CS 479/679 Pattern Recognition

Programming Assignment 1

Bayes Decision Theory

Jesus Manuel Aguilera
Eduardo Arce-Gutierrez

The programming assignment was divided mainly by collaboration and not by preassigned tasks. We both contributed with the Programming part as well as the report elaboration and revision.

1 Introduction

We are to consider a two-class classification problem where the data in each class is modeled by a two-dimensional gaussian density. Gaussian pseudo-random values are generated under a box-muller transformation given a source of uniform pseudo-random values (range from 0 to 1). The polar form version of the box-muller transformation takes in two parameters, the mean and standard deviation of each normal distribution and outputs random values that model a gaussian distribution. We are to design a bayes classifier for minimum error that determines whether a given set of random values, x and y, belong to the first normal distribution or second normal distribution using the fundamentals of bayesian decision theory.

2 Theory

2.1 Bayesian Decision Theory

Bayesian decision theory is a statistical approach to classifying patterns in large sets of data. The classifier uses Bayes' rule to make decisions on the data to determine whether a given set belongs to one category or another. It is important to note that bayesian decision theory assumes that all of the relevant probability values are known so that the classifier is able to make such decisions based on probabilistic circumstances. There must be some prior knowledge of how likely a class is to be the best decision to make under an event (feature) is to be presented for classification. Assuming that the same loss is assigned to every possible error during classification, we can design a Bayes classifier that maximizes the posterior probability for each class and choose the highest yielding value to make a decision. In order for the classifier to make the optimal decision, the classifier must choose a class that maximizes the posteriori probability given a sample set. In a case of two categories:

$$\text{Decide } \omega_1 \text{ if } P(\omega_1/x) > P(\omega_2/x) \text{otherwise decide } \omega_2 \tag{1}$$

Rearranging Eq. 1 using Bayes rule, classification becomes:

$$\text{Decide } \omega_1 \text{ if } \frac{p(x/\omega_1)\,P(\omega_1)}{p(x)} > \frac{p(x/\omega_2)\,P(\omega_2)}{p(x)} \text{, otherwise decide } \omega_2 \tag{2}$$

where $p(x/\omega_i)$ is the the conditional probability density that measures how frequently the classifier measures a pattern with feature x given that the feature belongs to class ω_i, $P(\omega_i)$ is the prior probability of a how likely classification results in being from class ω_i, and $p(x)$ is the frequency of a pattern being measured given feature x.

2.2 Discriminants

Using Bayes theorem we can estimate class-conditional posterior probabilities given a set of points in the n-dimensional input space, and assign these points to a class with the maximum posterior probability. This means that the n-dimensional input space is divided into regions that correspond to a particular class. We use the conditional probability density, also known as likelihood, to decide whether a feature belongs to one class or another. Since the likelihood is scaled by the prior probability, we can compare the scaled likelihoods to determine where in the input space the feature lands based on probabilistic computations. A function that defines the regions in the input space is called a discriminant function. This discriminant function is used to evaluate a sample of events to determine whether these events truly belong to the designated class. Since the n-dimensional input space is divided into regions corresponding to the amount of classes, there must be a discriminant function for n number of regions. An optimum classifier compares all values computed from the discriminant functions and makes a decision that a sample set belongs to a class based on the maximum value. Setting the discriminant functions equal results in a boundary that divides the regions called the decision boundary. There are many forms of discriminants that define the regions in the input space and different forms of decision boundaries (i.e., linear vs non-linear) when the likelihood is modelled. A specific application where there are different variations of discriminant functions and decision boundaries is in a multivariate gaussian distribution.

2.3 Multivariate Gaussian

In the case of two-class classification where the data in each class is modeled by a two-dimensional gaussian density, we can convert Bayes rule such that the discriminant is modeled by a monotonically increasing function (the value of the discriminant increases for every input). We consider the following discriminant function for a multivariate gaussian distribution:

$$g_i\left(x\right) = \ln p\left(x/\omega_i\right) + \ln P\left(\omega_i\right) \tag{3}$$

Assuming that the conditional density probability follows the following gaussian distribution:

$$N\left(\mu, \Sigma\right) = \frac{1}{\left(2\pi\right)^{\frac{d}{2}} \left|\Sigma\right|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}\left(x-\mu\right)^t \Sigma^{-1}\left(x-\mu\right)\right] x \in R^d \tag{4}$$

