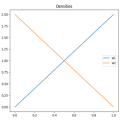
Amin Fadaee

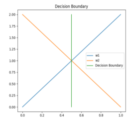
Pattern Recognition HW2

Python 3.5, Libraries: Numpy, Matplotlib, Documentation: HTML, Bootstrap 4

Problem 1

a. The densities can be seen in the figure (Densities).





b. $g(x) : p(x \mid w_1)p(w_1) - p(x \mid w_2)p(w_2) > 0$

$g(x) : 2x \times \frac{1}{2} - (2-2x) \times \frac{1}{2} > 0$

$g(x) : x - 1 + x = 2x - 1 > 0$

$$g(x) : x > \frac{1}{2}$$

The decision boundary is shown in the figure (Decision Boundary)

c. The bayes error can be calculated in the following manner:

$$\int_{\mathcal{R}_2} p(x \mid w_1)p(w_1)\,dx + \int_{\mathcal{R}_1} p(x \mid w_2)p(w_2)\,dx =$$

$$\int_0^{\frac{1}{2}} x\,dx + \int_{\frac{1}{2}}^{1} 1 - x\,dx = \frac{1}{4}$$

d. $g(x) : p(x \mid w_1)p(w_1) - p(x \mid w_2)p(w_2) > 0$

$g(x) : 2x \times \frac{3}{10} - (2-2x) \times \frac{7}{10} > 0$

$g(x) : x > \frac{7}{10}$

Problem 2

a. Because the covariance matrices are the same the discriminant function can be obtained using the following formula:

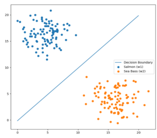
$$g_i(x) = -\frac{1}{2}(x-\mu_i)^t\Sigma_i^{-1}(x-\mu_i) + \ln P(w_i)$$

$g_1(x) = -\frac{1}{2}(x-\mu_1)^t\Sigma_1^{-1}(x-\mu_1) + \ln P(w_1)$

$g_1(x) = \begin{bmatrix} 2-\frac{x_1}{2} & 8-\frac{x_2}{2} \end{bmatrix} \begin{bmatrix} \frac{1}{4} & 0 \\ 0 & \frac{1}{4} \end{bmatrix} \begin{bmatrix} x_1 - 4 & x_2 - 16 \end{bmatrix}$

$g_1(x) = -\frac{x_1^2}{8} + x_1 - \frac{x_2^2}{8} + 4x_2 - 34 + \ln(0.6)$

$g_2(x) = -\frac{x_2^2}{8} + x_2 - \frac{x_1^2}{8} + 4x_1 - 34 + \ln(0.4)$



b. $g(x) : g_1(x) = g_2(x)$

$g(x) : -\frac{x_1^2}{8} + x_1 - \frac{x_2^2}{8} + 4x_2 - 34 + \ln(0.6) = -\frac{x_2^2}{8} + x_2 - \frac{x_1^2}{8} + 4x_1 - 34 + \ln(0.4)$

$$g(x) : x_2 = x_1 - 0.1351$$

The decision function is shown in the figure.

Problem 3

a. The prior probability is the same, because as far as the samples tell us there are equal instances of each class, hence $P(w_1) = P(w_2) = \frac{1}{2}$.

b.

$\mu_1 = \begin{bmatrix} 1.83 \\ 1.5 \end{bmatrix}$, $\mu_2 = \begin{bmatrix} 8.16 \\ 9.33 \end{bmatrix}$, $\Sigma_1 = \begin{bmatrix} 2.16 & 1.1 \\ 1.1 & 1.1 \end{bmatrix}$, $\Sigma_2 = \begin{bmatrix} 1.36 & -0.066 \\ -0.066 & 1.066 \end{bmatrix}$

c.

The posterior probability can be derived using the following formula:

$g_i(x) = x^tW_ix + w_i^tx + w_{i0}$

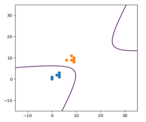
$$W_i = -\frac{1}{2}\Sigma_i^{-1}$$
$$w_i = \Sigma_i^{-1}\mu_i$$
$$w_{i0} = -\frac{1}{2}\mu_i^t\Sigma_i^{-1}\mu_i - \frac{1}{2}\ln\left|\Sigma\right| + \ln P(w_i)$$

