega_1)\,dx + P(\omega_2)\int_{k}^{\infty} p(x|\omega_2)\,dx$$

(b) We can rewrite $P(error)$ as

$$P(error) = P(\omega_1)\int_{-\infty}^{k} p(x|\omega_1)\,dx - P(\omega_2)\int_{\infty}^{k} p(x|\omega_2)\,dx$$

To find k value which minimizes $P(error)$, we can take the derivative of $P(error)$ with respect to k.

$$\frac{d(P(error))}{dk} = P(\omega_1)p(k|\omega_1) - P(\omega_2)p(k|\omega_2)$$

If we apply Bayes' rule and make it equal to zero to find minimizing k value, we get

$$P(\omega_1|k)p(k) - P(\omega_2|k)p(k) = 0$$

Therefore, k should satisfy

$$P(\omega_1|k) = P(\omega_2|k).$$

2. (a) Define critical region as

$$C_\alpha = [x \mid \frac{p(x \mid \omega_1)}{p(x \mid \omega_2)} \geq k_\alpha].$$

Since the class-conditional densities are Gaussian and variances are equal we can write the same statement as

$$C_\alpha = \left[x \mid \frac{\frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{1}{2}(\frac{x-\mu_1}{\sigma})^2)}{\frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{1}{2}(\frac{x-\mu_0}{\sigma})^2)} \geq k_\alpha \right].$$

If we reduce it and take ln of both side, we get:

$$C_\alpha = [x \mid \ln(\exp(-\frac{1}{2}\left(\frac{x-\mu_1}{\sigma}\right)^2)) - \ln(\exp(-\frac{1}{2}\left(\frac{x-\mu_0}{\sigma}\right)^2)) \geq \ln(k_\alpha)]$$

$$= \left[\frac{1}{2}\left(\frac{x-\mu_0}{\sigma}\right)^2 - \frac{1}{2}\left(\frac{x-\mu_1}{\sigma}\right)^2 \geq \ln(k_\alpha)\right]$$

$$= \left[\frac{x^2 - 2x\mu_0 + \mu_0^2 - x^2 + 2x\mu_1 - \mu_1^2}{2\sigma^2} \geq \ln(k_\alpha)\right]$$

$$= \left[x(2\mu_1 - 2\mu_0) \geq \ln(k_\alpha)2\sigma^2 + \mu_1^2 - \mu_0^2\right]$$

Finally, we can define the critical region as

$$C_\alpha = \left[x \mid x \geq x_0, \quad x_0 = \frac{\ln(k_\alpha)\sigma^2}{\mu_1 - \mu_0} + \mu_1 + \mu_0\right]$$

(b) After threshold value, x_0, is found then,

$$\alpha = \int_{x_0}^{\infty} p(x \mid \omega_0)dx.$$

We can write the same equation as

$$\alpha = \int_{x_0}^{\infty} \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{1}{2}(\frac{x-\mu_0}{\sigma})^2)dx.$$

Let $t = \frac{x-\mu_0}{\sigma\sqrt{2}}$. Then dx becomes $\sqrt{2}\sigma dt$ and the lower limit of the integral is changed as $\frac{x_0-\mu_0}{\sigma\sqrt{2}}$.

$$\alpha = \int_{\frac{x_0-\mu_0}{\sigma\sqrt{2}}}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-t^2}\sqrt{2}\sigma dt$$

$$\alpha = \frac{1}{2}\left(\frac{2}{\sqrt{\pi}}\int_0^{\infty} e^{-t^2}dt - \frac{2}{\sqrt{\pi}}\int_0^{\frac{x_0-\mu_0}{\sigma\sqrt{2}}} e^{-t^2}dt\right)$$

$$\alpha = \frac{1}{2}\left(erf(\infty) - erf\left(\frac{x_0-\mu_0}{\sigma\sqrt{2}}\right)\right)$$

