

Spring 2017 EECE 5644 Homework #1

Prof: Jaume Coll-Font TA: Aziz Kocanaogullari, Zhiqiang Tao

Make sure you read the problem statements and answer the questions. Submit the code online. This homework could be a bit challenging, so I suggest you start early.

1.1 (14 pts) Suppose two equally probable one-dimensional densities are of the form $p(x|\omega_i) \propto e^{-|x-a_i|/b_i}$ for $i = 1, 2$ and $b \geq 0$.

- (a) Write an analytic expression for each density, that is, normalize each function for arbitrary a_i , and positive b_i .
- (b) Calculate the likelihood ratio $p(x|\omega_1)/p(x|\omega_2)$ as a function of your four variables.
- (c) Plot a graph (using MATLAB or Python) of the likelihood ratio for the case $a_1 = 1, b_1 = 1, a_2 = 2$ and $b_2 = 2$. Make sure the plots are correctly labeled (axis, titles, legend, etc.) and that the fonts are legible when printed.

1.2 (18 pts) Consider a two-class, one-dimensional problem where $P(\omega_1) = P(\omega_2)$ and $p(x|\omega_i) \sim N(\mu_i, \sigma_i^2)$. Let $\mu_1 = 0, \sigma_1^2 = 1, \mu_2 = \mu, \sigma_2^2 = \sigma^2$.

- (a) Derive a general expression for the location of the Bayes optimal decision boundary as a function of μ and σ .
- (b) With $\mu = 2$ and $\sigma^2 = 2$, make two plots using MATLAB: one for the class conditional pdfs $p(x|\omega_i)$ and one for the posterior probabilities $p(\omega_i|x)$ with the location of the optimal decision regions. Make sure the plots are correctly labeled (axis, titles, legend, etc.) and that the fonts are legible when printed.
- (c) Estimate the Bayes error rate p_e .
- (d) Comment on the case where $\mu = 0$ and σ^2 is much greater than 1. Describe a practical example of a pattern classification problem where such a situation might arise.

1.3 (34 pts) Write a function that generates independent and identically distributed samples given class-specific parameters. The class conditional densities are normal distributions accepting class dependent means μ_i and covariances Σ_i . The function should also accept arbitrary class priors $P(\omega_i)$. Be sure to test your function with some simple cases to validate that it is working correctly. Here are the specifications for the function:

```
function [data, classIndex] = generateGaussianSamples(mu, sigma, nSamples, prior);  
%  
% Function to simulate data from k Gaussian densities (1 for each class) in d dimensions.  
%  
% INPUTS:  
% mu          - k-by-1 cell with the class dependent d-dimensional mean vector  
% sigma       - k-by-1 cell with the class dependent d-by-d covariance matrix  
% nSamples    - scalar indicating number of samples to be generated  
% prior       - k-by-1 vector with class dependent mean  
%  
% OUTPUTS:  
% data        - nSamples-by-d array with the simulated data distributed along the rows  
% classIndex  - vector of length nSamples with the class index for each datapoint
```

One way to write this function would be to first sample the class and then sample the data point from the respective class conditional density. Generate samples according to the parameters given below and produce two-dimensional scatter plots with the simulated datasets (six plots in total):

- (a) $n_{\text{Samples}} = 400$, $\mu_1 = [0,0]$, $\sigma_1 = I$, (with I being the identity matrix), and $\mu_2 = [-3,-3]$, $\sigma_2 = I$. Use equal priors.
- (b) Same as (a), but now $\sigma_1 = \sigma_2 = \begin{bmatrix} 3 & 1 \\ 1 & 0.8 \end{bmatrix}$. Use equal priors.
- (c) $n_{\text{Samples}} = 400$, $\mu_1 = [0,0]$, $\sigma_1 = \begin{bmatrix} 2 & 0.5 \\ 0.5 & 1 \end{bmatrix}$, $\mu_2 = [2,2]$, $\sigma_2 = \begin{bmatrix} 2 & -1.9 \\ -1.9 & 5 \end{bmatrix}$. Use equal priors.
- (d) Same as (a), but make the $P(\omega_1) = 0.05$
- (e) Same as (b), but make the $P(\omega_1) = 0.05$
- (f) Same as (c), but make the $P(\omega_1) = 0.05$

The plots must have a legend to indicate what class the samples belong to. Make sure the plots are correctly labeled (axis, titles, legend, etc.) and that the fonts are legible when printed.

1.4 (34 pts) Write a function for each of the three normal distribution discriminant functions discussed in class ([case I] class dependent mean with identity covariance, [case II] class dependent mean with shared covariance, and [case III] class dependent mean and covariances). The inputs to the functions should be similar to those of problem 3. The output of the discriminant functions should be a `numberOfSamples-by-k` array with the score for each class along the columns and the scores for each data point along the rows. Classify the simulated samples from question 1.3 (the 6 cases (a)-(f)) using the discriminant functions you coded. Make a scatter plot with the classification results. Compute the accuracy (number of correctly classified points divided by the total number of points) and display it on the title of the plot. Make sure the plot has a legend to indicate the classification results. In the same plot, show the decision boundary. Use the following instructions for each of the respective simulated datasets:

- (a) For data from 1.3(a): Since only the means change, use only the class dependent mean discriminant (case I). You should have 1 plot.
- (b) For data from 1.3(b): Since the covariances are the same, use the class dependent mean discriminant (case I) and the shared covariance discriminant (case II). You should have 2 plots. Do you see a difference in performance between the two classifiers? Why?
- (c) For data from 1.3(c): For this dataset, use the three discriminant functions (case I, II, and III). Notice that each discriminant function will make some assumptions about the data. For the shared covariance case (case II), just average the individual class covariances (can you explain if this is a reasonable thing to do?). You should have 3 plots. Do you see a difference in performance between the three classifiers? Which one works best? Why?

- (d) For data from 1.3(d): Repeat the same analysis as for the dataset used in (a). Additionally, compute the accuracy of classifying all data points as the class with the highest prior. How does it perform? Is this a good classifier? Why or why not? What can you say about accuracy as classification metric?
- (e) For data from 1.3(e): Repeat the same analysis as for the dataset used in (b). Additionally, compute the accuracy of classifying all data points as the class with the highest prior. How does it perform? Is this a good classifier? Why or why not? What can you say about accuracy as classification metric?
- (f) For data from 1.3(f): Repeat the same analysis as for the dataset used in (c). Additionally, compute the accuracy of classifying all data points as the class with the highest prior. How does it perform? Is this a good classifier? Why or why not? What can you say about accuracy as classification metric?

In total, you should have 12 plots.