|  |
| --- |
| **Maximum Likelihood Estimation and Face Detection** |
| **Josh Gleason and Rod Pickens** |
| **Programming Assignment 2** |
| **Computer Science 679 – Pattern Recognition, UNR, Dr. Bebis** |
| **Due: Monday March 23, 2015** |
| **Submitted: Monday March 23, 2015** |

Contents

[1 Abstract 3](#_Toc414055239)

[2 Technical Discussion 3](#_Toc414055240)

[2.1.1 Maximum Likelihood Estimation 3](#_Toc414055241)

[2.1.2 Classification 5](#_Toc414055242)

[One class classifier 6](#_Toc414055243)

[Two class classifier 7](#_Toc414055244)

[Comparison between 7](#_Toc414055245)

[2.1.3 Face Detection 7](#_Toc414055246)

[3 Project: Problems 1, 2, and 3 7](#_Toc414055247)

[3.1 Problem 1 8](#_Toc414055248)

[3.2 Problem 2 9](#_Toc414055249)

[3.3 Problem 3 10](#_Toc414055250)

[4 Results 12](#_Toc414055251)

[4.1 Maximum Likelihood Estimation 12](#_Toc414055252)

[4.1.1 Problem 1a 12](#_Toc414055253)

[4.1.2 Problem 1b 13](#_Toc414055254)

[4.1.3 Problem 2a 15](#_Toc414055255)

[4.1.4 Problem 2b 17](#_Toc414055256)

[4.2 Face Detection 19](#_Toc414055257)

[4.2.1 Problem 3a (Chromatic Color Space) 19](#_Toc414055258)

[4.2.2 Problem 3b (YCbCr Color Space) 19](#_Toc414055259)

[4.2.3 Problem 3c (RGB Color Space) 19](#_Toc414055260)

[4.2.4 Conclusion 19](#_Toc414055261)

[5 Conclusion 20](#_Toc414055262)

[6 Contributors 20](#_Toc414055263)

**Bayesian Classification of Classes with Features that are Normally Distributed**

# Abstract

This paper describes our research regarding the first class project for the Computer Science (CS) pattern recognition class CS 679 taught by Dr. Bebis Department Chair of the Computer Science Department at the University of Nevada in Reno, Nevada.

The topics included in this main body of this paper are a description of the project, the classifiers used in the project, and the results of the classification. The topic discussed in the appendix are a description of the theory of a Bayesian classifier and the derivation of the two classifiers used in the project, the theoretical errors expected from a minimum risk Bayesian classifier with a zero-one loss function, as well as the generation of normal Gaussian random variables and the evaluation of the Gaussian random numbers using a Goodness of Fitness test.

# Technical Discussion

## Maximum Likelihood Estimation

Bayesian Classifiers generally require knowledge of the both the likelihood and prior probability for all classes. In the previous project, the likelihood and prior probabilities were given. In this project we are assuming that the samples come from unknown distributions. Maximum Likelihood (ML) is one method of approximating the likelihood functions given that the parametric form of the distributions is known. Equation (1) describes the parametric form of the likelihood function with representing parameter number for class

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

For simplicity, the parameters will be manipulated in vector form

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

In order to find parameters which most accurately describe the likelihood function, training data is used. One set of training data is used for each class and that training data consists of samples independently drawn from the class. In other words, the training data used to determine the likelihood function for class are samples drawn from a random variable with probability density function . The training data for class is represented as follows

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

The premise of ML is to determine the parameters of the likelihood function that would be most likely to produce the training data. This leads to the following definition for the parameters estimated by maximum likelihood.

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

Where is an estimate of the true parameters for the likelihood function.

Because the samples in the training data were drawn independently, the probability density function in the right hand side of Equation (4) may be simplified to

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

Because Equation (4) is an optimization, and probability density functions are always positive, we can take the expression in the maximization and apply a log function (or any other monotonically increasing function). By also substituting Equation (5), Equation (4) becomes

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

Notice the product becomes a sum, which is the purpose of taking the natural log. The logarithm makes dealing with the maximization problem easier in many cases. For example, when the model for the likelihood function is a Normal distribution.

