DS 501: STATISTICAL AND MATHEMATICAL METHODS FOR DATA SCIENCE

Assignment 03

DUE: Wednesday, September 26, 2018.

PROBLEM

This problem is to be implemented in Matlab or Octave. **You have to submit code + hard copy of the report, whose template is given.**

Background Reading: Tom Mitchel's Machine Learning book chapter 1 at: www.cs.cmu.edu/~tom/mlbook.html and Christopher Bishop chapter 2

Dataset

Same as for assignment 02. The dataset is taken from causality workbech:
http://www.causality.inf.ethz.ch/challenge.php?page=datasetshttp://www.causality.inf.ethz.ch/challenge.php?page=datasets

1. Read the dataset training.txt First 11 columns are features and the last column is the label.

2. From the training set determine the parameters of the Gaussian distribution (likelihood function)

3. Determine the predictions on the test data given in testing.txt file using Bayes' theorem assuming first a diagonal covariance matrix and then repeat with the complete covariance matrix.

TO SELF STUDY / WORKOUT MATH

You will notice that when computing P(**x**|C), you may get an underflow error as probabilities are very small values and taking a product of all these values will result in an extremely small value. So instead of computing P(**x**|C) compute log P(**x**|C). The class for which log P(C|**x**) is maximum is the chosen class.

Rewrite the formula for log(p(C|**x**)) in your report.

HINTS FOR CODE IN MATLAB/OCTAVE

To understand implementation, lets take this example:
TRAINING DATA (labels known)

X_1	X_2	X_3	Label
0	1	1	1
0	1	0	1
0	0	1	1
0	1	0	1
1	0	1	1
1	0	0	1
1	1	1	1
0	1	0	0

0	0	1	0
1	1	0	0

TEST DATA (labels not known)

X_1	X_2	X_3
0	1	0
0	1	0
0	1	1

NOTE: Be systematic when implementing your program. You can implement the following functions along with a main script in Matlab/Octave for the problem

Accesing data, rows and columns

```
dat = load('...','-ascii');  %replace … with filename
%above is a built in function for reading text files.  The entrie dataset will be stored in dat

f1 = dat(:,1);  %this stores first column in f1
r1 = dat(1,:);  %this stores first row in r1
% last column in training file has the labels
trainLabels = dat(:,end);              %all labels are in trainLabels
trainX = dat(:,1:end-1);               %all the features are now in trainX
```

Generating model

Write two functions in Matlab/Octave

function [meanVecClass0 meanVecClass1 covMatrixClass0 covMatrixClass1 prior0 prior1] = learnGaussDiagonalCov(trainX,trainLabels)

function [meanVecClass0 meanVecClass1 covMatrixClass0 covMatrixClass1 prior0 prior1] = learnGauss(trainX,trainLabels)

For the example above for diagonal covariance matrix:

meanVecClass1 = $[0.4286 \quad 0.5714 \quad 0.5714]^T$

covMatrixClass1 = $\begin{pmatrix} 0.2857 & 0 & 0 \\ 0 & 0.2857 & 0 \\ 0 & 0 & 0.2857 \end{pmatrix}$

prior1 = 7/10

Compute the same for class0

For the example above for complete covariance matrix:

meanVecClass1 = $[0.4286 \quad 0.5714 \quad 0.5714]^T$

covMatrixClass1 = $\begin{pmatrix} 0.2857 & -0.1190 & 0.0476 \\ -0.1190 & 0.2857 & -0.0476 \\ 0.0476 & -0.0476 & 0.2857 \end{pmatrix}$

prior1 = 7/10
Compute the same for class0

Testing the model / Predicting the labels
Write two functions
function [predictedLabels MAPClass0 MAPClass1] = testMAP(meanVecClass0 meanVecClass1 covMatrixClass0 covMatrixClass1 prior0 prior1)
Call the above function twice. Once for diagonal matrix and once for complete covariance matrix

Example: Apply to test point (0,0,0) with complete covariance matrix
The test function should implement the computation of probability using Gaussian. For example, using complete covariance matrix:
MAPClass1 for test point **x**(0,0,0) = p(**x**|C=1)P(C=1)/P(**x**)
square of Mahalonobis distance between mean and (0,0,0) = 4.2857
MAPClass1 for test point **x**(0,0,0) = 0.0547*7/10/P(**x**)
Note compute P(**x**) by applying sum rule of probability:
P(**x**|C=0)P(C=0)+P(**x**|C=1)P(C=1)

Main script
Once you have implemented the above functions write a main script that:

a. Reads training data
b. Finds the model parameters (relevant to the distribution to use)
c. Reads the test data and classifies the test data

TO SUBMIT

1. Make a folder with your roll number as folder name. Put Matlab's source code in it and a soft copy of your report and upload it on slate. NO EMAIL SUBMISSIONS ACCEPTED.
2. **Hard copy** of the report. The template for the report is given and you have to fill out the necessary table. First type in the table, print the report and fill in the handwritten part.