**Department of Electrical and Computer Engineering**

Homework Assignment No. 01:

**HW No. 01: Gaussian Distributions**

submitted to:

Professor Joseph Picone

ECE 8527: Introduction to Pattern Recognition and Machine Learning

Temple University

College of Engineering

1947 North 12th Street

Philadelphia, Pennsylvania 19122

January 22, 2022

prepared by:

Gavin Koma  
Email: gavintkoma@temple.edu

# Description OF task 1

The goal of Task 1 is to populate data values given a covariance matrix and a mean for each of four classes. A covariance matrix is a square matrix that describes the covariance—the joint variability of two variables—given a vector. Thus, the primary coding goal of this assignment is to generate data that is described by the given mean and the covariance matrix.

In order to do so, I chose to utilize python and two common imported libraries: pandas and numpy. I then defined the four classes and the four corresponding mean values for each class. This can be seen in the following snippet of code:

Text

Description automatically generated

The means were condensed into a list to facilitate work with later for-loops. In order to create our data distribution, I used numpy’s random.multivariate\_normal function which allows one to draw random samples from a multivariate normal distribution. This function allows us to generate a dataset while providing a given mean and a covariance matrix (both of which we are given in the homework assignment). I elected to create a dictionary which will allow me to for-loop through both the varying classes and their means instead of repeating the same lines of code. This was performed as follows:

Text

Description automatically generated

This nested for-loop creates an output of four separate data frames; each of which contains an x-vector and a y-vector. At this point, I was unsure of how to use further loops or even list comprehension to facilitate the creation of our final concatenated data frame. I was able to complete this by using a naïve method to assign column names and create a final data frame.

Text

Description automatically generated

Table

Description automatically generatedThis final data frame was then saved as a .csv file and was then imported into JMP Pro 16 for further analysis.

JMP Pro 16 provides an extremely convenient method for performing the Naïve-Bayes Predictive Modeling method. With JMP Pro 16open, one can select their data and in the Analyze menu, can choose the Naïve-Bayes method. However, before we can do that, we must first organize our data into in three total columns. One column will assign labels of class where Class 1 is A, Class 2 is B, etc…, the second column will contain all x-vector components, and the third column will contain all y-vector components. Once properly organized, the data in JMP should look similar to this image on the right.

When selecting data for the Naïve-Bayes Predictive Modeling, I chose to place the y-vector as the response and both the class and the x-vector as the Factors. Doing so provides the following graph:

Chart, histogram

Description automatically generated

I guess now explain wtf this thing is and maybe double check me formatting?

# Description OF Task 2

The goal of Task 2 is the same as the goal of Task 1: to generate a dataset of four classes with a given mean and a given covariance matrix. The difference, however, is that the covariance matrix has been altered from [[0.1,0], [0,0.1]] to [[1.0,0], [0,1.0]], thus inducing more variance in our generated dataset. The same code that was utilized in part 1 was recycled for use in part 2. A screenshot of this code was included on the following page for ease of understanding. Note that the covariance matrix values have been changed as indicated but the rest of the code has been unchanged. The data was also reorganized in JMP Pro 16 to fit the needs for a Naïve-Bayes analysis. This screenshot has been excluded and if needed, an idea of the format can be referenced from Task 1.

Graphical user interface, text

Description automatically generated

# Summary

The Naïve-Bayes classifier is a simple classifying method that applies Bayes’ theorem with a strong assumption of independence between features. The essence of Bayes’ theorem is that it allows us to find the probability of some event A happening given that another event B has occurred. The independence assumption that was previously mentioned is again that our events are independent of each other.

This model assigns class labels to ?????