ECE8560 Takehome #1
Bayesian Minimum Error Classifier

Name: Madhumita Krishnan

CU Username: mkrishn

02/28/2019

1 Introduction

This report deals with the design and implementation of a Bayesian Classifier for the given data which minimizes classification error. Towards this objective a training set H_i for three classes $w_i, i = 1, 2, 3$ are provided. The classifier designed using the training set is used to classify samples in the test sets S_T. Finally the design is evaluated for its appropriateness and accuracy by measuring the classification errors.

2 Design of Classifier

Classifiers for this problem are developed for two cases:

- The first has equal apriori class probabilities i.e $P(w_i) = 1/c = 1/3$. The training and test data are provided in train_case_1 and test_case_1.
- The second has class probabilities defined as follows $P(w_1) = 1/2$, $P(w_2) = 1/3$,$P(w_3) = 1/6$. The training and test data are train_case_2 and test_case_2.

The training set is used to find the $P(w_i|\underline{x}, H_i)$ for each class w_i i= 1, 2, 3. An application of Bayes Rule yields:

$$P(w_i|\underline{x}, H_i) = \frac{p(\underline{x}|w_i, H_i)P(w_i|H_i)}{\sum\limits_{j=1,2,3} p(\underline{x}|w_j, H_i)P(w_j|H_i)} \tag{1}$$

Since apriori probabilities are independent of the training sets, $P(w_i|H_i)$ reduces to $P(w_i)$. Also it is assumed that H_i conveys information only about parameters in w_i. Hence $p(\underline{x}|w_i, H_i)$=$p(\underline{x}|H_i)$=$p(\underline{x}|w_i)$ The denominator of equation 1 is equal to $p(\underline{x})$. The class conditioned density functions $p(\underline{x}|H_i)$ are assumed to be normal Gaussian distributions given by

$$p(\underline{x}|H_i) = \frac{1}{(2\pi)^2|\Sigma_i|^{1/2}} e^{\frac{-1}{2}(\underline{x}-\underline{\mu_i})^T\Sigma_i^{-1}(\underline{x}-\underline{\mu_i})} \tag{2}$$

since d=4.

2.1 Parameter Estimations

The Maximum Likelihood estimation technique is then used to estimate the parameters μ_i and Σ_i for each class w_i where i= 1, 2, 3. Accordingly we have

$$\hat{\mu_i} = sample\,mean = \frac{1}{n}\sum_{k=1}^{n} \underline{x_k} \tag{3}$$

$$\hat{\Sigma}_i = \frac{1}{n}\sum_{k=1}^{n}(\underline{x_k} - \underline{\hat{\mu_i}})(\underline{x_k} - \underline{\hat{\mu_i}})^T \tag{4}$$

The classifier is built using the above parameters. The co-variance matrices estimated using Equation 4 are non-diagonal and shows class dependence.We obtain the corresponding discriminant functions. Simplifying the discriminant functions for the two cases we can classify our test data by concluding that the sample belongs to the class that maximizes the discriminant function.

2.2 Discriminant Functions

The discriminant function for the i^{th} class $i = 1, 2, 3$ used in classification is given as follows

$$g_i(x) = log[P(w_i|\underline{x}, H_i)P(\underline{x})] \tag{5}$$

$$= log[P(\underline{x}|H_i)] + log[P(w_i)] \tag{6}$$

Using Equation 2 and neglecting $-2\pi(log(2\pi))$ as it is class independent we obtain

$$g_i(x) = -\frac{1}{2}||\underline{x} - \underline{\mu_i}||_{\Sigma_i^{-1}} - \frac{1}{2}log(|\Sigma_i|) + log(P(w_i)) \tag{7}$$

Here the log function is used to reduce complex calculations and because any monotonically increasing functions (such as log function) of $P(w_i|\underline{x}, H_i)$ is a valid discriminant function.

For the first case in which we have equal apriori probabilities $P(w_i) = 1/c$, We can ignore the $log(P(w_i))$ term in Equation 7 to obtain the discriminant function since it is a constant. So the discriminant function for the first case is given by Equation 8

$$g_i(x) = -\frac{1}{2}||\underline{x} - \underline{\mu_i}||_{\Sigma_i^{-1}} - \frac{1}{2}log(|\Sigma_i|) \tag{8}$$

For the second case, the discriminant function given by Equation 7 is used since we are dealing with the classification of unequal apriori probabilities.

2.3 Classification Rule

For both cases the classification rule is as follows- Classify $\underline{\text{x}}$ in w_i if $g_i(x)$ is maximum, where i= 1, 2, 3. Below is a snippet of the code which shows the classification rule in both cases.

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

```
else g3(i) > g1(i) && g3(i) > g2(i);
    class(i)=3;
end
```

Based on this classification rule we classify our test data into the three classes.

