University of Tehran

School of Electrical and Computer Engineering

Machine Learning

Homework 1 : Bayesian Decision Theory

Author:
Alireza Javid

Student Number:
810198375

Contents

1 Problem 1: Bayesian Decision for Cauchy Distribution

$$P(x \mid \omega_i) = \frac{1}{\pi b} \frac{1}{1 + \left(\frac{x-a_i}{b}\right)^2} \quad i = 1,2 \quad a_2 > a_1$$

(a) From the Bayes formula we have:

$$P(\omega_i \mid x) = \frac{p(x \mid \omega_i) p(\omega_i)}{p(x)}$$

We assume $P(\omega_1) = P(\omega_2)$. Thus for finding x where $P(\omega_1 \mid x) = P(\omega_2 \mid x)$ we have:

$$P(\omega_1 \mid x) = P(\omega_2 \mid x) \rightarrow \frac{p(x \mid \omega_1) p(\omega_1)}{p(x)} = \frac{p(x \mid \omega_2) p(\omega_2)}{p(x)} = p(x \mid \omega_1) = p(x \mid \omega_2)$$

$$\rightarrow \frac{1}{\pi b} \frac{1}{1 + \left(\frac{x-a_1}{b}\right)^2} = \frac{1}{\pi b} \frac{1}{1 + \left(\frac{x-a_2}{b}\right)^2}$$

$$\rightarrow \left(\frac{x-a_1}{b}\right)^2 = \left(\frac{x-a_2}{b}\right)^2 \rightarrow \frac{x-a_1}{b} = -\frac{x-a_2}{b} \rightarrow x = \frac{a_1+a_2}{2}$$

Now we can plot $P(\omega_i \mid x)$ in python: (Note that in this problem evidence can be calculated with the law of total probability.)

$$P(x) = \sum_{i=1}^{2} P(x \mid \omega_i) P(\omega_i)$$

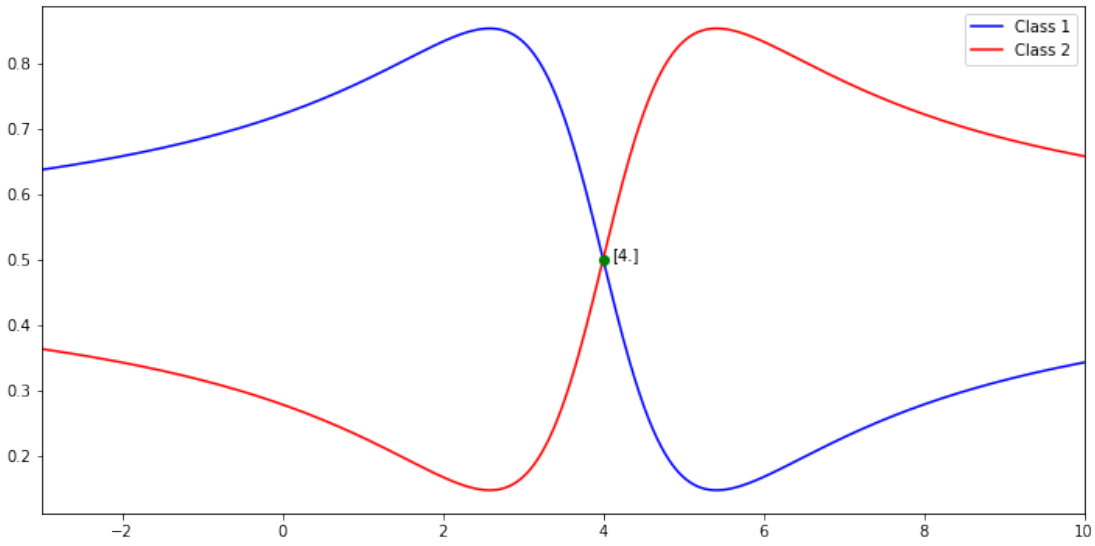

Figure 1: $P(\omega_i \mid x)$ Plot for $(a_1 = 3, a_2 = 5, b = 1)$

(b) Look at the picture below for $P(x \mid \omega_i)P(\omega_i)$. We must calculate the selected area to find the total error.

