# EECE5644 Fall 2025 – Assignment 1

**Submit:** Monday, 2025-Oct-13 before 23:59ET

Please submit your solutions on in a single PDF file on Canvas; include all math, numerical and visual results, and link to your code in an online repository. Only the contents of this PDF will be graded.

This is a graded assignment and the entirety of your submission must contain only your own work. You may benefit from publicly available literature including software (not from classmates), as long as these sources are properly acknowledged in your submission (i.e., include citations to sources of all materials you benefit from). While exchanging any written material between classmates and with former students of this course is not allowed, discussing with the instructor, the teaching assistant, Generative AI models, and classmates to get clarification or to eliminate doubts is acceptable. However, you must then synthesize this help into your own solution(s) including your own code implementation.

Indicate all external assistance you received in a *citations* section. Disclosed assistance from aforementioned sources in citation are acceptable, but not disclosing assistance received is considered as academic dishonesty. Note that the point of this exercise is for you to develop the necessary understanding and skills to be able to develop appropriate code/solution on your own even if you did not have access to these resources. By submitting a PDF file in response to this take home assignment you are declaring that the contents of your submission, and the associated code is your own work, and that all assistance you received from external sources is specified in your citations section.

# Question 1 (40%)

The probability density function (pdf) for a 3-dimensional real-valued random vector $\mathbf{X}$ is as follows: $p(\mathbf{x}) = p(\mathbf{x}|L=0)P(L=0) + p(\mathbf{x}|L=1)P(L=1)$. Here $L$ is the true class label that indicates which class-label-conditioned pdf generates the data.

The class priors are $P(L=0) = 0.65$ and $P(L=1) = 0.35$. The class class-conditional pdfs are $p(\mathbf{x}|L=0) = g(\mathbf{x}|\mathbf{m}_0, \mathbf{C}_0)$ and $p(\mathbf{x}|L=1) = g(\mathbf{x}|\mathbf{m}_1, \mathbf{C}_1)$, where $g(\mathbf{x}|\mathbf{m}, \mathbf{C})$ is a multivariate Gaussian probability density function with mean vector $\mathbf{m}$ and covariance matrix $\mathbf{C}$. The parameters of the class-conditional Gaussian pdfs are:

$$\mathbf{m}_0 = \begin{bmatrix} -1/2 \\ -1/2 \\ -1/2 \end{bmatrix} \quad \mathbf{C}_0 = \begin{bmatrix} 1 & -0.5 & 0.3 \\ -0.5 & 1 & -0.5 \\ 0.3 & -0.5 & 1 \end{bmatrix} \quad \mathbf{m}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \mathbf{C}_1 = \begin{bmatrix} 1 & 0.3 & -0.2 \\ 0.3 & 1 & 0.3 \\ -0.2 & 0.3 & 1 \end{bmatrix}$$

For numerical results requested below, generate 10000 samples according to this data distribution, keep track of the true class labels for each sample. Save the data and use the same data set in all cases. Here is a Matlab script that will generate data from this pdf:

```
clear all, close all,
N = 10000; p0 = 0.65; p1 = 0.35;
u = rand(1,N)>=p0; N0 = length(find(u==0)); N1 = length(find(u==1));
mu = [-1/2;-1/2;-1/2]; Sigma = [1,-0.5,0.3;-0.5,1,-0.5;0.3,-0.5,1];
r0 = mvnrnd(mu, Sigma, N0);
figure(1), plot3(r0(:,1),r0(:,2),r0(:,3),'.b'); axis equal, hold on,
mu = [1;1;1]; Sigma = [1,0.3,-0.2;0.3,1,0.3;-0.2,0.3,1];
r1 = mvnrnd(mu, Sigma, N1);
figure(1), plot3(r1(:,1),r1(:,2),r1(:,3),'.r'); axis equal, hold on,]]
```

**Part A (20 points):** ERM classification using the knowledge of true data pdf:

1. Specify the minimum expected risk classification rule in the form of a likelihood-ratio test: $\frac{p(\mathbf{x}|L=1)}{p(\mathbf{x}|L=0)} \overset{?}{>} \gamma$, where the threshold $\gamma$ is a function of class priors and fixed (nonnegative) loss values for each of the four cases $D = i|L = j$ where $D$ is the decision label that is either 0 or 1, like $L$.