We can rewrite the discriminant (Eq. 3) using Eq. 4:

$$g_i\left(x\right) = -\frac{1}{2}\left(x-\mu_i\right)^t \Sigma_i^{-1}\left(x-\mu_i\right) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln\left|\Sigma_i\right| + \ln P\left(\omega_i\right) \tag{5}$$

where Eq. 5 describes the discriminant for a multivariate gaussian distribution. When working with multiple variables, the covariance matrix provides a distinction between the joint variability of all the pairs of random variables. It describes the shape of the multivariate gaussian distribution, so we must consider multiple cases of the covariance matrix that define the discriminant function in more detail.

Consider a case where the covariance matrix is a diagonal matrix with equal diagonal elements. This yields a hyperspherical cluster of values where each component has the same variance. Then the discriminant functions becomes:

$$g_i\left(x\right) = \left(\frac{1}{\sigma^2}\mu_i\right)^t x + \left(-\frac{1}{2\sigma^2}\mu_i^t\mu_i + \ln P\left(\omega_i\right)\right) \tag{6}$$

where x is the features in n-dimensions. The decision boundary can be found by setting the discriminant equal. The following is the equation for the decision boundary given a classification belonging to this case:

$$w^t\left(x - x_0\right) = 0 \tag{7}$$

where:

$$w = u_i - u_j$$

$$x_0 = \frac{1}{2}\left(u_i + u_j\right) - \frac{\sigma^2}{||u_i - u_j||\,2}\ln\left(\frac{P\left(w_i\right)}{P\left(w_j\right)}\right)\left(u_i - u_j\right)$$

Another case for the gaussian distribution is when the covariance matrix is scaled by some value such that the values fall under an ellipsoid cluster. Such a case results in the discriminant becoming:

$$g_i\left(x\right) = \left(\Sigma^{-1}\mu_i\right)^t x + \left(-\frac{1}{2}\mu_i^t\Sigma^{-1}\mu_i + \ln P\left(\omega_1\right)\right) \tag{8}$$

The decision boundary for this case is defined as:

$$g_i\left(x\right) = w_i^t x + w_{i0} \tag{9}$$

where:

$$w_i = \Sigma^{-1}u_i$$

$$w_{i0} = -\frac{1}{2}u_i^t\Sigma^{-1}u_i + \ln P\left(w_i\right)$$

The final case is when the covariance matrix contains arbitrary values that are not necessarily in the diagonal sections of the matrix. This case results in the cluster skewing the orientation of the gaussian so that the cluster of random values result in unique shapes and sizes. This means that the components within the covariance matrix are not statistically independent and result in non-linear decision boundaries. In this case, the discriminant becomes:

$$g_i\left(x\right) = x^t\left(-\frac{1}{2}\Sigma_i^{-1}\right)x + \left(\Sigma_i^{-1}\mu_1\right)^t x - \frac{1}{2}\mu_i^t\Sigma_i^{-1}\mu_i - \frac{1}{2}\ln\left|\Sigma_i\right| + \ln P\left(\omega_i\right) \quad (10)$$

Converting Eq.10 to be easily understood, the following equation is used to determine the discriminant in a case where the covariance matrix contains arbitrary values that are not necessarily diagonal.

$$g_i\left(x\right) = x^t W_i x + w_i^t x + w_{i0} \quad (11)$$

where:

$$W_i = -\frac{1}{2}\Sigma_i^{-1}$$

$$w_i = \Sigma_i^{-1}u_i$$

$$w_{i0} = -\frac{1}{2}u_i^t\Sigma_i^{-1}u_i - \frac{1}{2}\ln\left|\Sigma_i\right| + \ln P\left(w_i\right)$$

2.4 Probability of Error

The operation of a general classifier is dependent on decision making. This means that we must also consider the probability of the classifier making the wrong decision. In a case of classification, an error is defined by the classification of an input belonging to the wrong class. The probability of error is defined as the frequency at which a certain decision made by the classifier will lead to the wrong decision. Since such events in probability are mutually exclusive, we are able to estimate the bound of error in the case of two-class classification as:

$$P\left(error\right) = \int \min\left[p\left(x/\omega_1\right)P\left(\omega_1\right), p\left(x/\omega_2\right)P\left(\omega_2\right)\right]dx \le P^{\beta}\left(\omega_1\right)P^{1-\beta}\left(\omega_2\right)\int p^{\beta}\left(x/\omega_1\right)p^{1-\beta}\left(x/\omega_2\right)dx \quad (12)$$

