Ravisutha Sakarepatna Srinivasamurthy

A. Training using training dataset

1. Choosing the suitable PDF

In order to represent the data in the closed form, analysis of the give data is necessary. Histogram for the training data for each random variable belonging to all classes are plotted and is shown in Figure 1.1 (eg. ClassA – [xA1; xA2; xA3; xA4]).

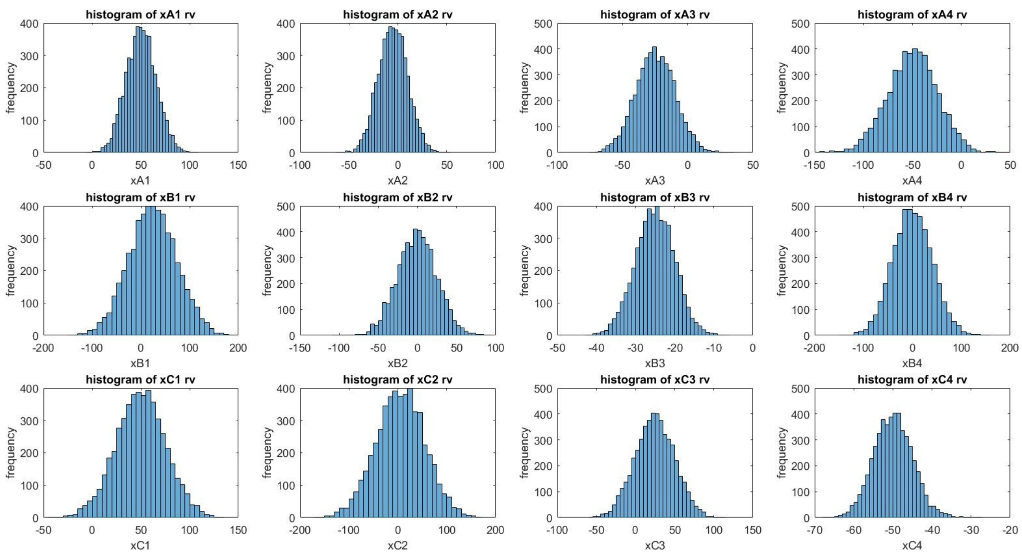

Figure 1.1 Histogram of Training set

It is very clear form the histogram that it follows a bell curve. Hence, in order to represent the data, Gaussian distribution was chosen.

2. Estimating the parameter

In order to estimate the data, one can follow either of two approaches: Maximum Likelihood or Bayesian Parameter Estimation. Bayesian approach requires prior knowledge or belief or expectation of the possible parameters Θ. By assuming that “generating pdf” is a normal distribution with same mean and identity matrix as covariance did not yield any better results than MLE. Hence, Maximum Likelihood was retained. The two parameters which characterizes Gaussian are mean (μ) and covariance (∑). Mean and covariance matrix for all three classes are calculated by using the equations shown below.

$$\mu_i = \frac{\sum_{j=1}^{N} x_j}{N} \tag{1}$$

$$\sum i = E\left[\left(X_i - E\left[X_i\right]\right)\left(X_i - E\left[X_i\right]\right)^T\right] \tag{2}$$

where i : can take on class 1, 2 and 3 and X: Feature vector(4 * 1)

Hence, the Gaussian pdf for all three classes are estimated.

$$p(x \vee w i) \sim Normal(\mu_i, \sum i)$$

3. A priori probability

A priori probability of the three classes are unknown. Hence the probability of all three classes are assumed to be same.

$$P(w1) = P(w2) = P(w3) = \frac{1}{3} \tag{3}$$

B. Design of the classifier

Bayesian Classifier is used to classify the data into three classes. Using the probability density function based on training data and assumed a priori probability for classes, discriminant function is derived. Using the discriminant function, the test data will be classified into one of the three classes.

$$g_i(x) = \log\{p(x|w_i ¿\}$$

(4)

$$g_i(x) = \frac{-1}{2}(x-\mu_i)^T inv(\sum i)(x-\mu_i) - \frac{1}{2}\log(det(\sum i)) \tag{5}$$

C. Testing on test data

1. Classification

The discriminant function g_i is calculated for all three classes. The decision boundary (hyperplanes) which separate different classes are given by

$$H_{12}: g1 - g2 = 0 \tag{6}$$

$$H_{13}: g1 - g3 = 0 \tag{8}$$

$$H_{23}: g2 - g3 = 0 \tag{7}$$

The hyperplanes H12, H13 and H23 separate different classes (but classes do overlap). The Bayesian classifier works as follows:

If a vector x belongs to class A then, g1 must be greater than g2 and g1 must be greater than g3. Or, equivalently, g1 – g2 should be positive and g2 – g3 must also be positive. Similar strategy is used to classify a data to class B and class C.