Using python (refer to `Problem 3.py`) we can compute the parameters for $g_1(x)$ and $g_2(x)$:

$$g_1(x) = x^t\begin{bmatrix} -0.46875 & 0.46875 \\ 0.46875 & -0.9232 \end{bmatrix}x + \begin{bmatrix} 0.3125 & 1.0511 \end{bmatrix}x - 1.8478$$
$$g_2(x) = x^t\begin{bmatrix} -0.3669 & -0.0229 \\ -0.0229 & -0.4701 \end{bmatrix}x + \begin{bmatrix} 6.4220 & 9.1513 \end{bmatrix}x - 69.8097$$

Now we can derive the boundary using $g_1(x) = g_2(x)$:

$$x(59.9853 + y \times (-9.65439)) + x^2 + 4.4487y^2 + 79.5307y - 667.274 = 0$$

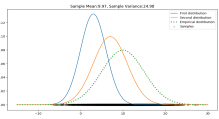


d. Different penalties for misclassification would definitely result in change in the decision boundary. For example if the error for classifying 1 as 2 is 100% and 2 as 1 is 0% the decision boundary would always favor class 1 and classify every instance as 1.

Problem 4

a. It is a proven fact that If $X_1, X_2, \ldots, X_n$ are mutually independent normal random variables with means $\mu_1, \mu_2, \ldots, \mu_n$ and variances $\sigma_1^2, \sigma_2^2, \cdots, \sigma_n^2$, then their sum follows the norma distribution $N\left(\sum_{i=1}^{n}\mu_i, \sum_{i=1}^{n}\sigma_i^2\right)$. We can empirically show that this is true, we consider two normal distributions:
1. $N(3, 3)$
2. $N(7, 4)$

If we sum multiple samples from each distribution, then based on the mentioned fact the distribution of this sample should follow the normal $N(10, 25)$. Using python (`Problem 4.py`) we obtained the empirical result.



b. The Mean is 9.97

c. The Variance is 24.98

The results is corresponding with our assertion and it can be seen in the figure.

Problem 5

a. Similar to what we did in **Problem 3.c.** we are going to find W_i, w_i and w_{i0} (refer to `Problem 4.py`) and we would finally derive:

$$g_1(x) = x^t\begin{bmatrix} -0.5 & 0 \\ 0 & -0.5 \end{bmatrix}x + \begin{bmatrix} 0 & 0 \end{bmatrix}x - 0.69314$$
$$g_2(x) = x^t\begin{bmatrix} -0.5 & 0 \\ 0 & -0.5 \end{bmatrix}x + \begin{bmatrix} 1 & 1 \end{bmatrix}x - 1.6931$$

Now we can derive the boundary using $g_1(x) = g_2(x)$:

$$x_2 \approx 1 - x_1$$

b. We can obtain the Bhattacharyya error bound using the following formula:

$$P(error) \leq \sqrt{P(w_1)P(w_2)}e^{-k\left(\frac{1}{2}\right)}$$
$$k\left(\frac{1}{2}\right) = \frac{1}{8}(\mu_2-\mu_1)^t\left[\frac{\Sigma_1+\Sigma_2}{2}\right]^{-1}(\mu_2-\mu_1) + \frac{1}{2}\ln\frac{\left|\frac{\Sigma_1+\Sigma_2}{2}\right|}{\sqrt{|\Sigma_1|\,|\Sigma_2|}}$$

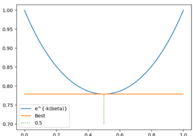
and implementing this formula results in **0.3894** for this problem.

c.

The Chernoff bound is computed in the following manner:

$$P(error) \leq P^{\beta}(w_1)P^{1-\beta}(w_2)\,e^{-k(\beta)}$$
$$k(\beta) = \frac{\beta(1-\beta)}{2}(\mu_2-\mu_1)^t\left[\beta\Sigma_1 + (1-\beta)\Sigma_2\right]^{-1}(\mu_2-\mu_1) + \frac{1}{2}\ln\frac{|\beta\Sigma_1 + (1-\beta)\Sigma_2|}{|\Sigma_1|^{\beta}\,|\Sigma_2|^{1-\beta}}$$

and β is a number in [0,1] that minimizes $e^{-k(\beta)}$. Using python we can plot this curve for every beta β and find the best one. This result can be shown in the figure and as it can be seen the best β is **0.5** which makes Chernoff bound equal to that of Bhattacharyya (**0.3894**).



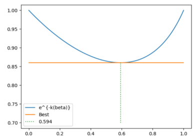
d.