Therefore, for a given α value x_0 can be written as

$$2\alpha = \left(1 - erf\left(\frac{x_0-\mu_0}{\sigma\sqrt{2}}\right)\right)$$

$$\frac{x_0 - \mu_0}{\sigma\sqrt{2}} = erf^{-1}(1 - 2\alpha)$$

$$x_0 = erf^{-1}(1 - 2\alpha)\sigma\sqrt{2} + \mu_0$$

In order to find power of the test, we should find $\int_{x_0}^{\infty} p(x \mid \omega_1)dx$. If we follow the similar steps above, we obtain

$$(power) 1 - \beta = \frac{1}{2}\left(erf(\infty) - erf\left(\frac{x_0 - \mu_1}{\sigma\sqrt{2}}\right)\right)$$

Replacing x_0 with $(erf^{-1}(1 - 2\alpha)\sigma\sqrt{2} + \mu_0)$ gives

$$1 - \beta = \frac{1}{2}\left(1 - erf\left(erf^{-1}(1 - 2\alpha) + \frac{\mu_0 - \mu_1}{\sigma\sqrt{2}}\right)\right).$$

3. (a) Define a discrimination function $g_i(x)$ which equals to $-R(\alpha_i|x)$. We calculate the risk by

$$R(\alpha_i|x) = \sum_{j=1}^{c} \lambda(\alpha_i|\omega_j)P(\omega_j|x).$$

If we rewrite this with the given loss function, we obtain the following risk function:

$$R(\alpha_i|x) = \begin{cases} \sum_{j=1}^{c} \lambda_s P(\omega_j|x) & i = 1,\dots,c \quad j \neq i \\ \sum_{j=1}^{c} \lambda_r P(\omega_j|x) & i = c+1 \end{cases}$$

Since λ_s and λ_r are just multipliers and $\sum_{j=1}^{c} P(\omega_j|x) = 1$,

$$R(\alpha_i|x) = \begin{cases} \lambda_s(1 - P(\omega_i|x)) & i = 1,\dots,c \\ \lambda_r & i = c+1 \end{cases}.$$

Now, substitute $-R(\alpha_i|x)$ to find $g_i(x)$.

$$g_i(x) = \begin{cases} \lambda_s(P(\omega_i|x) - 1) & i = 1,\dots,c \\ -\lambda_r & i = c+1 \end{cases}.$$

Divide each side by λ_s and add 1.

$$g_i(x) = \begin{cases} P(\omega_i|x) & i = 1,\dots,c \\ -\frac{\lambda_r}{\lambda_s} + 1 & i = c+1 \end{cases}$$

From Bayes' rule, $P(\omega_i|x) = \frac{P(\omega_i)p(x|\omega_i)}{p(x)}$. By applying Bayes' rule and multiplying both part with $p(x)$

$$g_i(x) = \begin{cases} P(\omega_i)p(x|\omega_i) & i = 1,\dots,c \\ \frac{\lambda_s - \lambda_r}{\lambda_s} p(x) & i = c+1 \end{cases}.$$

Finally, $p(x) = \sum_j^c P(\omega_j)p(x|\omega_j)$

$$g_i(x) = \begin{cases} P(\omega_i)p(x|\omega_i) & i = 1,\dots,c \\ \frac{\lambda_s - \lambda_r}{\lambda_s} \sum_j^c P(\omega_j)p(x|\omega_j) & i = c+1 \end{cases}$$

(b) Plots of the discriminant functions and decision regions can be seen below.