In order to solve the Equation (6) we take the gradient of the optimization expression and find where it becomes zero.

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

where

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

where is what we will call the log likelihood.

Solving for all solutions to Equation (7) will give the position of all local maxima, local minima, and inflection points. It is important to test each solution as well as the bounds of the function to determine where the true global maxima lies. The value of which results in the global maxima is the ML solution.

### Multivariate Normal Distribution

One interesting case to consider for ML is when likelihood takes the form of a Multivariate Normal distribution with an unknown mean and unknown covariance .

The parameters then consist of the mean and covariance of the distribution.

To begin the derivation we substitute the equation for the Multivariate Normal distribution into Equation (7)

|  |  |  |
| --- | --- | --- |
|  |  | (9) |

Because is composed of and we express Equation (9) as the following two derivatives

|  |  |  |
| --- | --- | --- |
|  |  | (10) |
|  |  | (11) |

Equation (10) may be solved as follows

|  |  |  |
| --- | --- | --- |
|  |  | (12) |

Similarly, Equation (11) may be solved as follows (substituting for )

|  |  |  |
| --- | --- | --- |
|  |  | (13) |

Together, Equations (12) and (13) express the ML estimates for the parameters of the Multivariate Normal distribution as

|  |  |  |
| --- | --- | --- |
|  |  | (14) |
|  |  | (15) |

which are the estimates that will be tested in Project 2

### One class classifier

### Two class classifier

### Comparison between

## Face Detection

### Features

### Classifiers

#### One class

#### Two class

# Project: Problems 1, 2, and 3

The project is to design a minimum error Bayesian classifier to assign a collected sample from a population that contains two classes, where = 1 or 2, to one or the other classes within the population. The project states that the features to be used have a Gaussian distribution with known mean and known covariance matrix. Since the covariance matrices are defined as diagonal and the features are defined as a joint distribution, it is known that the features are independent.[[1]](#footnote-3) The a-priori probabilities of each class are likewise specified.

The project does not explicitly specify costs for correct and incorrect decisions, so we do not build costs into the classifier. However, the costs can be inferred as coming from a zero-one loss function, or symmetric loss function, because the assignment states that the classifier be minimum error.[[2]](#footnote-4) The derivation of the minimum risk and minimum error classifiers are discussed in the appendix Sections **Error! Reference source not found.** and **Error! Reference source not found.**.

The classification is performed using two measurements, or features, from each sample. The samples are to be generated using random number generators, and the approach we used to generate the features is described in the Appendix.

The project asks that 10,000 samples be generated for each class and . The features for each class have 2D joint probability distributions.

Problem 1 is to run the following experiment twice with different a-priori class probabilities:

1. Design Bayes classifier for minimum error.
2. Plot the Bayes decision boundary together with the generated samples.
3. Classify the samples by the classifier and count the number of misclassified samples.
4. Plot the Chernoff bound as a function of β and find the optimum β for the minimum.
5. Calculate the Bhattacharyya bound.

Problem 2 is to again repeat the above steps, but with different feature density functions. These problems are discussed below.

## Problem 1

For Problem 1a, the a-priori class probabilities should be set such that , and for Problem 1b, the a-priori class probabilities should be set such that and

The normal probability distribution for the features is defined as follows:

|  |  |  |
| --- | --- | --- |
|  |  | (16) |

And normal probability distribution for the features is defined as follows

|  |  |  |
| --- | --- | --- |
|  |  | (17) |

Note that this description of features corresponds to designing a classifier that corresponds to Case 1[[3]](#footnote-5) in Duda, Hart, and Stork where . The cases are discussed further in Section **Error! Reference source not found.**.

