24-787 Artificial Intelligence and Machine Learning for Engineering Design
Homework 2

1. *(25 points)* Write a MATLAB program to do the following:

   (a) Generate 20 training points from each of the three Gaussian distributions described by the parameters listed below:

$$\mu_1 = (0,0,0)^T \qquad \Sigma_1 = \text{diag}[3,5,2]$$

$$\mu_2 = (1,5,-3)^T \qquad \Sigma_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 1 \\ 0 & 1 & 6 \end{bmatrix}$$

$$\mu_3 = (0,0,0)^T \qquad \Sigma_3 = 10\mathbf{I}$$

   In your report, describe how you generated these points and plot the resulting data on a single 3D plot with a unique color for each class. You are **not** allowed to use the built-in MATLAB function `mvnrnd`.

   (b) Estimate the means and covariance matrices for your data. Show the results. You are **not** allowed to use the built-in MATLAB functions `mean` and `cov`.

   (c) Use the estimated parameters from part (b) to classify the following test vectors, assuming each class is equally likely:

$$x_1 = (0,0,0)$$
$$x_2 = (1,6,-3)$$
$$x_3 = (0.8,3.0,-1.5)$$
$$x_4 = (8,-8,8)$$
$$x_5 = (-1,1,-1)$$
$$x_6 = (0,5,0)$$

   Show all steps in your calculation. Do the results make sense? Why or why not?

   (d) Repeat part (c), but now assume that the third class is three times as likely as the first or second class. Show all steps in your calculation. Do the results make sense? Why or why not?

Probability Density Functions

2. *(15 points)*

   In class, we defined the entropy for a discrete random variable X. Let us now consider the case where X is a continuous random variable with probability density function $p(x)$. The entropy is then defined as:

$$H(X) = -\int p(x) \ln p(x) \mathrm{d}x$$

   Assume that X follows a Gaussian distribution with mean μ and variance σ^2, such that:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

   Compute the entropy $H(X)$ as a function of the distribution parameters.

3. *(15 points)*

   Expectation is similar to "mean" or "average". The difference is that an average usually refers to the average of the data we have collected/observed from a phenomenon, whereas expectation usually refers to the underlying distribution from which the data is sampled. For a discrete random variable X, the expectation is defined as

$$E[X] = \sum_{-\infty}^{\infty} iP(X = i)$$

   Similarly, the expectation for a continuous random variable Y is given by

$$E[Y] = \int_{-\infty}^{\infty} yp(y) \mathrm{d}y$$

   (a) Show that for discrete random variables W and Z, $E[W + Z] = E[W] + E[Z]$.

   (b) Show that for continuous random variables W and Z, $E[W + Z] = E[W] + E[Z]$.

4. *(10 points)*

   Variance of a random variable X indicates how spread out the distribution is. Precisely, if $\bar{X} = E[X]$, the variance is defined as $Var[X] = E[(X - \bar{X})^2]$.

   (a) Show that for any random variable X, $Var[X] = E[X^2] - (E[X])^2$.

   (b) Show that $Var[X] = \sigma^2$ if we let X be a continuous random variable that follows a normal distribution such that

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

   where σ can be any positive real number, and μ can be any real number.

5. *(10 points)*

Rayleigh distribution is given by:

$$f(x;\sigma) = \frac{x}{\sigma^2} e^{-x^2/(2\sigma^2)}, \quad x \geq 0,$$

Show that the MLE estimation of $\boldsymbol{\sigma^2}$ is:

$$\widehat{\sigma^2} \approx \frac{1}{2N} \sum_{i=1}^{N} x_i^2$$

6. *(25 points)*

You will write a program to recognize single-stroke gestures. See the folder named 'Data.' There are three gestures: Alpha, Beta, Delta. Each folder contains training data for each of these gestures.

(a) For each gesture type, **visualize the first three training data** using Matlab's Plot function (you may have to invert the y-axis). Each gesture is a single stroke symbol; consecutive sample x-y points within each file can be joined with a line segment in Matlab's plot function.

(b) After studying all the data files by visualizing them, **devise a 3D feature vector that can be used to represent each gesture**. For instance an example feature vector could be:

- Width of the bounding box
- Height of the bounding box
- Distance between the first and last point of each stroke

While this might work, it may not be a good feature vector: The bounding box may be similar for many of the gestures, and the width and height may not be very informative. Moreover, this representation will be 'scale' dependent. That is, if a gesture were drawn twice as large, the corresponding two feature vectors would be different (which is not good).

You want to design features that are **scale** and **position** independent. They do not need to be rotation independent. That is, the letter 'p' and 'd' should have different feature vectors. (Rotation independent feature vectors are more difficult to devise, so your job here is easier).

Describe your features in detail. Tell us how you compute your features. For the first training file for each gesture, print the feature vector (hence a total of 3 vectors).

(c) Using your feature vectors, compute the mean and covariance from the samples you have. Note that each gesture has a different number of training samples. **Print the mean and covariance vectors.**

(d) Matlab code:

You must submit a **single Matlab file called hwk2gestures.m** that:

- When we run it (we will open it in Matlab editor and click 'Run,' we shall not need to do anything else), it should automatically open and process the files in 'Data' and learn the mean and covariances for each of the data files. Folder **'Data'** will be in the same folder containing **hwk2gestures.m. '**Data' will have the three gesture subfolders.

- Your code should then open a 'Test' folder (to be provided, or you can prepare one yourself) that contains 50 test samples. The test samples will start at test1.txt and will end at test50.txt and their format will be similar to the training data. You may visualize them similar to the training data, but the goal is to treat them as novel, unseen data. Folder **'Test** will be in the same folder containing **hwk2gestures.m**

- Your code should output a single text file **testresult.txt that in the same folder that hwk2gestures.m is contained.** This output text file should have 50 rows. Row *n* should correspond to the recognition result of your code for test file *n*, with the following mapping:

  Alpha → 1

  Beta → 2

  Delta →3

  For instance, if your code determines that the first five test files represent (in order): Alpha, Beta, Beta, Delta, Alpha, the first five lines of your testresult.txt file should look like this:

  1
  2
  2
  3
  1

  We will run your code on further test samples that you will not have, so make sure your code works well.