In a two-category case, the general error of the integral can be approximated analytically to give us the upper bound on the error. If the class conditional distributions are Gaussian, then the probability of error becomes:

$$P\left(error\right) \le P^{\beta}\left(\omega_1\right)P^{1-\beta}\left(\omega_2\right)e^{-k\left(\beta\right)} \quad (13)$$

where

$$k\left(\beta\right) = \frac{\beta\left(1-\beta\right)}{2}\left(\mu_1-\mu_2\right)^t\left[\left(1-\beta\right)\Sigma_1+\beta\Sigma_2\right]^{-1}\left(\mu_1-\mu_2\right) + \frac{1}{2}\ln\left(\frac{\left|\left(1-\beta\right)\Sigma_1+\beta\Sigma_2\right|}{\left|\Sigma_1\right|^{1-\beta}\left|\Sigma_2\right|^{\beta}}\right) \quad (14)$$

A slightly less tight bound compared to the Chernoff bound in the probability of error is called the Bhattacharyya bound. It is a special case where $\beta = 0.5$.

2.5 Euclidean Distance Classifier

An alternative to a linear discriminant function is the euclidean distance classifier. Under certain conditions, the euclidean distance classifier performs better compared to the linear discriminant functions. As the number of features in the classifier increases, the linear discriminant performs poorly since the number of features is large relative to the size of the data during training. If the prior probabilities are equal, then the discriminant becomes:

$$g_i(x) = -\left((x - \mu_i)^t (x - \mu_i)\right) \tag{15}$$

where x is an n-dimensional vector defined in the n-dimensional input space. The euclidean distance classifier calculates a distance between two points in the input space. For every value within the feature vector x, the euclidean distance classifier assigns the feature to a class whose mean is closest to the value of every pixel.

3 Implementation

3.1 Linear Algebra Computations

In order to design a two-category classifier using the fundamentals of bayesian decision theory, we decided to include a third-party C++ library that is not a part of the standard set of libraries. This allowed us to perform linear algebra computations in a more efficient way. Fastor is a high performance fixed multi-dimensional array (Tensor) library made for high-level linear algebra computations. This includes element-wise binary operations, inverse, determinants, transpose, inner-product, etc. This allowed us to implement Eqs. 1-12 in C++ more efficiently. https://github.com/romeric/Fastor.

3.2 Data Generation

Given a source of uniform pseudo-random numbers (range from 0 to 1), the box-muller transformation allowed us to generate gaussian pseudo-random numbers. The box-muller algorithm accepts two independent random numbers that come from a uniform distribution and generates an equal size pair of independent, normally distributed random numbers. The normally distributed random numbers result in having zero mean and a standard deviation of one since it is a symmetric distribution. The random values generated from the box-muller algorithm cluster around the central peak of the gaussian curve and the probabilities of the values are equal on both sides of the curve. This is the exact definition of a gaussian distribution.

In the case of two-class classification, we need to call the box-muller algorithm twice since the provided C code only generated samples from a one-dimensional gaussian distribution. This allowed us to generate pseudo-gaussian random values in the two-dimensional sample space.

3.3 Misclassification Rate

As each value from the feature vector was generated from the box-muller algorithm, the samples were fed into a method that calculated the value for the discriminant function. At each iteration of a new set of randomly generated samples, this resulted in two different discriminant functions for each 60,000 (or 140,000) samples. In a case of two categories, it is more common to use a single discriminant function called the dichotomizer instead of two. The dichotomizer is defined as follows:

$$g(x) = g_1(x) - g_2(x) \tag{16}$$

Decide w_1 if $g(x) > 0$; otherwise decide w_2

If the dichotomizer resulted in a positive number, two predefined variables will be incremented to simulate a count for correct and incorrect classifications. This was done for each normal distribution. This allowed us to report the misclassification rate for each class separately and the total misclassification rate for each data set.