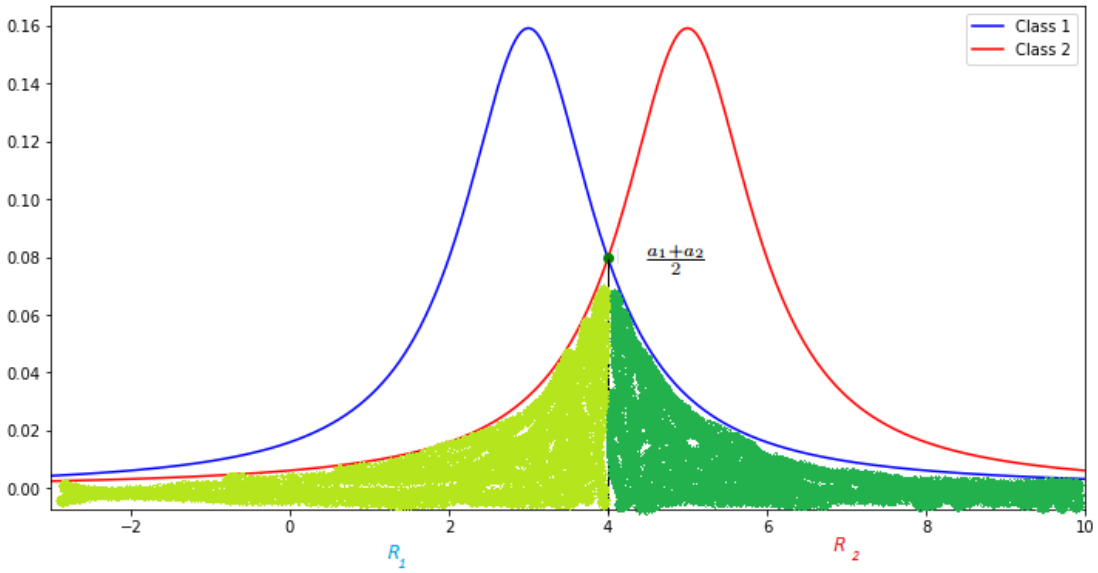

Figure 2: Selected area in $P(x \mid \omega_i)P(\omega_i)$ plot, indicates our total error

$$P(Error) = P(x \in R_1, \omega_2) + P(x \in R_2, \omega_1)$$

$$P(Error) = \int_{-\infty}^{\frac{a_1+a_2}{2}} P(x|\omega_2)P(\omega_2)dx + \int_{\frac{a_1+a_2}{2}}^{\infty} P(x|\omega_1)P(\omega_1)dx$$

$$P(Error) = \int_{-\infty}^{\frac{a_1+a_2}{2}} \frac{1}{\pi b}\frac{1}{1+\left(\frac{x-a_2}{b}\right)^2} \times 0.5\,dx + \int_{\frac{a_1+a_2}{2}}^{\infty} \frac{1}{\pi b}\frac{1}{1+\left(\frac{x-a_1}{b}\right)^2} \times 0.5\,dx$$

$$P(Error) = \frac{0.5}{\pi}tan^{-1}(\frac{x-a_2}{b})\Big|_{-\infty}^{\frac{a_1+a_2}{2}} + \frac{0.5}{\pi}tan^{-1}(\frac{x-a_1}{b})\Big|_{\frac{a_1+a_2}{2}}^{\infty}$$

$$= \frac{1}{2\pi}tan^{-1}(\frac{a_1-a_2}{b}) + \frac{1}{4} + \frac{1}{4} + \frac{1}{2\pi}tan^{-1}(\frac{a_2-a_1}{b})$$

$$P(Error) = \frac{1}{2} - \frac{1}{\pi}tan^{-1}\left|\frac{a_2-a_1}{b}\right|$$

(c) We know $tan^{-1}(.)$ for the positive input is between 0 and 1. So for maximizing $P(Error)$ we need to have 0 in $tan^{-1}(.)$ expression. Thus Two possible cases are:

$$\begin{cases} a_1 = a_2 \\ b \to \infty \end{cases}$$

(d) To perform Bayesian classification, we need to compute the posterior probability of the new observation belonging to each class. If the prior probabilities of the two classes are equal, we only need to compute the likelihood.

$$\omega_i = \begin{cases} \omega_1 & \frac{P(x|\omega_1)}{P(x|\omega_2)} > 1 \\ \omega_2 & o.w. \end{cases}$$

The decision boundary for this classifier is x where $\frac{P(x|\omega_1)}{P(x|\omega_2)} = 1$. So to derive this boundary we have:

$$P(x \mid \omega_1) = P(x \mid \omega_2)$$

$$\rightarrow \frac{1}{\pi b}\frac{1}{1+\left(\frac{x-a_1}{b}\right)^2} = \frac{1}{\pi b}\frac{1}{1+\left(\frac{x-a_2}{b}\right)^2} \rightarrow \mid x-a_1 \mid = \mid x-a_2 \mid$$

$$\begin{cases} a_1 = a_2 \rightarrow \text{all } x \text{ is decision boundary} \\ a_1 \neq a_2 \rightarrow x = \frac{a_1+a_2}{2} \end{cases}$$

To calculate error probability, we can use our results in part b :

$$P(Error) = \frac{1}{2} - \frac{1}{\pi} tan^{-1}\left|\frac{a_2-a_1}{b}\right| \rightarrow P(Error) = \begin{cases} \frac{1}{2} & a_1 = a_2 \\ \frac{1}{2} - \frac{1}{\pi} tan^{-1}\left|\frac{a_2-a_1}{b}\right| & a_1 \neq a_2 \end{cases}$$

And we can show the decision boundary below:

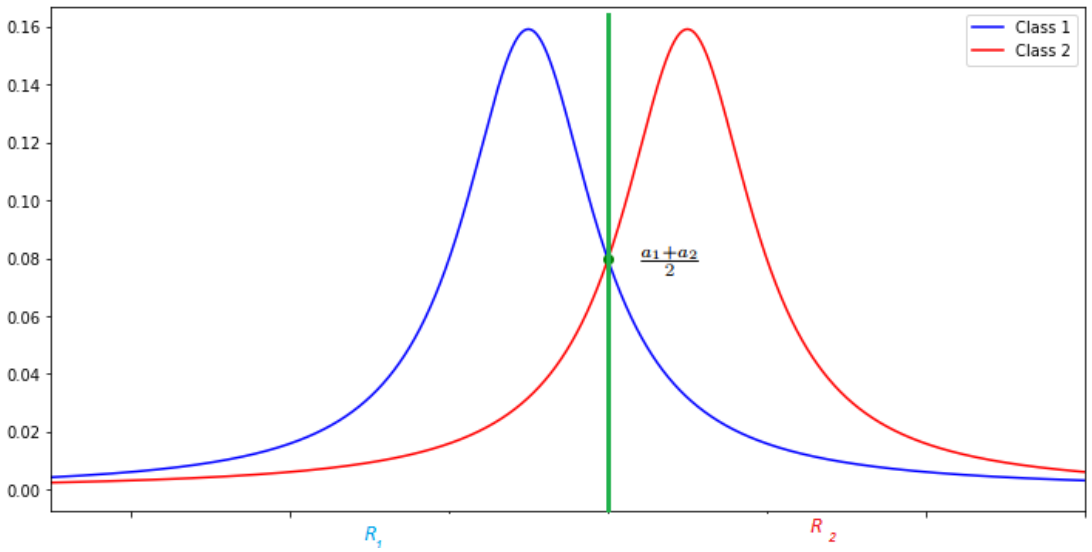

Figure 3: Two areas are separated by the decision boundary

(e) To design a classifier for Risk minimization, we use the below equation for decision boundary:

$$\frac{P(x \mid \omega_1)}{P(x \mid \omega_2)} > \frac{\lambda_{11}-\lambda_{12}}{\lambda_{22}-\lambda_{21}} = \frac{1}{2}$$

$$\omega_i = \begin{cases} \omega_1 & \frac{P(x|\omega_1)}{P(x|\omega_2)} > \frac{1}{2} \\ \omega_2 & o.w. \end{cases}$$

The decision boundary for this classifier is x where $\frac{P(x|\omega_1)}{P(x|\omega_2)} = \frac{1}{2}$. So to derive this boundary we have:

$$P(x \mid \omega_1) = \frac{1}{2} \times P(x \mid \omega_2)$$

$$\rightarrow \frac{1}{\pi b}\frac{1}{1+\left(\frac{x-a_1}{b}\right)^2} = \frac{1}{2} \times \frac{1}{\pi b}\frac{1}{1+\left(\frac{x-a_2}{b}\right)^2} \rightarrow 1+2(\frac{x-a_2}{b})^2 = (\frac{x-a_1}{b})^2$$

$$x^2 - 2(2a_2 - a_1)x + 2a_2^2 - a_1^2 + b^2 = 0$$

$$x = 2a_2 - a_1 \pm \sqrt{2a_2^2 + 2a_1^2 - 4a_2a_1 - b^2}$$

$$x_1 = 2a_2 - a_1 - \sqrt{2a_2^2 + 2a_1^2 - 4a_2a_1 - b^2}, x_2 = 2a_2 - a_1 + \sqrt{2a_2^2 + 2a_1^2 - 4a_2a_1 - b^2}$$

And we can show the decision boundary below:

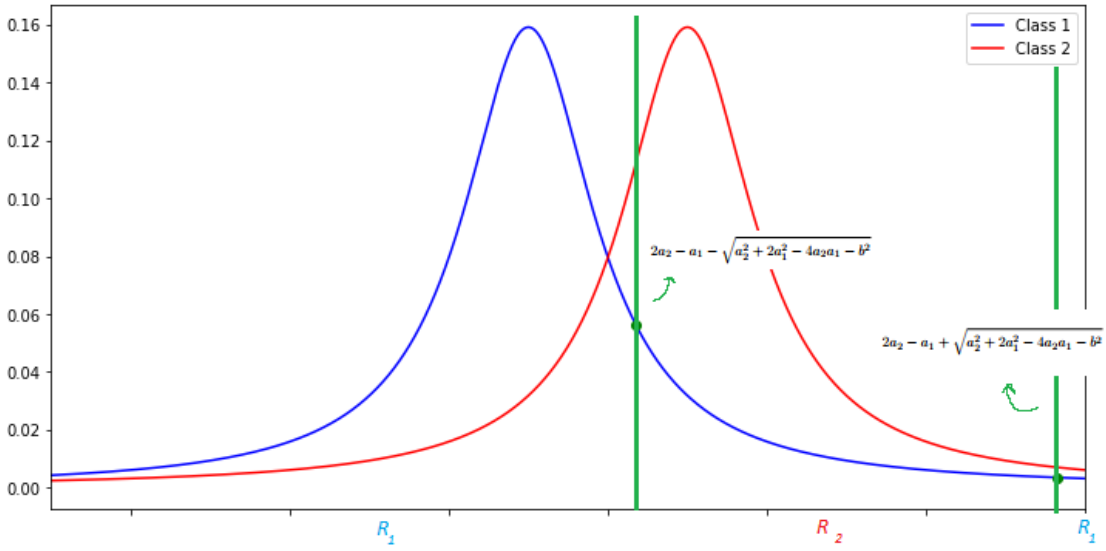

Figure 4: Two areas are separated by the decision boundary

As we expect $\lambda_{21} > \lambda_{12}$ implies increasing the risk of choosing R_2 over R_1 and it makes R_1 more favorable.
Now we calculate error probability as below:

$$P(Error) = P(x \in R_1, \omega_2) + P(x \in R_2, \omega_1)$$

$$P(Error) = \int_{-\infty}^{x_1} P(x|\omega_2)P(\omega_2)dx + \int_{x_1}^{x_2} P(x|\omega_1)P(\omega_1)dx + \int_{x_2}^{\infty} P(x|\omega_2)P(\omega_2)dx$$

$$P(Error) = \int_{-\infty}^{x_1} \frac{1}{\pi b} \frac{1}{1 + \left(\frac{x-a_2}{b}\right)^2} \times 0.5\, dx + \int_{x_1}^{x_2} \frac{1}{\pi b} \frac{1}{1 + \left(\frac{x-a_1}{b}\right)^2} \times 0.5\, dx$$

$$+ \int_{x_2}^{\infty} \frac{1}{\pi b} \frac{1}{1 + \left(\frac{x-a_2}{b}\right)^2} \times 0.5\, dx$$

$$P(Error) = \frac{0.5}{\pi} tan^{-1}\left(\frac{x-a_2}{b}\right)\Big|_{-\infty}^{x_1} + \frac{0.5}{\pi} tan^{-1}\left(\frac{x-a_1}{b}\right)\Big|_{x_1}^{x_2} + \frac{0.5}{\pi} tan^{-1}\left(\frac{x-a_2}{b}\right)\Big|_{x_2}^{\infty}$$

$$= \frac{1}{2\pi} tan^{-1}\left(\frac{x_1-a_2}{b}\right) + \frac{1}{4} + \frac{1}{2\pi} tan^{-1}\left(\frac{x_2-a_1}{b}\right) - \frac{1}{2\pi} tan^{-1}\left(\frac{x_1-a_1}{b}\right) + \frac{1}{4} - \frac{1}{2\pi} tan^{-1}\left(\frac{x_2-a_2}{b}\right)$$

$$P(Error) = \frac{1}{2} + \frac{1}{2\pi}\left(tan^{-1}\left(\frac{x_1-a_2}{b}\right) + tan^{-1}\left(\frac{x_2-a_1}{b}\right) - tan^{-1}\left(\frac{x_1-a_1}{b}\right) - tan^{-1}\left(\frac{x_2-a_2}{b}\right)\right)$$

The total error in this case also can be shown below: (It's equal to the value derived earlier)

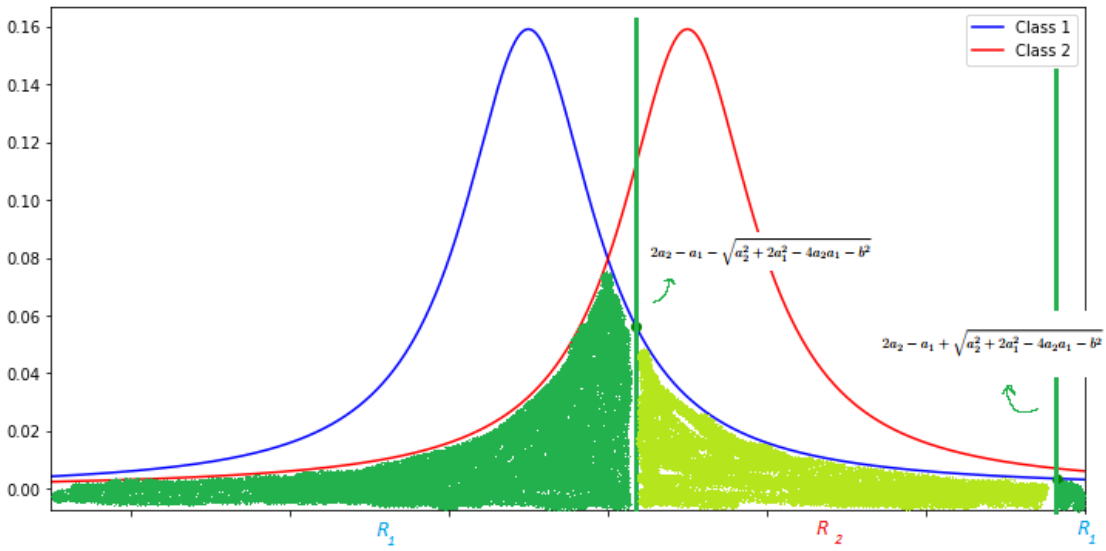

Figure 5: Selected area in $P(x \mid \omega_i)P(\omega_i)$ plot, indicates our total error in second case

The Bayesian classifier to minimize total risk seeks to find boundaries that make our total risk minimum. In order to do that it makes one area more favorable to the other (in this problem R_1) and contains a reducible error area. Now our classifier is optimum based on total risk and its total error might be higher than the usual Bayesian classifier to minimize total error.