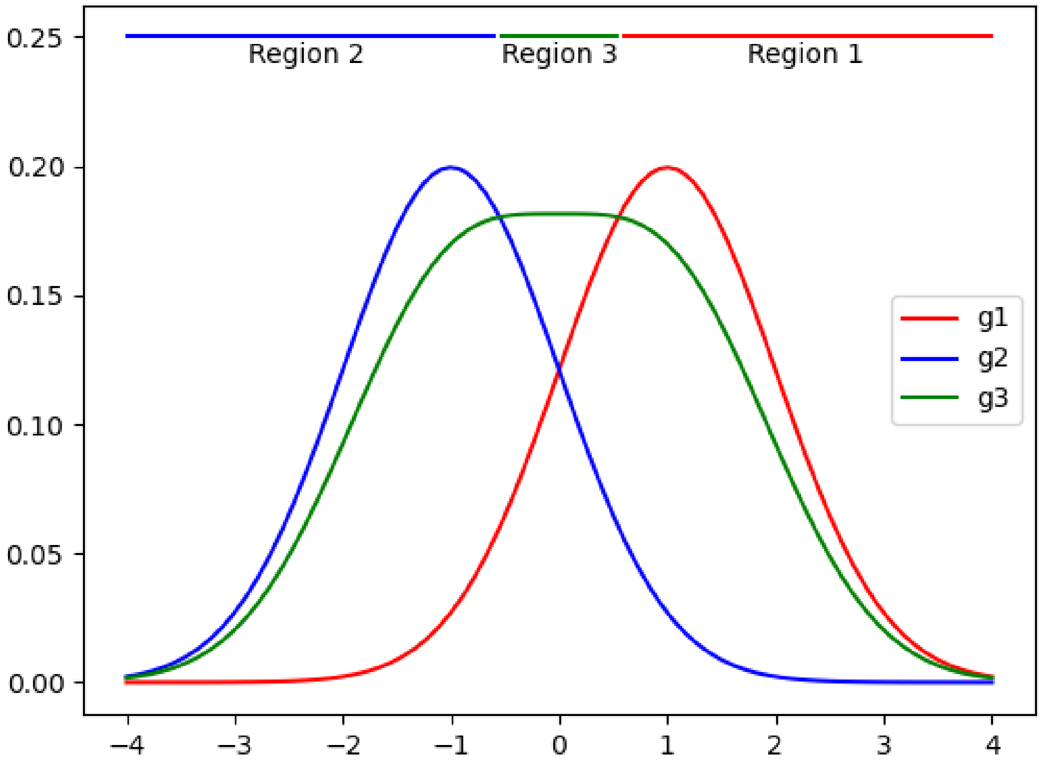

Figure 1: Discriminant functions and decision regions

The following script is used for plotting:

```
import matplotlib.pyplot as plt
import numpy as np
import scipy.stats as stats
import math

mu1 = 1
mu2 = -1
var = 1
sigma = math.sqrt(var)
lambda_s, lambda_r = 4, 1
coeff = (lambda_s-lambda_r) / lambda_s
p_omega= 0.5

x = np.linspace(mu2 - 3*sigma, mu1 + 3*sigma, 100)

y1 = stats.norm.pdf(x, mu1, sigma)*p_omega
```

```
y2 = stats.norm.pdf(x, mu2, sigma)*p_omega
y3 = (y1+y2)*coeff

r1, r2, r3 = [], [], []
for i in range(len(x)):
    if y1[i] > y2[i] and y1[i] > y3[i] : r1.append(x[i])
    elif y2[i] > y3[i]: r2.append(x[i])
    else: r3.append(x[i])

r1_line = np.full((len(r1), ), 0.25)
r2_line = np.full((len(r2), ), 0.25)
r3_line = np.full((len(r3), ), 0.25)

plt.plot(x, y1, color=’r’, label=’g1’)
plt.plot(r1, r1_line, ’r-’)
plt.text(r1[int(len(r1)/3)], 0.24, ’Region 1’)
plt.plot(x, y2, color=’b’, label=’g2’)
plt.plot(r2, r2_line, ’b-’)
plt.text(r2[int(len(r2)/3)], 0.24, ’Region 2’)
plt.plot(x, y3, color=’g’, label=’g3’)
plt.plot(r3, r3_line, ’g-’)
plt.text(r3[0], 0.24, ’Region 3’)
plt.legend()

plt.show()
```