4 Results and Discussion

4.1 Data Set A

4.1.1 Data Generation

Generate 60,000 random samples from $N(\mu_1, \Sigma_1)$with the box-muller algorithm using the following parameters:

$$\mu_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \ \Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Generate 140,000 random samples from $N(\mu_2, \Sigma_2)$ with the box-muller algorithm using the following parameters:

$$\mu_2 = \begin{bmatrix} 4 \\ 4 \end{bmatrix} \ \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Fig. 1 shows the generated data on the same plot with the decision boundary in yellow. Proper computation on how the decision boundary was calculated is mentioned in section [4.1.4].

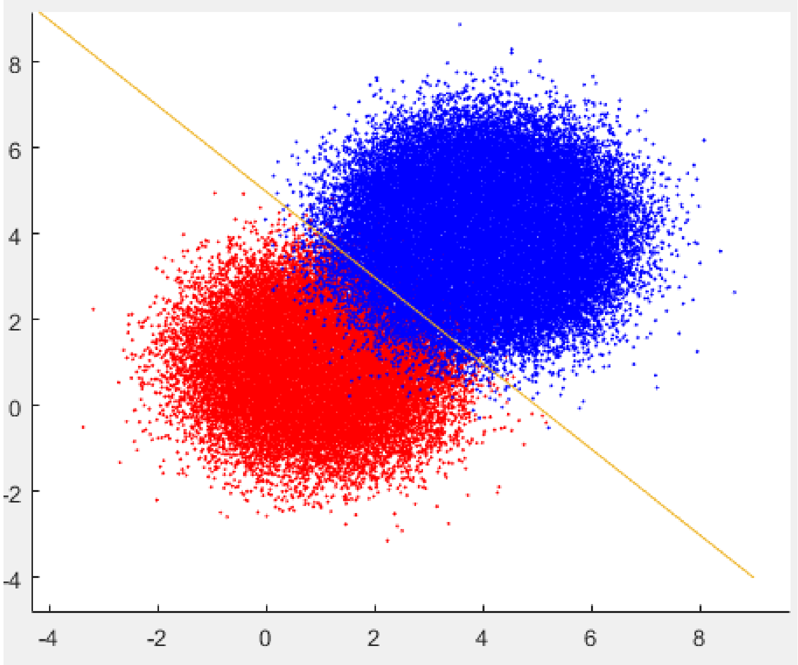

Figure 1: Data set A with the discriminant decision boundary. The blue cluster corresponds to the 140,000 random samples from$N(u_2, \Sigma_2)$while the red cluster corresponds to the 60,000 random samples from $N(u_1, \Sigma_1)$.

4.1.2 Prior probabilities

Since there is no evidence that determines a randomly generated sample belongs to a particular class, the prior probabilities were set to equal values. In a case of two-class classification, the probability of a randomly generated sample belonging to class one was set to 0.50 as well as 0.50 for class two. This allowed for much simpler computations for the discriminant decision boundary. Since $P(w_1) = P(w_2)$, Eq. 7 becomes:

$$w^t (x - x_0) = 0 \tag{17}$$

where

$$w = u_i - u_j$$

$$x_0 = \frac{1}{2} (u_i + u_j)$$

since

$$\ln \left(\frac{P(w_i)}{P(w_j)} \right) = \ln \left(\frac{0.50}{0.50} \right) = 0$$

4.1.3 Covariance Matrix Case

The discriminant function for data set A was determined by the mean and covariance matrix of the data set as well as the visualization of the generated data. Since the generated data form hyperspherical clusters for both normal distributions, Eq. 6 was used to describe the discriminant function in each region of the sample space (class 1 and class 2). The generated data correctly follow the bayesian decision theory for determining the discriminant function based on the infrastructure of the covariance matrix. Since the covariance matrix for both normal distributions form a diagonal and equal element matrix, the data is generated as expected.

4.1.4 Decision Boundary

The decision boundary for data set A was calculated with Eq. 7. Since the prior probabilities are equal for both classes, we can use Eq. 17 to solve for the decision boundary. This resulted in the following equation:

$$-3x - 3y + 15 = 0$$

Fig 1. shows the decision boundary plotted with each normal distribution cluster.