2 Problem 2: Decision Boundary of Rayleigh Distribution

$$P(x \mid \omega_i) = \begin{cases} \frac{x}{\sigma_i^2} \exp\left(-\frac{x^2}{2\sigma_i^2}\right) & x \geq 0 \\ 0 & x < 0 \end{cases}$$

To find the decision boundary of this problem first, we see their intersection:

$$P(\omega_1 \mid x) = P(\omega_2 \mid x)$$

Using Bayes formula:

$$\frac{p(x \mid \omega_1)p(\omega_1)}{p(x)} = \frac{p(x \mid \omega_2)p(\omega_2)}{p(x)}$$

Since prior distributions are the same we can simplify the problem as below.

$$\rightarrow \frac{x}{\sigma_1^2} \exp\left(-\frac{x^2}{2\sigma_1^2}\right) = \frac{x}{\sigma_2^2} \exp\left(-\frac{x^2}{2\sigma_2^2}\right)$$

$$\rightarrow log(\frac{\sigma_2^2}{\sigma_1^2}) = \frac{x^2}{2\sigma_1^2} - \frac{x^2}{2\sigma_2^2}$$

We know $x \geq 0$

$$x = \sqrt{\frac{4log(\frac{\sigma_2}{\sigma_1})}{\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}}}$$

And finally, we derived our Bayesian classification as below:

$$\omega_i = \begin{cases} \omega_1 & x > \sqrt{\frac{4log(\frac{\sigma_2}{\sigma_1})}{\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}}} \\ \omega_2 & o.w. \end{cases}$$

3 Problem 3: Binary Classification

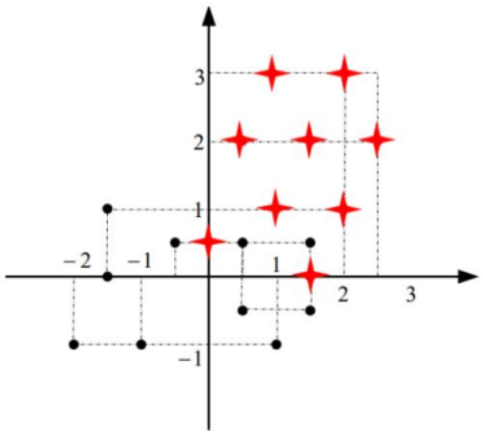

Figure 6: Class distribution of the problem

(a) In the following parts, we assume both classes have the Gaussian distribution and we try to perform Bayesian classification on them.

(b)

$$\hat{\mu} = \frac{1}{n}\sum x_i \quad , \quad \hat{\Sigma} = \frac{1}{n}\sum (x_i - \hat{\mu})(x_i - \hat{\mu})^T$$

- Red class:

$$\hat{\mu_r} = \frac{1}{9}\left(\begin{bmatrix}1\\3\end{bmatrix} + \begin{bmatrix}2\\3\end{bmatrix} + \begin{bmatrix}0.5\\2\end{bmatrix} + \begin{bmatrix}1.5\\2\end{bmatrix} + \begin{bmatrix}2.5\\2\end{bmatrix} + \begin{bmatrix}1\\1\end{bmatrix} + \begin{bmatrix}2\\1\end{bmatrix} + \begin{bmatrix}0\\0.5\end{bmatrix} + \begin{bmatrix}1.5\\0\end{bmatrix} \right) = \begin{bmatrix}1.3333\\1.6111\end{bmatrix}$$

$$\hat{\Sigma_r} = \begin{bmatrix}0.5556 & 0.1852\\0.1852 & 0.9877\end{bmatrix}$$

- Black class:

$$\hat{\mu_b} = \frac{1}{10}\left(\begin{bmatrix}-1.5\\1\end{bmatrix} + \begin{bmatrix}-0.5\\0.5\end{bmatrix} + \begin{bmatrix}0.5\\0.5\end{bmatrix} + \begin{bmatrix}1.5\\0.5\end{bmatrix} + \begin{bmatrix}0.5\\-0.5\end{bmatrix} + \begin{bmatrix}-1.5\\0\end{bmatrix} + \begin{bmatrix}1.5\\-0.5\end{bmatrix} + \begin{bmatrix}-2\\-1\end{bmatrix} + \begin{bmatrix}-1\\-1\end{bmatrix} + \begin{bmatrix}1\\-1\end{bmatrix} \right) = \begin{bmatrix}-0.15\\-0.15\end{bmatrix}$$

$$\hat{\Sigma_b} = \begin{bmatrix}1.5525 & 0.0025\\0.0025 & 0.5025\end{bmatrix}$$

(c) Given the identical prior distribution between the red and black classes, the Gaussian discriminant functions are defined below:

$$g_i(x) = p(x \mid \omega_i) = x^T \mathbf{W}_i x + \mathbf{w}_i^T x + w_{i0}$$

Where

$$\mathbf{W}_i = -\frac{1}{2}\Sigma_i^{-1}$$

$$\mathbf{w}_i = \Sigma_i^{-1}\mu_i$$

$$w_{i0} = -\frac{1}{2}\mu_i^t \Sigma_i^{-1}\mu_i - \frac{1}{2}\ln|\Sigma_i| + \ln P(\omega_i)\,.$$

Now we can easily evaluate the decision boundary:

- Red class:

$$\mathbf{W}_2 = \begin{bmatrix} -0.96 & 0.18 \\ 0.18 & -0.54 \end{bmatrix} \quad \mathbf{w}_2 = \begin{bmatrix} 1.98 \\ 1.26 \end{bmatrix} \quad w_{20} = -2.6958$$

$$g_2(x) = (-0.96x_1^2 + 0.36x_1x_2 - 0.54x_2^2) + (1.98x_1 + 1.26x_2) - 2.6958$$