The classifier chosen for Problem 1a is a dichotomizer because it is defined for a two-class decision problem in which the features are uncorrelated and have identical covariance matrices for both classes.[[4]](#footnote-6) The dichotomizer is defined as the following:

|  |  |  |
| --- | --- | --- |
|  | = 0 | (18) |

Because the features are independent, normally distributed and have identical covariance matrices for each class, the dichotomizer reduces to the following discriminant function

|  |  |  |
| --- | --- | --- |
|  |  | (19) |

Where

|  |  |  |
| --- | --- | --- |
|  |  | (20) |

And

|  |  |  |
| --- | --- | --- |
|  |  | (21) |

## Problem 2

Problem 2 is to estimate the parameters of the distributions and then rerun the above experiment except that the probability distribution for the features changes. The features for are defined as follows

|  |  |  |
| --- | --- | --- |
|  |  | (22) |

And the features for are defined as follows

|  |  |  |
| --- | --- | --- |
|  |  | (23) |

Since the covariance matrix for each class is different, the classifier used in the Problem 2 is based on the classifier defined in Duda, Hart, and Stork in Section 2.6.3 where Given these facts, the discriminant function chosen for each class is defined as follows:

|  |  |  |
| --- | --- | --- |
|  |  | (24) |

Where

|  |  |  |
| --- | --- | --- |
|  |  | (25) |

and

|  |  |  |
| --- | --- | --- |
|  |  | (26) |

and

|  |  |  |
| --- | --- | --- |
|  |  | (27) |

For Problem 2a, the a-priori class probabilities should be set such that , and for Problem 2b, the a-priori class probabilities should be set such that and The a-priori probabilities are the same as in Problem 1.

## Problem 3

The project is to develop a classifier that performs face detection in 2D visible red, green, and blue imagery based on the skin color of the people in the 2D imagery. We design two types of classifier for this classification: one class and two class. For designing and analyzing the one class classifier, we based our classifier on the Mahalanobis distance[[5]](#footnote-8) . For the designing and analyzing the two class classifier, we based our classifier on the Bayes dichotomizer.[[6]](#footnote-9)

# Results

## Maximum Likelihood Estimation

### Problem 1a

For Problem 1a, we generate a plot of the ten-thousand samples from two classes along with the decision boundary that separates the two classes. Note that the distributions for the features of each class are defined as follows:

|  |  |  |
| --- | --- | --- |
|  |  | (28) |
|  |  | (29) |

The a-priori probabilities are as follows

|  |  |  |
| --- | --- | --- |
|  |  | (30) |

In Figure 1, the two classes are colored differently: class 1 (red dots) is and class 2 (blue dots) is The decision boundary between the two classes is clearly an affine function (a linear function with an additive intercept). The Bayes error is 1.76 percent.

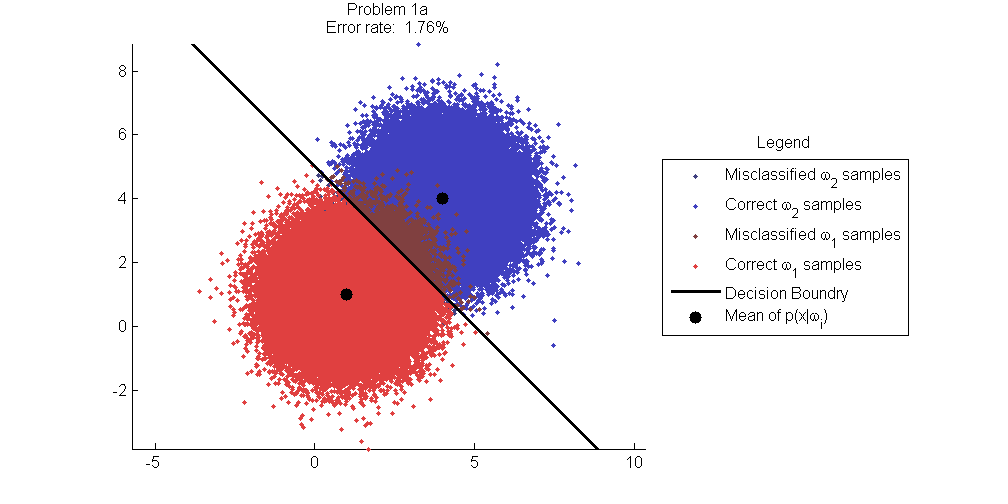


Figure : Classification results for Problem 1a.