4.1.5 Results

Table 1. shows an ordered chart with all computed results for data set A using a linear discriminant function (Eq. 6). This includes the misclassification rate for each class. The Bayes classifier was designed to classify the random samples with the minimum possible rate of error. The Bayes classifier for data set A, using a linear discriminant function, classified correctly at a rate of 98.34%. This means that the random sample generated from a particular gaussian distribution was correctly classified 98.34% of the time.

Table 1: Computed misclassification rates using a linear discriminant function for data set A.

Linear	Number of misclassified samples	Percent of misclassification
Misclassified rate for class one	995	1.66%
Misclassification rate for class two	2371	1.69%

Data set A also included an euclidean distance classifier to compare to the linear discriminant function. The discriminant function was replaced with Eq. 15 for each class. There are certain requirements that the set of data must fulfill inorder to make the euclidean distance classifier the optimum solution to perform classification for a case of two classes. The prior probabilities for both classes must be equal. Since the size of data for classification did not exceed the size of the training data set for the Bayes classifier, the euclidean distance classifier is expected to perform equally or better than the linear discriminant approach. Table 2. shows that the euclidean distance classifier performed equally to the linear discriminant function. Table 2. is not a typo, these are the actually results for a euclidean distance classifier. Since there are no distinct differences between choosing the linear discriminant or euclidean distance classifier for this data set, the euclidean distance classifier is an optimum approach since it requires less computation.

Table 2: Computed misclassification rates using an euclidean distance classifier for data set A.

Euclidean	Number of misclassified samples	Percent of misclassification
Misclassified rate for class one	995	1.66%
Misclassification rate for class two	2371	1.69%

The theoretical probability error using bhattacharyya bound was also computed for data set A and compared to the total misclassification rate. Since the misclassification rate for both gaussian distributions resulted in equal values, the total misclassification rate was only computed once for both the linear discriminant and euclidean distance classifier. The computed total misclassification rate for data set A was 1.683%. This means that the Bayes classifier performs correct classifications 98.317% of the time. In order to compute the probability of error, the value for the bhattacharyya bound needed to be computed. Using Eq. 14, the resulting value was 2.25. This is the approximated upper bound on the probability that the Bayes classifier is to make an error during classification. In order to compute the probability of error using the value of the bhattacharyya bound, Eq.13 was modified to become:

$$P\left(error\right) = \sqrt{P\left(w_1\right)P\left(w_2\right)}e^{-k(\beta)}$$

where $k(\beta)$is the value of the bhattacharyya bound. Table 3. shows the total misclassification rate and the probability of error for data set A. Since the probability of error is higher than the total misclassification rate, this tells us that the highest possible rate of error for the Bayes classifier of a random outcome is 5.27%. This is expected since the total misclassification rate is lower than the upper bound probabilistic error.

Table 3: Total misclassification rate for data set A using a linear discriminant function and euclidean distance classifier. The probability of error is also listed and defines the probability that the classifier is to make the wrong decision.

	Percent
Total misclassification rate	1.683%
Probability of error≤	0.053 (5.27%)

4.2 Data Set B

4.2.1 Data Generation

Generate 40,000 random samples from $N(u_1, \Sigma_1)$with the box-muller algorithm using the following parameters: $\mu_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Generate 160,000 random samples from$N(u_2, \Sigma_2)$with the box-muller algorithm using the following parameters:

$$\mu_2 = \begin{bmatrix} 4 \\ 4 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 4 & 0 \\ 0 & 8 \end{bmatrix}$$

Fig. 2 shows the random samples plotted on the same graph along with the decision boundary.