- Black class:

$$\mathbf{W}_1 = \begin{bmatrix} -0.3221 & 0.0016 \\ 0.0016 & -0.9950 \end{bmatrix} \quad \mathbf{w}_1 = \begin{bmatrix} -0.0961 \\ -0.2980 \end{bmatrix} \quad w_{10} = -0.5986$$

$$g_1(x) = (-0.3221x_1^2 - 0.0032x_1x_2 - 0.995x_2^2) - (0.0961x_1 + 0.2980x_2) - 0.5986$$

So the decision boundary is :

$$g(x) = g_2(x) - g_1(x) = 0$$

$$\rightarrow g(x) = -0.6379x_1^2 + 0.3632x_1x_2 + 0.455x_2^2 + 2.0761x_1 + 1.558x_2 - 2.0972 = 0$$

$$\omega_i = \begin{cases} \omega_1 & \mathbf{x} < g(\mathbf{x}) \text{ (} \mathbf{x} \text{ be under the curve)} \\ \omega_2 & o.w. \end{cases}$$

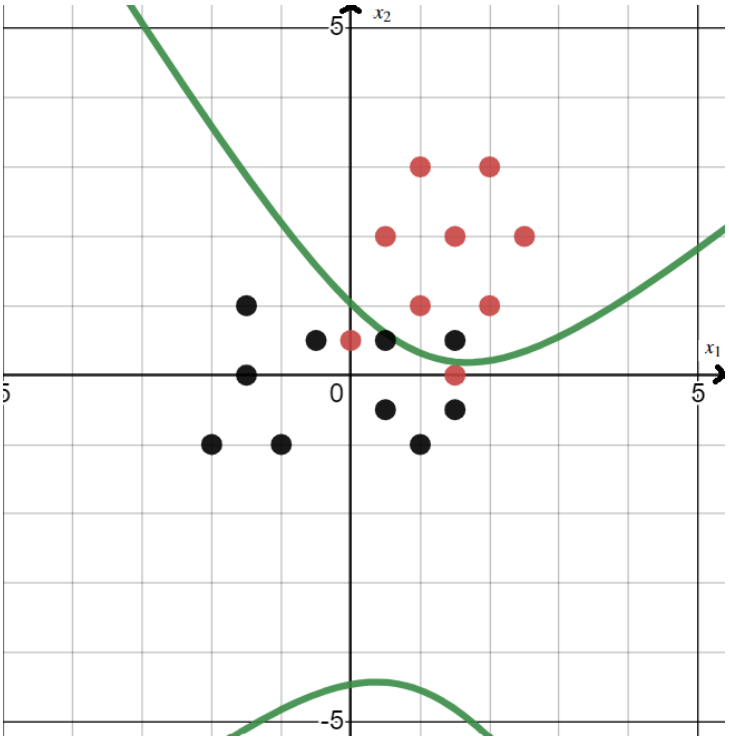

Figure 7: Decision Boundary for Minimization of Error

The experimental error is equal to $\frac{3}{19} \approx 0.158$ or %15.8

(d) If we add "Risk" to the previous part we should change the discriminant function for minimizing the risk as follow:

$$g_i(x) = \sum_{j=1}^{c} \lambda_{ij} P(x \mid \omega_j) P(\omega_j)$$

Actually, we weighed our errors according to the given λ_{ij} and remember, again prior distributions are the same:

- Red class:
$$g_2(x) = a \times P(x \mid \omega_1)$$
- Black class:
$$g_1(x) = 2a \times P(x \mid \omega_2)$$

And the decision boundary is :

$$g(x) = log(g_1(x)) - log(g_2(x)) = 0$$

We just need to add $log(2)$ to the $g(x)$ in the previous part:

$$g(x) = -0.6379x_1^2 + 0.3632x_1x_2 + 0.455x_2^2 + 2.0761x_1 + 1.558x_2 - 1.4041 = 0$$

$$\omega_i = \begin{cases} \omega_1 & \mathbf{x} < g(\mathbf{x}) \text{ (}\mathbf{x}\text{ be under the curve)} \\ \omega_2 & o.w. \end{cases}$$

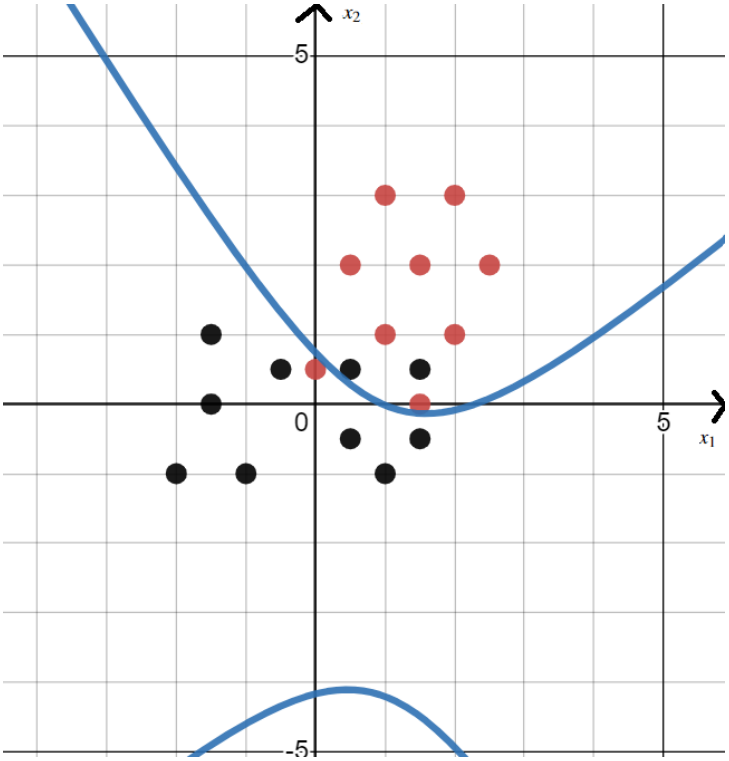

Figure 8: Decision Boundary for Minimization of Risk

After analyzing Figure 8, a shift towards minimizing total risk instead of total error increased the area to select the red class over the black class. This is attributed to the $\lambda_{12} > \lambda_{21}$ and results in an elevated risk of selecting black over red, making red the more favorable option for decision-making.

