Name: Netra Inamdar

CU Username: ninamda

ECE 8560- Pattern Recognition

Take-home #1: Results Report

Engineering decisions and associated rationale:

For case 1, pw1= pw2= pw3= 1/3 and first 5000 samples are in class 1, next 5000 in class 2, last 5000 in class 3.

For case 2, pw1=1/2 pw2=1/3 pw3= 1/6 and first 5000 samples are in class 1, next 5000 in class 2, last 5000 in class 3.

Except for the apriori probabilities, and test dataset distribution, rest all is same for both the cases. Combined approach for both is discussed below:

For training, train_case_1.dat and train_case_2.dat file is used. For classifier design- mean, covariance matrix and correct discriminant function is required.

The required parameters were calculated by MATLAB code as below:

mu_w1 = mean(w1); sigma_w1 = cov(w1);	mu_w2 = mean(w2); sigma_w2 = cov(w2);	mu_w3 = mean(w3); sigma_w3 = cov(w3);

After this, discriminant function based on the gaussian model was calculated for each class, (g1,g2,g3 respectively), for each 4d data vector in test_case_1.dat and test_case_2.dat file. The function for each class is given in the next section of the report.
Then, depending on the discriminant function, the class is selected for which the discriminant function is largest.

```
if g1 > g2 && g1 > g3
	class(i) = 1;
elseif g2 > g1 && g2 > g3
	class(i) = 2;
else
	class(i) = 3;
end
```

The classification results for case 1 are written to file- takehome_case_1.txt, and,
the classification results for case 2 are written to file- takehome_case_1.txt.

Engineering decisions and associated rationale:

For cases where P(wi) != P(wj) and $sigma_i$!= sigma != **I**, We know that, posterior probability is estimated as: P(wi |x) = (p(x |wi) * P(wi)) / p(x)

Density function is taken as: [log P(wi |x)] Hence, discriminant function used here is : g_i(x)= -½ (x-u_i)' * inv($sigma_i$) * (x-u_i) - (d/2)*log(2*pi) - ½ (log(det($sigma_i$))) + log(P(w_i)) - log(p(x))

Here, 2nd and 5th terms are independent of class hence removed from the final equation for discriminant function. 1st term is the mahalanobis distance calculation for each vector in file.

The performance of classifier on training data (for both cases) is checked as follows:

After assigning class labels to data in training files, based on discriminant function, they are compared with the train_labels given in the problem definition: (5000 in each class)

```
if i < 5001
        train_labels(i) = 1;
elseif i > 5000 && i < 10001
        train_labels(i) = 2;
else
        train_labels(i) = 3;
end
```

```
sum = 0;
for i = 1:data_size
    if class(i) ~= train_labels(i)
        sum = sum + 1;
    end
end
```

The sum variable helps to calculate the error on training data classification. The performance accuracy is elaborated in next sections of the report.

Discriminant function used for each class in each case:

Case 1:

For class i: mean=mu_wi and covariance matrix=sigma_wi, then:

g1= -(0.5*(x-mu_w1)*inv(sigma_w1)*(x-mu_w1)') - (0.5*log(det(sigma_w1)))+(log(pw1));

g2= -(0.5*(x-mu_w2)*inv(sigma_w2)*(x-mu_w2)') - (0.5*log(det(sigma_w2)))+(log(pw2));

g3= -(0.5*(x-mu_w3)*inv(sigma_w3)*(x-mu_w3)') - (0.5*log(det(sigma_w3)))+(log(pw3));

(Here, since pw1=pw2=pw3=⅓, third term is same for all the functions)

Case 2:

For class i: mean=mu_wi and covariance matrix=sigma_wi, then:

g1= -(0.5*(x-mu_w1)*inv(sigma_w1)*(x-mu_w1)') - (0.5*log(det(sigma_w1)))+(log(pw1));

g2= -(0.5*(x-mu_w2)*inv(sigma_w2)*(x-mu_w2)') - (0.5*log(det(sigma_w2)))+(log(pw2));

g3= -(0.5*(x-mu_w3)*inv(sigma_w3)*(x-mu_w3)') - (0.5*log(det(sigma_w3)))+(log(pw3));

(Here, pw1=1/2, pw2=1/3 and pw3=1/6)

The classification results for the first 30 samples of each test set:

CASE 1:	CASE 2:
For sample 1 class is 2 For sample 2 class is 3 For sample 3 class is 1 For sample 4 class is 3 For sample 5 class is 1 For sample 6 class is 1 For sample 7 class is 2 For sample 8 class is 3 For sample 9 class is 1 For sample 10 class is 3 For sample 11 class is 1 For sample 12 class is 2 For sample 13 class is 2 For sample 14 class is 3 For sample 15 class is 1 For sample 16 class is 3 For sample 17 class is 1 For sample 18 class is 1 For sample 19 class is 2 For sample 20 class is 3 For sample 21 class is 1 For sample 22 class is 2 For sample 23 class is 1 For sample 24 class is 2 For sample 25 class is 2 For sample 26 class is 3 For sample 27 class is 1 For sample 28 class is 3 For sample 29 class is 1 For sample 30 class is 2	For sample 1 class is 2 For sample 2 class is 3 For sample 3 class is 1 For sample 4 class is 1 For sample 5 class is 1 For sample 6 class is 1 For sample 7 class is 2 For sample 8 class is 3 For sample 9 class is 1 For sample 10 class is 1 For sample 11 class is 1 For sample 12 class is 2 For sample 13 class is 2 For sample 14 class is 3 For sample 15 class is 1 For sample 16 class is 1 For sample 17 class is 1 For sample 18 class is 1 For sample 19 class is 2 For sample 20 class is 3 For sample 21 class is 1 For sample 22 class is 1 For sample 23 class is 1 For sample 24 class is 2 For sample 25 class is 2 For sample 26 class is 3 For sample 27 class is 1 For sample 28 class is 1 For sample 29 class is 1 For sample 30 class is 2

1. <u>Question 1: How well does the classifier classify the training data?</u>

<u>Case 1:</u>

The performance is checked using the file: case1_training_performance.m
The classifier correctly classified **89.85%** of training vectors.

No. of samples in class 1: 5512 ,class 2: 5031 , class 3: 4457

No. of samples in each class after classification is close to 5000. (This can be verified by the command window output displayed in the last section.)

<u>Case 2:</u>

The performance is checked using the file: case1_training_performance.m
The classifier correctly classified **89.47%** of training vectors.

No. of samples in class 1: 5797 ,class 2: 5090 ,class 3: 4113

No. of samples in each class after classification is close to 5000. (This can be verified by the command window output displayed in the last section.)

2. Question 2: Does the classier, used on the given test data, produce a distribution of classes consistent with the pre specified apriori probabilities?

Case 1: Yes, the distribution is consistent with the equal apriori probabilities specification. For test data in case 1, no. of samples in each class after classification is as follows:

```
*** Case 1 test data results: ***
No. of samples in class 1: 5537 ,    class 2: 5012 ,     class 3: 4451
Deviation from ideal results for each class in test set:
class 1: 0.0358       class 2: 0.0008       class 3: 0.0366
>>
```

For class 1, $p(w_i)$=1/3 and no. of samples are close to 5000 (**96.42% accurate**),
For class 2, $p(w_i)$=1/3 and no. of samples are close to 5000 (**99.92% accurate**),
For class 3, $p(w_i)$=1/3 and no. of samples are close to 5000 (**96.34% accurate**).

Case 2: Yes, the distribution is consistent with the unequal apriori probabilities specification. For test data in case 2, no. of samples in each class after classification is as follows:

```
*** Case 2 test data results: ***
No. of samples in class 1: 8010 ,     class 2: 4854,     class 3: 2136
Deviation from ideal results for each class in test set:
class 1: 0.0340       class 2: 0.0097       class 3: 0.0243
>>
```

For class 1, $p(w_i)$=1/2 and no. of samples are close to 7500 (**96.60% accurate**),
For class 2, $p(w_i)$=1/3 and no. of samples are close to 5000 (**99.03% accurate**),
For class 3, $p(w_i)$=1/6 and no. of samples are close to 2500 (**97.57% accurate**).

<u>P(Error) estimation using training data with known class:</u>

<u>Case 1:</u>

As seen above in case of training data performance, the classifier correctly classified 89.85% of training vectors.
The classification error on training data for case 1 is **0.102 or 10.2%**

Command window output for the P(error) estimation:

```
*** Case 1 training data performance results: ***
The classifier correctly classified 89.85% of training vectors.
The classification error on training data is 0.102.

No. of samples in class 1: 5512    ,class 2: 5031         ,class 3: 4457
Deviation from ideal results for each class in training set:
class 1: 0.0341        class 2: 0.0021        class 3: 0.0362
>>
```

<u>Case 2:</u>

As seen above in case of training data performance, the classifier correctly classified 89.47% of training vectors.
The classification error on training data for case 2 is **0.105 or 10.5%**

Command window output for the P(error) estimation:

```
*** Case 2 training data performance results: ***
The classifier correctly classified 89.47% of training vectors.
The classification error on training data is 0.105.

No. of samples in class 1: 5797    ,class 2: 5090         ,class 3: 4113
Deviation from ideal results for each class in training set:
class 1: 0.0531        class 2: 0.0060        class 3: 0.0591
>>
```