3 Classification Results

Below are the classification results for the first thirty samples of both the test sets.

```
Case1:
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
```

```
Case2:
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
```

3.1 Answers to Question #1

In the first case when train_case_1 was classified using the Bayesian Classifier which was built using the discriminant function given by Equation 8, the percentage of error $P(error)$ is found to be 10.15%.

The classifier has shown large error due to misclassification of w_3. Since the apriori probabilities of the different classes are equal, the misclassification error of w_3 contributes a lot towards the total $P(error)$.

In the second case where train_case_2 was classified using the Bayesian Classifier which was built using the discriminant function in Equation 7, the percentage of error is found to be 7.67%.

The classifier has shown large error of misclassification in w_3. But since the apriori probability of w_3 is the least, the percentage error is not greatly affected by this. So we can say that the quality of the classifier is not influenced greatly by the misclassification of w_3.

3.2 Answers to Question #2

In the first case, the samples in test_case_1 were classified. The class distribution is as follows- 5537 samples in w1, 5012 samples in w2 and 4451 samples in w_3. Using these results probability of distribution of classes w_1, w_2 and w_3 are :

$$P(w_1) = 0.369$$
$$P(w_2) = 0.334$$
$$P(w_3) = 0.297$$

The classification results are not consistent with the pre-specified apriori probabilities which is 1/3=0.3333 in each case. Probability of distribution of class w_3 is significantly reduced.

In the second case when samples in test_case_2 were classified, the class distribution is as follows- 8010 samples in w_1, 4853 samples in w_2 and 2137 samples in w_3. Using these results the probability of distribution of classes w_1,w_2 and $w3$ are as follows:

$$P(w_1) = 0.534$$
$$P(w_2) = 0.324$$
$$P(w_3) = 0.142$$

The pre-specified apriori probabilities of the classes are $P(w_1) = 0.5$, $P(w_2) = 0.333$ and $P(w_3) = 0.167$. The classifier results are consistent with pre-specified apriori probabilities with minimum error.

4 Estimation of P(Error)

In the first case using MATLAB the number of samples in the training set that are misclassified by our classifier is estimated . From this the following results are obtained:
P(decide $w_2 \cap$ really w_1)=144
P(decide $w_3 \cap$ really w_1)=87
P(decide $w_1 \cap$ really w_2)=354
P(decide $w_3 \cap$ really w_2)=154
P(decide $w_1 \cap$ really w_3)=389
P(decide $w_2 \cap$ really w_3)=395

$$\begin{aligned} P(error) = {} & P(\underline{x} \in w_2|w_1)P(w_1) + P(\underline{x} \in w_3|w_1)P(w_1) + P(\underline{x} \in w_1|w_2)P(w_2) \\ & + P(\underline{x} \in w_3|w_2)P(w_2) + P(\underline{x} \in w_1|w_3)P(w_3) + P(\underline{x} \in w_2|w_3)P(w_3) \end{aligned} \tag{9}$$

$$P(error) = \tfrac{144}{5000} * \tfrac{1}{3} + \tfrac{87}{5000} * \tfrac{1}{3} + \tfrac{354}{5000} * \tfrac{1}{3} + \tfrac{154}{5000} * \tfrac{1}{3} + \tfrac{389}{5000} * \tfrac{1}{3} + \tfrac{395}{5000} * \tfrac{1}{3}$$

$$P(error) = 0.1015$$

$$\text{Percentage of } P(error) = 10.15\%$$

In the second case using MATLAB the number of samples in the training set that are misclassified by our classifier is estimated. Following are the results obtained:
P(decide $w_2 \cap$ really w_1)=96
P(decide $w_3 \cap$ really w_1)=24
P(decide $w_1 \cap$ really w_2)=415
P(decide $w_3 \cap$ really w_2)=67
P(decide $w_1 \cap$ really w_3)=502
P(decide $w_2 \cap$ really w_3)=476

$$\begin{aligned} P(error) = {} & P(\underline{x} \in w_2|w_1)P(w_1) + P(\underline{x} \in w_3|w_1)P(w_1) + P(\underline{x} \in w_1|w_2)P(w_2) \\ & + P(\underline{x} \in w_3|w_2)P(w_2) + P(\underline{x} \in w_1|w_3)P(w_3) + P(\underline{x} \in w_2|w_3)P(w_3) \end{aligned} \tag{10}$$

$$P(error) = \tfrac{96}{5000} * \tfrac{1}{2} + \tfrac{24}{5000} * \tfrac{1}{2} + \tfrac{415}{5000} * \tfrac{1}{3} + \tfrac{67}{5000} * \tfrac{1}{3} + \tfrac{502}{5000} * \tfrac{1}{6} + \tfrac{476}{5000} * \tfrac{1}{6}$$

$$P(error) = 0.0767$$

$$\text{Percentage of } P(error) = 7.67\%$$