2. Implement this classifier and apply it on the 10K samples you generated. Vary the threshold $\gamma$ gradually from 0 to $\infty$, and for each value of the threshold compute the true positive (detection) probability $P(D = 1|L = 1; \gamma)$ and the false positive (false alarm) probability $P(D = 1|L = 0; \gamma)$. Using these paired values, trace/plot an approximation of the ROC curve of the minimum expected risk classifier. Note that at $\gamma = 0$ The ROC curve should be at $\binom{1}{1}$, and as *gamma* increases it should traverse towards $\binom{0}{0}$. Due to the finite number of samples used to estimate probabilities, your ROC curve approximation should reach this destination value for a finite threshold value. Keep track of $(D = 0|L = 1; \gamma)$ and $P(D = 1|L = 0; \gamma)$ values for each *gamma* value for use in the next section.

3. Determine the threshold value that achieves minimum probability of error, and on the ROC curce, superimpose clearly (using a different color/shape marker) the true positive and false positive values attained by this minimum-P(error) classifier. Calculate and report an estimate

of the minimum probability of error that is achievable for this data distribution. Note that $P(error; \gamma) = P(D = 1|L = 0; \gamma)P(L = 0) + P(D = 0|L = 1; \gamma)P(L = 1)$. How does your empirically selected $\gamma$ value that minimizes P(error) compare with the theoretically optimal threshold you compute from priors and loss values?

**Part B (10 points):** ERM classification attempt using incorrect knowledge of data distribution (Naive Bayesian Classifier:

Assume that the features are independent given each class label. Specifically, assume that you know the true class prior probabilities, but for some reason you think that the class conditional pdfs are both Gaussian with the true means, but (incorrectly) with covariance matrices both equal to the identity matrix (assuming that off-diagonal values are all zeros). Analyze the impact of this model mismatch in this Naive Bayesian (NB) approach to classifier design by repeating the same steps in Part A on the same 10K sample data set you generated earlier. Report the same results, answer the same questions. Did this model mismatch negatively impact your ROC curve and minimum achievable probability of error?

**Part C (10 points):** Fisher LDA Classifier:

Repeat the process using a Fisher Linear Discriminant Analysis (LDA) based classifier. Using the 10K available samples, estimate the class conditional pdf mean and covariance matrices using sample average estimators for mean and covariance. From these estimated mean vectors and covariance matrices, determine the Fisher LDA projection weight vector (via the generalized eigendecomposition of within and between class scatter matrices): $\mathbf{w}_{LDA}$. For the classification rule $\mathbf{w}_{LDA}^T\mathbf{x}$ compared to a threshold $\tau$, which takes values from $-\infty$ to $\infty$, trace the ROC curve. Identify the threshold at which the probability of error (based on sample count estimates) is minimized, and clearly mark that operating point on the ROC curve estimate. Discuss how this LDA classifier performs relative to the previous two classifiers.

*Note: When finding the Fisher LDA projection matrix, do not be concerned about the difference in the class priors. When determining the between-class and within-class scatter matrices, use equal weights for the class means and covariances, like we did in class.*

# Question 2 (40%)

A 2-dimensional random vector **X** takes values from a mixture of four Gaussians. Each Gaussian pdf is the class-conditional pdf for one of four class labels $L \in \{1,2,3,4\}$. For this problem, pick your own 4 distinct Gaussian class conditional pdfs $p(\mathbf{x}|L = j)$, $j \in \{1,2,3,4\}$ with arbitrary mean vectors and arbitrary covariance matrices. Set all class priors to 0.25. *Note that equal class prior does NOT mean equal number of samples come from each class; the label for each sample must be randomly selected in accordance with the class prior distribution.*

**Part A (20 points):** Minimum probability of error classification (0-1 loss, MAP classification rule).

1. Generate 10000 samples from this data distribution and keep track of the true labels of each sample.
2. Specify the decision rule that achieves minimum probability of error (i.e., use 0-1 loss), implement this classifier with the true data distribution knowledge, classify the 10K samples and count the samples corresponding to each decision-label pair to empirically estimate the confusion matrix whose entries are $P(D = i|L = j)$ for $i, j \in \{1,2,3,4\}$.
3. Provide a visualization of the data (scatter-plot in 2-dimensional space), and for each sample indicate the true class label with a different marker shape (dot, circle, triangle, square) and whether it was correctly (green) or incorrectly (red) classifier with a dofferent marker color as indicated in parantheses.

