"Machine Learning and Computational Statistics"

6th Homework

Exercise 1:

Consider a two-class 1-dim. classification problem of two equiprobable classes ω_1 and ω_2 that are modeled by the normal distributions $N(0,1)$ and $N(0,10)$, respectively. Determine the decision regions R_1 and R_2 corresponding to the two classes.

Exercise 2:

(α) Consider a three-class 1-dim. problem where the classes ω_1, ω_2 και ω_3 are modeled by the following uniform distributions

$$p(x|\omega_1) = \begin{cases} 1/4, & x \in (0,2) \cup (5,7) \\ 0, & \text{otherwise} \end{cases} \qquad p(x|\omega_2) = \begin{cases} 1/10, & x \in (0,10) \\ 0, & \text{otherwise} \end{cases}$$

$$\text{και} \quad p(x|\omega_3) = \begin{cases} 1, & x \in (3,4) \\ 0, & \text{otherwise} \end{cases}$$

(I) Assume that all classes are equiprobable.

(i) Depict graphically in the same figure $P(\omega_i)p(x|\omega_i)$ (as functions of x) and identify the respective decision regions, as they are specified by the Bayes classifier.
(ii) Compute the error classification probability of the Bayes classifier.
(iii) Classify the point $x' = 3.5$ to one of the three classes using the Bayes classifier.

(II) Assume that the classes are **not** equiprobable.

(i) Determine a set of values for the a priori probabilities of the three classes that guarantee that $x' = 3.5$ is assigned to class ω_2. Justify briefly your choice.

(ii) Is there any combination of the a priori probabilities that guarantees that $x' = 3.5$ will be assigned to ω_1? Explain.

Hints:

(*H1*) Focus only in the interval [0,10] since all pdfs are zero out of this interval.

(*H2*) The error classification probability for the Bayes classifier is

$$P_e = \sum_{i=1}^{M} \int_{R_i} \left(\sum_{k=1, k \neq i}^{M} p(x/\omega_k) P(\omega_k) \right) dx$$

Exercise 3 (python code + text):

Consider a two-class, two-dimensional classification problem for which you can find attached two sets: one for training and one for testing (file *HW6.mat*). Each of these sets consists of pairs of the form $(y_i, \mathbf{x}_i)$, where y_i is the class label for vector $\mathbf{x}_i$. Let N_{train} and N_{test} denote the number of training and test sets, respectively. The data are given via the following arrays/matrices:

- ***train_x*** (a N_{train} x2 **matrix** that contains in its rows the training vectors $\mathbf{x}_i$)
- ***train_y*** (a N_{train}–dim. column **vector** containing the **class labels** (1 or 2) of the corresponding training vectors $\mathbf{x}_i$ included in ***train_x***).
- ***test_x*** (a N_{test}x2 **matrix** that contains in its rows the test vectors $\mathbf{x}_i$)
- ***test_y*** (a N_{test}–dim. column **vector** containing the **class labels** (1 or 2) of the corresponding test vectors $\mathbf{x}_i$ included in ***test_x***).

Assume that the two classes, ω_1 and ω_2 are modeled by normal distributions.

Adopt the **Bayes classifier**.

i. Use the training set to **estimate** $P(\omega_1)$, $P(\omega_2)$, $p(\mathbf{x}|\omega_1)$, $p(\mathbf{x}|\omega_2)$ (Since $p(\mathbf{x}|\omega_j)$ is modeled a normal distribution, it is completely identified by $\boldsymbol{\mu}_j$ and Σ_j. Use the **ML estimates** for them as given in the lecture slides).
ii. **Classify** the points $\mathbf{x}_i$ of the test set, using the **Bayes classifier** (for each point apply the Bayes classification rule and keep the class labels, to an a N_{test}–dim. column **vector** , called ***Btest_y*** containing the **estimated class labels** (1 or 2) of the corresponding test vectors $\mathbf{x}_i$ included in ***test_x.***).
iii. Estimate the error classification probability ((1) **compare** ***test_y*** and ***Btest_y*** , (2) **count** the positions where both of them have the same class label and (3) **divide** with the total number of test vectors).

Recall that $\mathbf{x} = [x_1, x_2]^T$

(b) Depict graphically the training set, using different colors for points from different classes.
(c) Report the classification results and comment **very briefly** on them.

Hint: After downloading the attached MATLAB file, use the following python code to retrieve the above mentioned matrices and vectors:

```
import scipy.io as sio

Dataset = sio.loadmat('HW6.mat')

train_x = Dataset['train_x']
train_y = Dataset['train_y']
test_x = Dataset['test_x']
test_y = Dataset['test_y']
```