[25 points]

Iris flower dataset (iris_dataset.dat)

- Introduced by Ronald Fisher in 1936 (sometimes called Anderson's Iris dataset)
- The first four columns list the measurements in centimeters of the variables sepal length, sepal width, petal length and petal width.
- The last column shows the class label of each example (1=setosa, 2=versicolor, 3=virginica)

Iris Versicolor

Iris Virginica

Iris Setosa

Fisher Iris (sepal length, sepal width, petal length, petal width, class)

5.3000	3.7000	1.5000	0.2000	setosa
5.0000	3.3000	1.4000	0.2000	setosa
7.0000	3.2000	4.7000	1.4000	versicolor
6.9000	3.1000	5.4000	2.1000	virginica

In this coding assignment you are to implement a Minimum Risk Bayes Decision Theoretic classifier and use it to classify the test examples in the provided datasets.

Assume the following:

- All conditional density functions are multivariate Gaussian
- Each class has its own covariance matrix
- Equally likely prior probabilities
- 0-1 loss function
- 10-fold cross validation for training & testing

The suggested MATLAB commands are included. You don't have to use these commands. Type "doc <command>" at the MATLAB prompt to see more information on the commands.

1. Read "iris_dataset.dat" and randomly shuffle the data
2. Using the training data, estimate the parameters by MLE method
    - MATLAB commands: mean, inv
    - Do not use MATLAB cov function
3. Classify test data
4. Show the classification accuracy per iteration.
5. Show the average classification accuracy after the 10-fold CV is completed.
6. Repeat 1-5 with "corrupted_iris_dataset.dat". This data set is corrupted by Gaussian noise. You should see some reduction in the classification accuracy

Hints:

```
clear all; close all; clc;

%data = dlmread('iris_dataset.dat');
data = dlmread('corrupted_iris_dataset.dat');

N = 150;  % total number of samples
NC = 50;  % size of each class
K = 10;  % K-fold

% Randomly shuffle data
%seed = 150; rand('seed', seed);
index = randperm(N);
data_shuffled = data(index,:);


% 10-fold cross validation
for k=1:K

  % TRAINING:
  % Separate training data and test data

  % Using training data from each class, find mean_mle (u1, u2, u3)
  % and cov_mle (cov1, cov2, cov3)
  % Do not use MATLAB cov function, use covmle provided with
  % this assignment


  % TESTING:
  % Using u1, u2, u3, and cov1, cov2, cov3 found in the training phase,
  % and test data (x), compute the discriminant function for each class,
  % g1, g2, g3.  Assume prior=1/3 for each class.
  % You can use MATLAB inv function for matrix inversion

  % Predicted class label is the largest of g1, g2, g3
  % Check predicted label against the given label in the test data set

end

% Evaluate classification accuracy
%   Accuracy per iteration = no of correct classification / 15
%   Average accuracy for all 10-fold CV
%   fprintf('Accuracy = %5.4f\n', ...) generates nice format
```

Expected output:

“iris_dataset.dat”

```
Classification accuracy
ans =
    0.9333
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    0.9333
    0.9333
    1.0000
    0.9333
Average accuracy = 0.9733
```

“corrupted_iris_dataset.dat”

```
Classification accuracy
ans =
    0.8667
    0.8667
    0.9333
    0.7333
    0.8000
    0.8667
    0.8667
    0.7333
    0.7333
    0.8000
Average accuracy = 0.8200
```