- $g_1(x) = x^t\begin{bmatrix} -0.2666 & 0 \\ 0.0666 & -0.2666 \end{bmatrix}x + \begin{bmatrix} 0 & 0 \end{bmatrix}x - 1.3540$

  $g_2(x) = x^t\begin{bmatrix} -0.1190 & 0.0476 \\ 0.0476 & -0.1190 \end{bmatrix}x + \begin{bmatrix} 0.1428 & 0.1428 \end{bmatrix}x - 2.3582$

  $x^2 - 0.257453xy + 0.96748x + y^2 + 0.96748y = 6.80352$

  (form of an ellipse).
- Bhattacharyya Bound= 0.4323
- $\beta = 0.594$
- Chernoff Bound= 0.4300



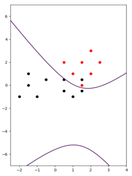
Problem 6

We are going to use the exact same steps we did for **Problem 3**:

1. By counting the instances we can derive that $P(w_1) = 0.41$ and $P(w_2) = 0.59$.
2. $\mu_1 = \begin{bmatrix} 1.5714 \\ 1.5714 \end{bmatrix}$,
3. $\mu_2 = \begin{bmatrix} -0.15 \\ -0.15 \end{bmatrix}$,
4. $\Sigma_1 = \begin{bmatrix} 0.4523 & 0.1190 \\ 0.1190 & 0.9523 \end{bmatrix}$,
5. $\Sigma_2 = \begin{bmatrix} 1.725 & 0.0027 \\ 0.0027 & 0.5583 \end{bmatrix}$
6. $g_1(x) = x^t\begin{bmatrix} -1.1428 & 0.1428 \\ 0.1428 & -0.5428 \end{bmatrix}x + \begin{bmatrix} 3.1428 & 1.2571 \end{bmatrix}x - 3.9067$

   $g_2(x) = x^t\begin{bmatrix} -0.2898 & 0.0014 \\ 0.0014 & -0.8955 \end{bmatrix}x + \begin{bmatrix} -0.0865 & -0.2682 \end{bmatrix}x - 0.5384$

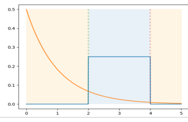
$$x^2 + x(-0.331536y - 3.78581) - 0.413482y^2 - 1.78816y = -3.94877$$



Problem 7

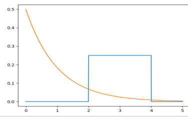
a.

Bayes classifiers classifies instances based on their posterior probability and based on this criteria and the help of the figure we can safely assume that the classifier classifies all xs in range 2 to 4 as class 1 and all the others as class 2.



b.

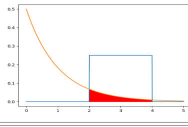
The result can be seen in the figure. As the uniform distribution must have area of 1 the height of $U(2, 4)$ would be 0.5 that becomes 0.25 after multiplying it by the prior probability.



c.

As can be seen in the figure, the error is the bayes error is the red shaded area which can be calculated in the following way:

$$\int_2^4 \frac{e^{-x}}{2}dx = \frac{e^2-1}{2e^4} \approx 0.058510$$



Computer Project

a. Function `generate_sample_points` in `Computer Project` is a a procedure to generate sample points in d dimensions with a normal distribution having mean μ and covariance matrix Σ.

b. By doing the same thing we did in previous problem we can obtain:

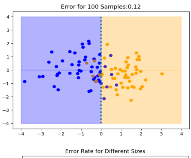
$$g_1(x) = x^t\begin{bmatrix} -0.50 & \\ 0 & -0.5 \end{bmatrix}x + [-1, 0]x - 1.19314$$
$$g_2(x) = x^t\begin{bmatrix} -0.50 & \\ 0 & -0.5 \end{bmatrix}x + [1, 0]x - 1.19314$$

and therefore:

$$x_1 = 0$$

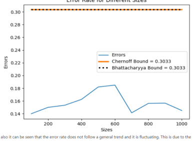
c.

For 100 Samples we obtained 0.12 error (of course as the samples are drawn randomly this number may change in repeating the process). The figure shows the decision boundary and the samples.



d.

The plot for repeating the above process for different sample sizes is shown in the figure.



e. As can be seen in the above figure both the Chernoff and Bhattacharyya errors are above the empirical error, also it can be seen that the error rate does not follow a general trend and it is fluctuating. This is due to the fact that the error is averaged over the number of samples.