CS 479/679 Pattern Recognition
Spring 2023 – Prof. Bebis
Programming Assignment 1 – Due on 3/1/2023 at 11:59pm

Consider a two-class classification problem where the data of each class is modeled by a 2D Gaussian density $N(\mu_1,\Sigma_1)$ and $N(\mu_2,\Sigma_2)$.

Data Generation: using the parameters shown below, generate 60,000 random samples from $N(\mu_1,\Sigma_1)$ and 140,000 samples from $N(\mu_2,\Sigma_2)$ (i.e., 200,000 samples total). We will be referring to this data set as "data set **A"**.

$$\mu_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \mu_2 = \begin{bmatrix} 4 \\ 4 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Notation:

$$\mu = \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix} \quad \Sigma = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}$$

Note: you need to use the **Box-Muller** transformation to generate the samples from each distribution; please review the document "Generating Gaussian Random Numbers", which is posted on the course's webpage, for more information. A link to the C code is also provided on the webpage. Since the code generates samples from a 1D Gaussian distribution, you should call the **Box-Muller** function twice to generate each 2D sample (x, y); use (μ_x, σ_x) to generate the x value and (μ_y, σ_y) to generate the y value.

Note: *ranf()* is not defined in the standard library, you could use the simple implementation:

```
/* ranf - return a random double in the [0,m] range.*/

double ranf(double m) {
   return (m*rand()/(double)RAND_MAX);
}
```

1. In this experiment, use data set **A.**

a. Design a Bayes classifier for minimum error to classify the samples from set **A**. Which discriminant (i.e., case I, II, or III) would be optimum in this case and why? How would you set the prior probabilities $P(\omega_1)$ and $P(\omega_2)$?
b. Plot both the Bayes decision boundary and the samples from data set **A** on the **same plot** to better visualize how the Bayes rule would classify the data in this case.
c. Next, classify all 200,000 samples and report (i) the misclassification rate for each class **separately** (i.e., the percentage of misclassified samples for each class) and (ii) the **total** misclassification rate (i.e., the percentage of misclassified samples overall).
d. Calculate the theoretical probability error (e.g., Bhattacharyya bound) and compare it with the misclassification rate from part (c). What do you observe?

Data Generation: using the parameters shown below, generate 60,000 random samples from $N(\mu_1,\Sigma_1)$ and 140,000 samples from $N(\mu_2,\Sigma_2)$ (i.e., 200,000 samples total). We will be referring to this data set as "data set **B"**.

$$\mu_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \mu_2 = \begin{bmatrix} 4 \\ 4 \end{bmatrix} \Sigma_2 = \begin{bmatrix} 4 & 0 \\ 0 & 8 \end{bmatrix}$$

2. Repeat experiment 1 using data set **B**. How do your results from this experiment compare with your results from experiment 1 and why?

3. Quite often, the **Euclidean distance classifier** (shown below but also discussed in the lecture) is used for classification without understanding that it is an optimum classifier **only** when certain assumptions hold true as we discussed in the lecture. Classify the samples from data set **A** using the Euclidean distance classifier and compare your results (i.e., misclassification rates) with those obtained from experiment 1. Explain your findings.

$$g_i(\mathbf{x}) = -\| \mathbf{x} - \boldsymbol{\mu}_i \|^2$$

4. Repeat experiment 3 using the samples from data set **B**. Compare and discuss your results with those obtained from experiments 2 and 3. Explain your findings.