The Bayes error (notes as true error in the Figure) and the Chernoff and Bhattacharyya error bounds for the classification are given in Figure 2. The true error at 1.69% is significantly lower than are either the Chernoff and Bhattacharyya error bounds, both at 5.27%.

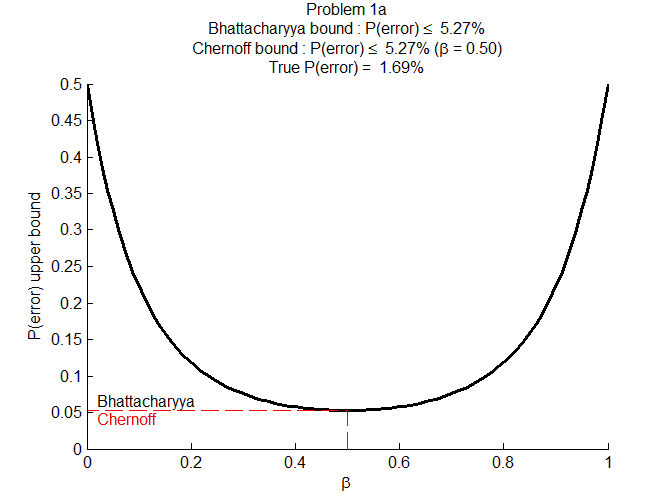


Figure : Chernoff and Bhattacharyya bounds for Problem 1a.

### Problem 1b

For Problem 1b, we again generate a plot of the ten-thousand samples from two classes along with the decision boundary that separates the two classes. Note that the distributions for the features of each class are defined as follows:

|  |  |  |
| --- | --- | --- |
|  |  | (31) |
|  |  | (32) |

The a-priori probabilities are different and are set as follows

|  |  |  |
| --- | --- | --- |
|  | and | (33) |

In Figure 3, the two classes are colored differently: class 1 (red dots) is and class 2 (blue dots) is Note that the distribution of the samples are identical to the samples in Problem a, which they should identically resemble unless a different random seed was used which was not. The decision boundary between the two classes is again clearly an affine function (a linear function with an additive intercept). The Bayes error is 1.56%. This improvement over Problem 1a is due to the fact that class 2 is more likely than class 1 .

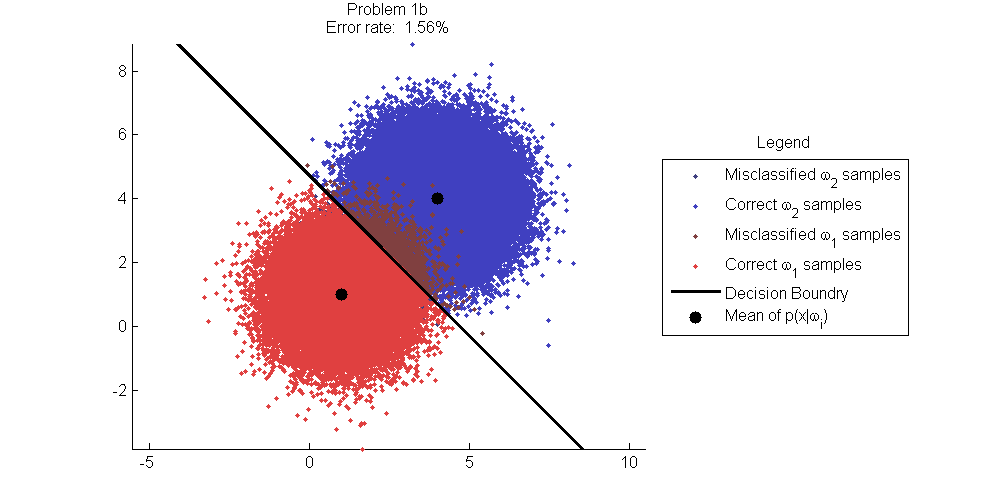


Figure : Classification results Problem 1b.

The Bayes error (labeled as true error) and the Chernoff and Bhattacharyya error bounds for the classification are given in Figure 4. The true error is 1.53% and is again significantly lower than are either the Chernoff and Bhattacharyya error bounds; the Chernoff at 4.73% and the Bhattacharyya at 4.83%. This difference in error bounds matches our expectations in that Bayes error Chernoff error Bhattacharyya.

