

Assignment 4

Harshit Vijayvargia

UFID- 1935-5645

Question:1:

Solution: For implementing a Bayes classifier we need to decide first how many sample we will be considering in training and testing data. As suggested I took first 70% data as training as rest as testing.

Classification for each input will be made based on value returned by discriminant function for each class.

To calculate the discriminant value first, I calculated different statistics of training data which are governing discriminant function. These parameters are mean, covariance, inverse of matrices etc. **Considering all 7 features Initially** I check my algorithm on training, testing and entire dataset one by one.

The confusion Matrix which I computed for different sets of data along with their computation time:

Actual: 0, represents Species 0

Predicted-0 represents that decision made is 0

Predicted-1 represents that decision made is 1

Actual: 1, represents Species 1

1) For training data (140 samples)

n = 140	Predicted- 0	Predicted - 1
Actual : 0	72	0
Actual : 1	0	68

Computation time: 0.078125000000000 seconds

2) For testing data (60 samples)

n = 60	Predicted- 0	Predicted - 1
Actual : 0	28	0
Actual : 1	0	32

Computation time: 0.046875000000000 seconds

3) For entire dataset (200 samples)

n = 200	Predicted- 0	Predicted - 1
Actual : 0	100	0
Actual : 1	0	100

Computation time: 0.062500000000000 seconds

During this process, what I observe was that covariance matrix was becoming singular. Although the computations were working but since the determinant of matrix has become almost zero, it is affecting computation time. Also, if there were more samples then this algorithm would have failed since inverse of covariance matrix will become infinite in such case. To avoid this singularity problem I checked the features which are given and found that two features (Male and female) are complement of each other which is creating this problem. To avoid this I created one feature gender where I will represent male as 1 and female as 0. By doing this I am not losing any essential information but trying to eliminate redundant feature. This process not only solved the problem of covariance matrix getting singular but also decreased the computation time since we have decreased one feature. This computation time change can be observed from table below:

Computation time (in seconds) comparison:

	Testing	Training	Whole Data Set
With 7 features	0.046875	0.078125	0.0625
With 6 features	0.03325	0.046875	0.056875

Question 2:

Solution: The second problem asks for LDA which is a classifier that attempts to approximate Bayes classifier. LDA assumes each class has normal distribution with a class specific mean vector and common variance. Here we won't be using prior probabilities as we used in Bayes. Generally, LDA is used when we have more than two classes where we require dimensionality reduction while at the same time preserving class discrimination information.

Since I already observed that computation time is less with 6 features and accuracy is good. So for this problem too I will consider 6 features.

But here the discriminant function will be different. Instead of comparing discriminant values we will be comparing our discriminant value with a threshold value. The choice of threshold value will determine the accuracy of our result and confusion matrix. I experimented with different values of T on entire data set and created the confusion matrix.

I got best results at $T = -20$.

Experimentation

For $T = 20$: Results not good, Computation time: 0.031250000000000

n = 200	Predicted- 0	Predicted - 1
Actual : 0	100	0
Actual : 1	98	2

For T = 0, Computation time: 0.015625000000000

n = 200	Predicted- 0	Predicted - 1
Actual : 0	100	0
Actual : 1	59	41

For T = -20 (Best Classification results), Computation time: 0.015625000000000

n = 200	Predicted- 0	Predicted - 1
Actual : 0	100	0
Actual : 1	0	100

Confusion matrix for Training and testing data set at T =-20

Training set:

n = 140	Predicted- 0	Predicted - 1
Actual : 0	72	0
Actual : 1	0	68

Testing Set:

n = 60	Predicted- 0	Predicted - 1
Actual : 0	28	0
Actual : 1	0	32

The results were good for all datasets at threshold value(T) =-20.