(e) Now we have a different prior distribution and we note that our discriminant function is defined as $P(\omega_i)P(x \mid \omega_i)$ so we can say:

$$log(P(\omega_2)P(x \mid \omega_2)) - log(P(\omega_1)P(x \mid \omega_1)) = 0$$

Again, we just need to add $log(2)$ to the $g(x)$ in the part (c):

$$g(x) = -0.6379x_1^2 + 0.3632x_1x_2 + 0.455x_2^2 + 2.0761x_1 + 1.558x_2 - 1.4041 = 0$$

$$\omega_i = \begin{cases} \omega_1 & \mathbf{x} < g(\mathbf{x}) \text{ (}\mathbf{x}\text{ be under the curve)} \\ \omega_2 & o.w. \end{cases}$$

As evident from the previous section, the same results can be utilized in this section too. It's worth noting that, since we arrived at the same conclusion in the last two sections, we can conclude that in certain cases, incorporating prior knowledge in the form of risk can be beneficial.

The experimental error is equal to $\frac{3}{19} \approx 0.158$ or $\%15.8$

4 Problem 4: Parameter Estimation

$$P(X) = \frac{\lambda^x e^{-\lambda}}{x!} \quad , \quad D = \{X_1, X_2, ..., X_n\}$$

(a) In this case, we use Maximum Likelihood (ML) approach and assume all X_i are i.i.d:

$$\hat{\lambda}_{ML} = \arg\max_{\lambda} P(D;\lambda)$$

$$P(D;\lambda) = P(X_1, X_2, ..., X_n;\lambda) = \prod_{i=1}^{n} P(X_i;\lambda)$$

Now we take logarithm and simplify the equation:

$$LL(\lambda) = log(\lambda) \sum_{i=1}^{n} x_i - n\lambda - \sum_{i=1}^{n} log(x_i!)$$

$$\frac{\partial LL(\lambda)}{d\lambda} = 0 \rightarrow \frac{1}{\lambda} \sum_{i=1}^{n} x_i - n = 0$$

$$\rightarrow \hat{\lambda}_{ML} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

(b) According to the problem we know $P(\lambda) = \text{Gamma}(\lambda \mid \alpha, \beta) = c\lambda^{\alpha-1} e^{-\beta\lambda}$. Now we have to find $P(\lambda \mid D)$

$$P(\lambda \mid D) \sim P(D \mid \lambda) P(\lambda) = P(X_1, X_2, ..., X_n \mid \lambda) P(\lambda) \overset{i.i.d}{=} \prod_{i=1}^{n} P(X_i \mid \lambda) P(\lambda)$$

$$= \Big(\prod_{i=1}^{n} \frac{\lambda^{x_i} e^{-\lambda}}{x_i!} \Big) c\lambda^{\alpha-1} e^{-\beta\lambda} = c' \lambda^{\sum_{i=1}^{n} x_i + \alpha - 1} e^{-\lambda(\beta+n)}$$

$$\rightarrow \quad P(\lambda \mid D) = \text{Gamma}(\lambda \mid \sum_{i=1}^{n} x_i + \alpha, \beta + n)$$

(c) Yes. The following result shows that both prior and posterior distributions are Gamma distributions with different parameters. So $P(\lambda \mid D)$ is a reproducing density and $P(\lambda)$ is a **conjugate prior**.

(d) Maximum a posteriori estimation (MAP) aims for the point where the posterior distribution has the maximum value. Maximum point for Gamma distribution is $\lambda = \frac{\alpha-1}{\beta}$.

$$\hat{\lambda}_{MAP} = \arg\max_{\lambda} P(\lambda \mid D) P(\lambda)$$

$$\rightarrow \quad \hat{\lambda}_{MAP} = \frac{\sum_{i=1}^{n} x_i + \alpha - 1}{n + \beta}$$

(e) We can rewrite the previous equation as below:

$$\hat{\lambda}_{MAP} = \frac{\sum_{i=1}^{n} x_i + \alpha - 1}{\beta + n} = \left[\frac{n}{n+\beta}\right]\left(\frac{\sum_{i=1}^{n} x_i}{n}\right) + \frac{\alpha - 1}{n+\beta}$$

We know from part one that $\hat{\lambda}_{ML} = \frac{1}{n}\sum_{i=1}^{n} x_i$. So we can say:

$$\hat{\lambda}_{MAP} = \left[\frac{n}{n+\beta}\right]\hat{\lambda}_{ML} + \frac{\alpha - 1}{n+\beta}$$

And when $n \to \infty$:

$$\lim_{n\to\infty} \hat{\lambda}_{MAP} = \lim_{n\to\infty} \left[\frac{n}{n+\beta}\right]\hat{\lambda}_{ML} + \frac{\alpha - 1}{n+\beta} = \hat{\lambda}_{ML}$$

$$\lim_{n\to\infty} \hat{\lambda}_{MAP} = \hat{\lambda}_{ML}$$

We can see when the number of data goes to infinity, the MAP estimation tends asymptotically to the ML estimation and the prior becomes irrelevant given a large number of observations.