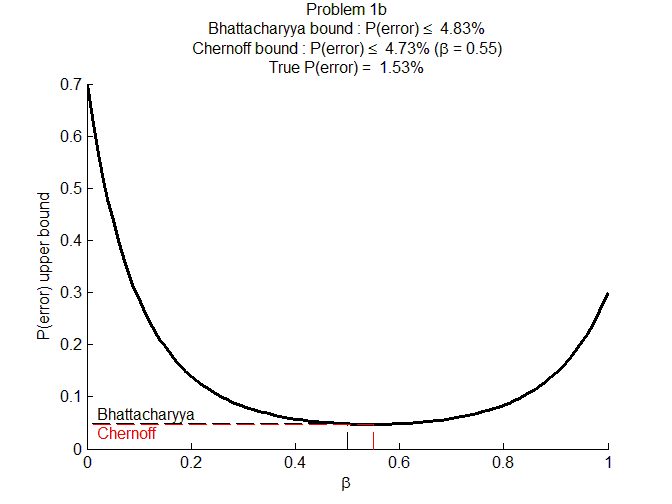


Figure : Chernoff and Bhattacharyya bounds for Problem 1b.

### Problem 2a

For Problem 2a, we again generate a plot of the ten-thousand samples from two classes along with the decision boundary that separates the two classes. Note that the distributions for the features of each class are defined as follows:

|  |  |  |
| --- | --- | --- |
|  |  | (34) |
|  |  | (35) |

The a-priori probabilities are as follows

|  |  |  |
| --- | --- | --- |
|  |  | (36) |

In Figure 5, the two classes are colored differently: class 1 (red dots) is and class 2 (blue dots) is Note that the distribution of the samples are identical to the samples in Problem 1a, which they should identically resemble unless a different random seed was used which was not. The decision boundary between the two classes is clearly an ellipse. The Bayes error is 6.71 percent.

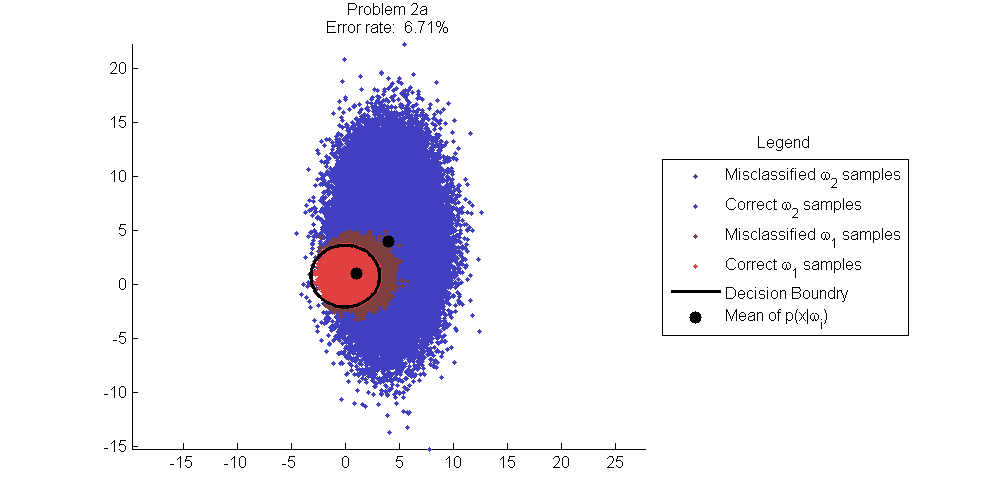


Figure : Classification results Problem 2a.

The Bayes error (labeled as true error) and the Chernoff and Bhattacharyya error bounds for the classification are given in Figure 6. The true error is 6.76% and is again significantly lower than are either the Chernoff and Bhattacharyya error bounds; the Chernoff at 14.78% and the Bhattacharyya at 17.14%. This difference in error bounds matches our expectations in that Bayes error Chernoff error Bhattacharyya. Also, there is considerable overlap in the feature spaces between the two classes, and we would expect larger errors than observed in Problems 1a and 1b.