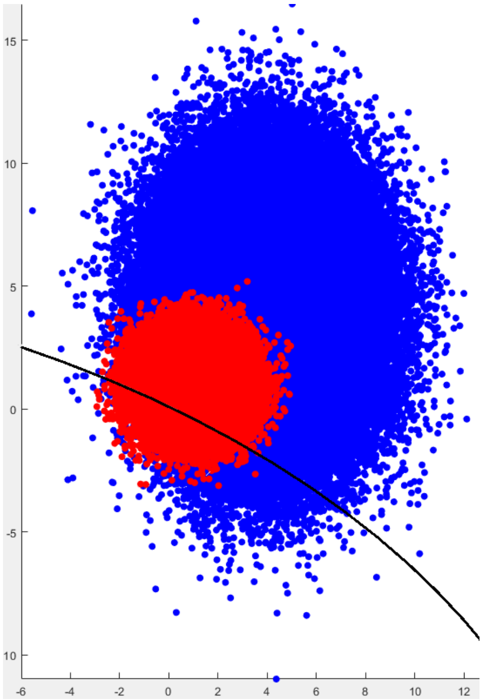

Figure 2: Random sampling from both normal distributions in data set B. The blue cluster corresponds to the second normal distribution, while the red cluster corresponds to the first normal distribution. The decision boundary is modelled by the black line.

4.2.2 Prior probabilities

The same approach used with set A was used for set B. The prior probabilities were set to equal values. In a case of two-class classification, the probability of a randomly generated sample belonging to class one was set to 0.50 as well as 0.50 for class two.

4.2.3 Covariance Matrix Case

Since the covariance matrix for class one and class two are different, we decided to use the case of the covariance matrix having arbitrary values that are not necessarily diagonal. Eq. 10 was used to describe the discriminant function in each region of the sample space. Since the covariance matrix for both normal distributions have different variance values and are arbitrary, the visualization of data is generated as clusters of different shapes and sizes.

4.1.4 Decision boundary

In order to calculate the decision boundary for data set B, we used Eq. 11. and set the discriminant equal for both classes. This resulted in the following equation:

$$\frac{3}{8}x^2 + \frac{7}{16}y^2 + 15x + 31y - 3.73 = 0$$

After plotting the above equation we found that it describes a circle that cuts the samples areas in two as shown in figure 2.
Because the formula is quadratic, resolving for y results in two different equations with a real and imaginary part. So we plotted the original formula instead using MatLab.

4.1.5 Results for Set B

Table 4. shows the classification errors and the percentage of misclassification for each class. We obtained those results using a quadratic discriminant function (Eq. 10). The Bayes classifier was designed to classify the random samples with the minimum possible rate of error. The Bayes classifier for data set B using a quadratic discriminant function classified correctly at a rate of 91.09% using the quadratic discriminant function. This means that the random sample generated from a particular gaussian distribution was misclassified 8.913% of the time.

Table 4: Computed misclassification rates using a quadratic discriminant function for data set B.

Quadratic function Classifier	Number of misclassified samples	Percent of misclassification
Misclassified rate for class one	1559	3.90%
Misclassification rate for class two	16,267	10.17%

Next, the discriminant function was replaced with Eq. 15 for each class. In contrast with the results from Set A, the results on set B show that the Euclidean distance classifier is not an optimum choice for the classification. Table 5. shows that the euclidean distance classifier performed worse compared to the quadratic discriminant function since it favored one class over the other.

Table 5: Computed misclassification rates using an euclidean distance classifier for data set B.

Euclidean	Number of misclassified samples	Percent of misclassification
Misclassified rate for class one	684	1.14%
Misclassification rate for class two	30,986	22.13%

Since the misclassification rate for the distributions resulted in different values, the total misclassification rate was computed for both the quadratic discriminant and euclidean distance classifier. The computed total misclassification rate for data set B was 8.913% using the quadratic discriminant and 27.354% using the Euclidean distance classifier.

The same method was used to compute the probability of error and Eq. 14 resulted in a bhattacharyya bound of 0.7. Table 6. shows the total misclassification rate using a linear discriminant function, total misclassification rate using euclidean distance classifier, and the probability of error for data set B. Since the probability of error is higher than the total misclassification rate, this tells us that the highest possible rate of error for the Bayes classifier of a random outcome is 24.8%. This is expected since the total misclassification rate is lower than the upper bound probabilistic error. This also means that the euclidean distance classifier is not the optimum approach to classification for this data set, following arbitrary covariance matrix since there is a distinct difference in misclassification rates. The euclidean distance classifier also exceeds the upper error bound.

Table 6: Total misclassification rate for data set B using a quadratic discriminant function and euclidean distance classifier. The probability of error is also listed and defines the probability that the classifier is to make the wrong decision.

	Percent
Total misclassification rate using linear discriminant	8.913%
Total misclassification rate using euclidean	27.354%
Probability of error≤	0.248 (24.8%)