(f) For data sets of limited length, MAP estimation is more accurate and we can add our previous knowledge in terms of the prior distribution. If we have large data sets or our previous knowledge for prior distribution is limited it's better to use ML estimation.

5 Problem 5: Naïve Bayes

(a) Assume we have a problem that has n features. Bayes' theorem states the following relationship, given class variable y and dependent feature vector x_1 through x_n:

$$P(y \mid x_1, ..., x_n) = \frac{P(y)P(x_1, ..., x_n \mid y)}{P(x_1, ..., x_n)}$$

Naive Bayes is a machine learning algorithm that is based on the Bayes theorem. It is called 'naive' because it makes a naive assumption that all the features are independent of each other, which is often not entirely true in real-world scenarios. Using the naive conditional independence assumption we have:

$$P(x_i \mid y, x_1, ..., x_{i-1}, x_{i+1}, ..., x_n) = P(x_i \mid y)$$

And we can rewrite the first equation:

$$P(y \mid x_1, ..., x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, ..., x_n)} \sim P(y)\prod_{i=1}^{n} P(x_i \mid y)$$

In the Naive Bayes assumption, we deliberately add bias to our model in order to have more generalization. Also, Naive Bayes calculations are very simple compared to dependent feature assumptions.

There are three types of Naive Bayes algorithms: Gaussian Naive Bayes, Multinomial Naive Bayes, and Bernoulli Naive Bayes. Gaussian Naive Bayes is used for continuous data, while Multinomial Naive Bayes is generally used for discrete data like natural language processing. Bernoulli Naive Bayes is best suited for binary classification problems.

(b) In this problem, we use Gaussian Naive Bayes for the classification of the penguin's classes. We assume all features have Gaussian distribution and we compare multiple of each feature conditional distribution and choose the maximum:

$$\hat{y} = \arg\max_{y} P(y)\prod_{i=1}^{n} P(x_i \mid y)$$

You can see detailed codes in *"HW1_Problem5_codes.ipynb"* attached with this file.

Results:

```
Accuracy:  0.9703264094955489
```

Figure 9: Accuracy without Sklearn

```
Confusion Matrix for Adelie
                  Predicted Negative  Predicted Positive
Actual Negative                  182                   5
Actual Positive                    5                 145
Precision:  0.9666666666666667
Recall:  0.9666666666666667
```

```
Confusion Matrix for Chinstrap
                  Predicted Negative  Predicted Positive
Actual Negative                  265                   5
Actual Positive                    5                  62
Precision:  0.9253731343283582
Recall:  0.9253731343283582
```

```
Confusion Matrix for Gentoo
                  Predicted Negative  Predicted Positive
Actual Negative                  217                   0
Actual Positive                    0                 120
Precision:  1.0
Recall:  1.0
```

Figure 10: Confusion Matrix without Sklearn

(c) Now we use the Sklearn library and compare the results:

```
[[145   5   0]
 [  5  62   0]
 [  0   0 120]]
Classification Report:
              precision    recall  f1-score   support

      Adelie       0.97      0.97      0.97       150
   Chinstrap       0.93      0.93      0.93        67
      Gentoo       1.00      1.00      1.00       120

    accuracy                           0.97       337
   macro avg       0.96      0.96      0.96       337
weighted avg       0.97      0.97      0.97       337
```

```
Confusion Matrix for Adelie
                  Predicted Negative  Predicted Positive
Actual Negative                  145                   5
Actual Positive                    5                 145
Confusion Matrix for Chinstrap
                  Predicted Negative  Predicted Positive
Actual Negative                   62                   5
Actual Positive                    5                  62
Confusion Matrix for Gentoo
                  Predicted Negative  Predicted Positive
Actual Negative                  120                   0
Actual Positive                    0                 120
```

Figure 11: Results and Confusion Matrix with Sklearn

As we can see the results are similar and our model works well.

6 Problem 6: Image Classification

As we know the Blue color is dominant in sea pictures. So a straightforward solution is to open pictures with open-cv and get the corresponding BGR vectors for each one.

Then calculate the mean for the BGR value of vectors and get only one BGR vector for each figure. Now for the classification part, we can compare the values of B(blue) and G(green) for each figure and if the related value for B was greater than G we say it belongs to the sea figures.

You can see detailed codes in *"HW1_Problem6_codes.ipynb"* attached with this file.

This method gives us the following result:

```
Confusion Matrix =  [[39, 1], [2, 40]]
accuracy =  0.9634146341463414
precision =  0.9512195121951219
recall =  0.975
```

Figure 12: Accuracy with comparing RGB colors

Now we can take a look at pictures that are misclassified.

Figure 13: Misclassified Pictures

In this classification algorithm, we consider only the blue and green colors of each picture. So it's natural autumn jungles with less green color are not categorized as jungles and the picture with more green color is classified as a jungle.