**Part B (20 points):** Let's designate the Gaussian that overlaps most with other class labels (e.g. the Gaussian in the middle of the triangle scenario suggested above) as $L = 4$. Repeat the exercise for the ERM classification rule with the following loss values: $\lambda_{ii} = 0$, $i \in \{1,2,3,4\}$, $\lambda_{ij} = 1$, $i \in \{1,2,3,4\}, j \in \{1,2,3\}$, $i \neq j$, and $\lambda_{i4} = 3$, $i \in \{1,2,3,4\}$:

$$\Lambda = \begin{bmatrix} 0 & 10 & 10 & 100 \\ 1 & 0 & 10 & 100 \\ 1 & 1 & 0 & 100 \\ 1 & 1 & 1 & 0 \end{bmatrix} \tag{1}$$

Note that with this loss matrix choice, different error types are penalized differently. For instance, misclassifying a sample from label 4 is penalized with 100 units of loss, implying the design will attempt to highly prefer classifying in favor of this class.

Using sample average over the 10K samples, estimate the minimum expected risk that this optimal ERM classification rule will achieve. *Hint: When counting errors, each sample for which $D = i|L = l$ where $i \neq l$ will contribute the corresponding loss matrix entry to the average risk calculation.*

# Question 3 (20%)

Download the following datasets...

- Wine Quality dataset located at `https://archive.ics.uci.edu/ml/datasets/Wine+Quality` consists of 11 features, and class labels from 0 to 10 indicating wine quality scores. There are 4898 samples.

- Human Activity Recognition dataset located at `https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones` consists of 561 features, and 6 activity labels. There are 10299 samples.

Implement minimum-probability-of-error classifiers for these problems, assuming that the class conditional pdf of features for each class you encounter in these examples is a Gaussian. Using all available samples from a class, with sample averages, estimate mean vectors and covariance matrices. Using sample counts, also estimate class priors. In case your sample estimates of covariance matrices are ill-conditioned, consider adding a regularization term to your covariance estimate as in: $\mathbf{C}_{Regularized} = \mathbf{C}_{SampleAverage} + \lambda \mathbf{I}$ where $\lambda > 0$ is a small regularization parameter that ensures the regularized covariance matrix $\mathbf{C}_{Regularized}$ has all eigenvalues larger than this parameter.

With these estimated (trained) Gaussian class conditional pdfs and class priors, apply the minimum-P(error) classification rule on all (training) samples, count the errors, and report the error probability estimate you obtain for each problem. Also report the confusion matrices for both datasets, for this classification rule.

Visualize the datasets in various 2 or 3 dimensional projections (either subsets of features, or using the first few principal components). Discuss if Gaussian class conditional models are appropriate for these datasets and how your model choice might have influenced the confusion matrix and probability of error values you obtained in the experiments conducted above. Make sure you explain in rigorous detail what your modeling assumptions are, how you estimated/selected necessary parameters for your model and classification rule, and describe your analyses in mathematical terms supplemented by numerical and visual results in a way that conveys your understanding of what you have accomplished and demonstrated.

*Hint: Later in the course, we will talk about how to select regularization/hyper-parameters. For now, you may consider using a value on the order of arithmetic average of sample covariance matrix estimate non-zero eigenvalues $\lambda = \alpha trace(\mathbf{C}_{SampleAverage})/rank(\mathbf{C}_{SampleAverage})$ or geometric average of sample covariance matrix estimate non-zero eigenvalues, where $0 < \alpha < 1$ is a small real number. This makes your regularization term proportional to the eigenvalues observed in the sample covariance estimate.*