D. **Discriminant Function**

The discriminant function can be written as

$$g_i(x) = w_i^T x - w_{i0} - \left(\frac{d}{2}\right)\log(2\Pi) - \frac{1}{2}\log\left(det\left(\sum i\right)\right) \qquad (8)$$

where $w_i = inv\left(\sum i\right)\mu_i$ and $w_{i0} = -\left(\frac{1}{2}\right)\|\mu_i\|^2_{\sum^{-1}}$

As third term is constant and common to all classes, it can be ignored.

$$g_i(x) = w_i^T x - w_{i0} - \frac{1}{2}\log\left(det\left(\sum i\right)\right) \qquad (9)$$

For Class A,

$$g_1(x) = [0.2210 - 0.0254 - 0.1159 - 0.0818]^T x + 176.1318 \qquad (10)$$

For Class B,

$$g_2(x) = [0.0095\ 0.0034 - 0.9793\ 0.0022]^T x + 318.9215 \qquad (11)$$

For Class C,

$$g_3(x) = [0.0867\ 0.0020\ 0.0325 - 2.0398]^T x + 5734.7 \qquad (12)$$

Ravisutha Sakarepatna Srinivasamurthy

E. **Probability of Error**

Probability of error is given as follows:

$$Classify\ x\ as\ A \wedge x\ belongs$$
$$Classify\ x\ as\ A \wedge x\ belongs$$
$$Classify\ x\ as\ B \wedge x\ belongs$$
$$Classify\ x\ as\ B \wedge x\ belongs$$
$$Classify\ x\ as\ C \wedge x\ belongs$$
$$Classify\ x\ as\ C \wedge x\ belongs$$
$$P(error) = P(¿B) + P(¿C) + P(¿A) + P(¿C) + P(¿A) + P(¿B)$$

Table gives the classification details for training data:

	A	**B**	**C**
ClassA	4539	318	143
ClassB	577	4389	34
ClassC	236	50	4714

Hence the probability of error is calculated as follows:

$$P(error) = \frac{577}{15000} + \frac{236}{15000} + \frac{318}{15000} + \frac{50}{15000} + \frac{143}{15000} + \frac{34}{15000}$$

$$P(error) = 0.09053$$

$$P(error) = 9.053$$

Hence the probability of error estimation using training data is 9.053%.

F. **Result**

Test data has been classified and the result is stored in ravisutha-classified-takehome1.txt.

Appendix 1: Matlab Implementation

A. File Reading

```
function [Class] = get_data()
        fileID = fopen (‘filename’, 'r');

    formatSpec = '%f %f %f %f';

    for j=1:15000
        Class(:,j) = fscanf (fileID, formatSpec, 4);
    end

    fclose('all');
end
```

B. Calculate Class

```
function [decision] = calulate_class(theta1, theta2, theta3, x)
    %   theta   : Parameters
    %   x       : Input vector
    %   decision: Class 1,2 or 3

    %   C : Covariance
    %   M : Mean
    C1 = theta1(:,2:end);
    M1 = theta1(:,1);
    C2 = theta2(:,2:end);
    M2 = theta2(:,1);
    C3 = theta3(:,2:end);
    M3 = theta3(:,1);

    %   iC: Inverse of Covariance
    iC1 = inv(C1);
    iC2 = inv(C2);
    iC3 = inv(C3);

    % Discriminant Function for each class
    decision1 = (-0.5 * (x - M1)' * iC1 * (x - M1)) - 0.5 * log(det (C1));
    decision2 = (-0.5 * (x - M2)' * iC2 * (x - M2)) - 0.5 * log(det (C2));
    decision3 = (-0.5 * (x - M3)' * iC3 * (x - M3)) - 0.5 * log(det (C3));

    %Classify 'x' into one of three classes
    if (decision1 >= decision2 & decision1 >= decision3)
        decision = 1;

    elseif (decision2 >= decision1 & decision2 >= decision3)
        decision = 2;

    elseif (decision3 >= decision1 & decision3 >= decision2)
        decision = 3;

    else
        decision = 2;
    end
```

```
end
```

C. Covariance and Mean

```
function [theta1, theta2, theta3] = mean_covariance()

	[Class] = File_opening();
	ClassA = Class(:, 1:5000);
	ClassB = Class(:, 5001:10000);
	ClassC = Class(:, 10001:15000);

	% Mean vector for each class
	meanA = mean (ClassA, 2);
	meanB = mean (ClassB, 2);
	meanC = mean (ClassC, 2);

	% Covariance Vector for each class
	C1 = cov (ClassA')
	C2 = cov (ClassB')
	C3 = cov (ClassC')

	theta1 = [meanA C1];
	theta2 = [meanB C2];
	theta3 = [meanC C3];
end
```