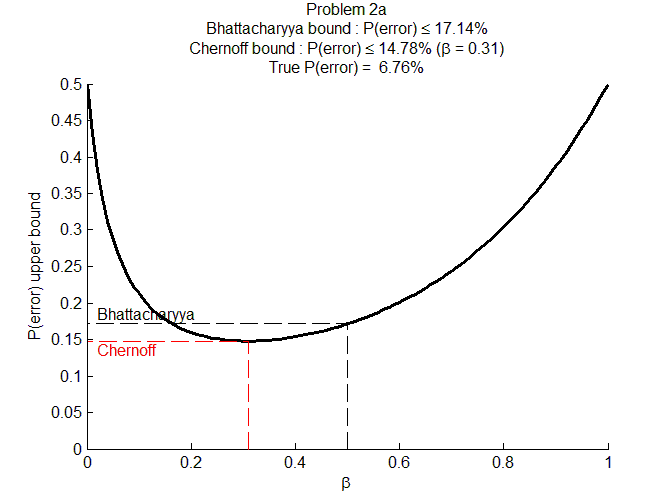


Figure : Chernoff and Bhattacharyya bounds for Problem 2a.

### Problem 2b

For Problem 2b, we again generate a plot of the ten-thousand samples from two classes along with the decision boundary that separates the two classes. Note that the distributions for the features of each class are defined as follows:

|  |  |  |
| --- | --- | --- |
|  |  | (37) |
|  |  | (38) |

The a-priori probabilities are as follows

|  |  |  |
| --- | --- | --- |
|  |  | () |

In Figure 7, the two classes are colored differently: class 1 (red dots) is and class 2 (blue dots) is Note that the distribution of the samples are identical to the samples in Problem 1a, which they should identically resemble unless a different random seed was used which was not. The decision boundary between the two classes is clearly an ellipse. The Bayes error is 7.35%.

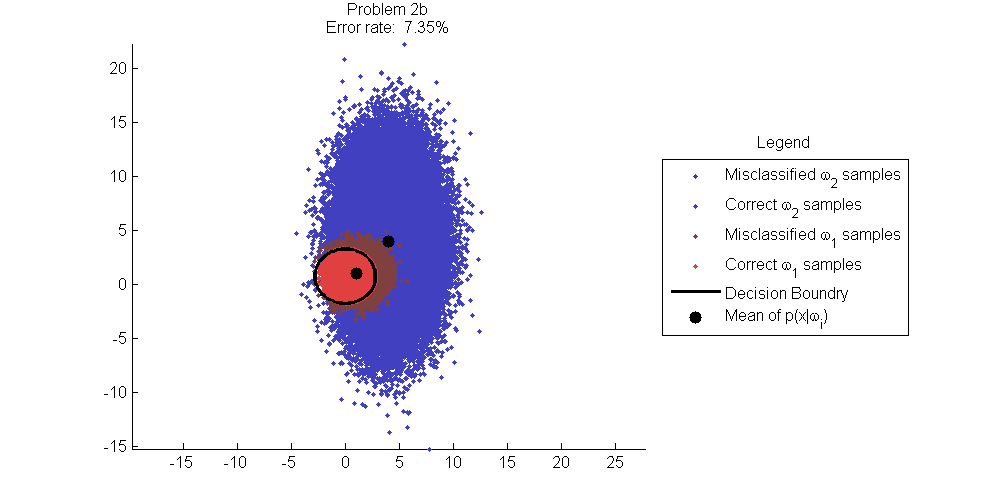


Figure : Classification results Problem 2b.

The Bayes error (plotted as the true error in the Figure) and the Chernoff and Bhattacharyya error bounds for the classification are given in Figure 8. The true error is 6.76% and is again significantly lower than are either the Chernoff and Bhattacharyya error bounds; the Chernoff at 15.31% and the Bhattacharyya at 15.71%. This difference in error bounds matches our expectations in that Bayes error Chernoff error Bhattacharyya. Also, there is considerable overlap in the feature spaces between the two classes, and we would expect larger errors than observed in problem 1a and 1b. As observed early, overall errors are smaller than in problem 2a because class 2 is favored probabilistically over class 1 (the a-priori probabilities differ).

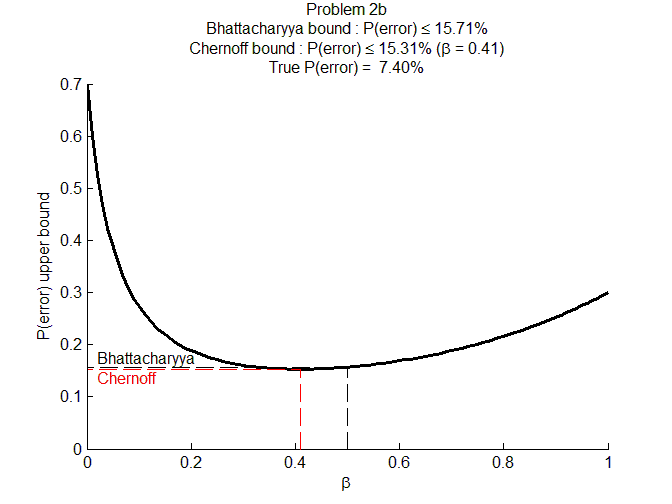


Figure : Chernoff and Bhattacharyya bounds Problem 2b.

## Face Detection

### Problem 3a (Chromatic Color Space)

One class

Two class

### Problem 3b (YCbCr Color Space)

One class

Two class

### Problem 3c (RGB Color Space)

One class

Two class

### Conclusion

# Conclusion

The Bayesian classifier for all test cases in this project produces results in which the error is minimized over all classifiers. This minimization occurs because the classifier was designed as a minimum risk classifier using a zero-one loss function.[[7]](#footnote-10) The classifier is known as a Bayesian classifier because the decision rule is based on estimating a-posteriori class probabilities using the Bayes rule that

|  |  |  |
| --- | --- | --- |
|  |  | (39) |

The resulting classifier is both a Bayes minimum risk classifier and minimum probability of error classifier.[[8]](#footnote-11)

# Contributors

Josh Gleason and Rod Pickens each wrote their own MATLAB software to perform the classification, the maximum likelihood estimation, the error estimation, the calculation of the Bayes error, the Chernoff error bound, and the Bhattacharyya error bound.

Josh generated the maximum likelihood performance charts, and Rod generated the face detection performance charts.

Josh wrote the theory section on maximum likelihood along with the results section, and as an added feature, Josh wrote the section on one class classification using as reference Fukunaga[[9]](#footnote-12), as suggested by Rod. Rod Pickens wrote the theory section on face detection along with the results section, and Rod also read and discussed with Josh the theory of one class classification.

1. http://en.wikipedia.org/wiki/Normally\_distributed\_and\_uncorrelated\_does\_not\_imply\_independent [↑](#footnote-ref-3)
2. Duda, Richard O., Hart, Peter O. and Stork, David G, “Pattern Classification,” Wiley Interscience, Second Edition, page 26, equation 19. [↑](#footnote-ref-4)
3. Ibid, page 36, Section 2.6.1. [↑](#footnote-ref-5)
4. Ibid, page 41, Section 2.6.3. [↑](#footnote-ref-6)
5. Keinosuke Fukunaga, Raymond Hayes, and Leslie Novak, “The Acquisition Probability for a Minimum Distance One-Class Classifier,” IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-23, No. 4, July 1987, pages 493-499. [↑](#footnote-ref-8)
6. Richard O. Duda, Peter O. Hart, and David G. Stork, “Pattern Classification,” Wiley Interscience, Second Edition, page 31, equation 29. [↑](#footnote-ref-9)
7. Ibid., page 26, Equation 19. [↑](#footnote-ref-10)
8. Ibid., page 27, Equation 21. [↑](#footnote-ref-11)
9. Keinosuke Fukunaga, Raymond Hayes, and Leslie Novak, “The Acquisition Probability for a Minimum Distance One-Class Classifier,” IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-23, No. 4, July 1987, pages 493-499. [↑](#footnote-ref-12)