Abstract

This report covers the design and implementation of a set of Baysian decision rules in order to perform pattern classification. Both likelihood ratio and discriminant function methods are implemented. A two-category synthetic data set is used to evaluate the performance of the classifiers. The data set is initially assumed to have a Gaussian distribution. Two algorithms are used to estimate the parameters of Gaussian, including maximum likelihood estimation and Bayes learning. Based on the behavior of the data set, a two-modal Gaussian distribution is also used. The performance of each classification method using different distribution functions with different parameter estimation algorithms was evaluated and compared. It was found that by carefully choosing the loss ratio and the distribution of the data set, we can obtain an accuracy rate as high as 92%.

1. Introduction

Is this an apple or an orange? The way we as humans observe the world makes this a trivial question. But with machines, the decision isn't as straightforward. The computer doesn't have the luxury of experience to quickly decide which type of fruit they are presented with. That is where pattern classification comes into play [9].

With pattern classification we are able to provide the computer with hard, concise numbers, and a decision algorithm to help it properly decide the classification of samples. The purpose of this project was to introduce us to some of the basic types of decision methods, namely the likelihood ratio, and the discriminant function [9].

Pattern Recognition is a varied and useful field [7]. There exist many real-world scenarios in which a decision must be made based on the weights of one or more inputs. For example, an automobile’s onboard computer may decide to illuminate the check engine light based upon the input values of electronic sensors measuring features such as temperature and pressure. Similarly, an image segmentation algorithm may classify pixels as part of an object or background based upon the pixels’ intensity and location. In any of these examples, the user wants to classify something with minimal error [1].

Bayes Decision Theory is a supervised classification method based on Bayes Formula for maximum posteriori probability [7]. It has been widely used in pattern classification. Classification using a decision rule often involves using inputs, probability distributions, and some prior knowledge to decide the most likely, correct output classification. One can calculate the posterior probability that an event occurred by taking the product of that event’s likelihood with its prior probability and dividing by some given evidence [2]. This expression is known as Bayes’ formula:

\[
P(w_j | x) = \frac{p(x | w_j) \cdot P(w_j)}{p(x)},
\]
where \( p(x|w_j) \) is the likelihood of \( w_j \) with respect to \( x \), \( P(w_j) \) is the prior probability of \( w_j \), \( p(x) \) is the given evidence, and \( P(w_j|x) \) is the posteriori probability that the event \( w_j \) occurred. In a situation in which there are only two categories, Bayes’ decision rule for minimizing the probability of error concludes that event \( w_j \) occurred if \( P(w_j|x) \) is greater than \( P(w_j|x) \). Otherwise, event \( w_j \) occurred [1].

In Bayes’ formula, an appropriate estimation of the distribution function is the key to the success of any classifiers. In many cases, it is reasonable to assume it to be Gaussian. In order to estimate the parameters of Gaussian, both Maximum likelihood and Bayesian learning methods can be used.

The objective of this project was to design and implement a series of pattern classification rules in order to determine whether or not a given item belongs to a certain class. The experiment gave an opportunity to see how pattern recognition and classification could be put into practice from an understanding of the theory [1]. In this project, two variants of Bayesian classifiers are examined to classify a two-class data set, including maximum likelihood ratio and the discriminant function. In the discriminant function method, three cases are evaluated with different assumptions that simplify the calculation to different degrees. The data are assumed to have a normal distribution in each class. The parameters of Gaussian are estimated using maximum likelihood and Bayesien inference. The effect of different decision rules and parameter estimation methods are evaluated [5].

2. Technical Approach

2.1. Variants of Baysian Decision Rule

If considering the conditional risk of the decision, the Bayse’ formula can be extended to maximum likelihood ratio method. The risk \( R_1 \) associated with choosing \( w_1 \), and the risk \( R_2 \) associated with choosing \( w_2 \) is calculated as:

\[
R_1 = \lambda_{11}P(w_1 | x) + \lambda_{12}P(w_2 | x)
\]

\[
R_2 = \lambda_{21}P(w_1 | x) + \lambda_{22}P(w_2 | x)
\]

where \( \lambda_{ij} \) is the loss incurred for deciding \( w_j \) when the actual state is \( w_j \) [2]. Employing these conditional risks along with Bayes’ formula, one can calculate a likelihood ratio:

\[
\frac{p(x | w_1)}{p(x | w_2)} > \frac{\lambda_{12} - \lambda_{22}}{\lambda_{21} - \lambda_{11}} \cdot \frac{P(w_2)}{P(w_1)}
\]

If the above equality holds true, then it is decided that \( w_j \) occurred; otherwise, \( w_2 \) is believed to have occurred.

Discriminant functions provide another form of representation for classifying patterns. A commonly used discriminant is given below:
\[ g_i(x) = \ln p(x \mid w_i) + \ln P(w_i). \]

A two-category classifier decides that event \( w_i \) occurred if \( g_i \) is greater than \( g_2 \), and visa-versa. If the probability density functions \( p(x \mid w_i) \) are normal Gaussian distributions, then the discriminant can be evaluated for three different cases:

**Case I:** The features, or inputs to the decision-making algorithm, are statistically independent and have the same variance \( \Sigma^2 \). If the probability of the outcomes \( P(w_i) \) are all the same for every class, then the discriminant functions are simply a measure of the Euclidean distance between an input \( x \) and the mean of each class.

**Case II:** The covariance matrices for all of the classes are identical, but the entries of the covariance matrices are arbitrary \( \Sigma^2 \). This allows the discriminant to be calculated as:

\[ g_i(x) = \frac{1}{2} (x - \bar{\mu}_i)\Sigma^{-1}(x - \bar{\mu}_i) + \ln P(w_i), \quad (5) \]

where \( \mu_i \) is the mean of class \( i \), and \( \Sigma^{-1} \) is the inverse of the covariance matrix. The first term of Eq. 5 represents the Mahalanobis distance.

**Case III:** The covariance matrices are different for each category \( \Sigma^2 \). Thus, the discriminant function becomes:

\[ g_i(x) = \frac{1}{2} (x - \bar{\mu}_i)^\top \Sigma^{-1}(x - \bar{\mu}_i) - \frac{1}{2} \ln |\Sigma| + \ln P(w_i). \quad (6) \]

### 2.2. Parameter Estimation

The parameters \( \bar{\mu} \) and \( \Sigma \) in the pdfs were first derived using Maximum Likelihood Estimation. This method involves finding the maximizing value of \( \theta = [\theta_1 \ \theta_2] = [\bar{\mu} \ \Sigma] \) in the function

\[ l(\hat{\theta}) = \sum_{k=1}^{n} \left( -\frac{1}{2} \ln(2\pi) - \frac{1}{2} \ln \theta_2 - \frac{1}{2}(\bar{x}_k - \bar{\theta}_1)^\top \theta_2^{-1} (\bar{x}_k - \bar{\theta}_1) \right). \]

The calculated parameters are substituted directly into the two-dimensional Gaussian pdf [4].

Baysian parameter estimation includes our prior knowledge about the data distribution into the parameter estimation. The more accurate we know about the parameter beforehand, the better estimation we can make than using Maximum likelihood method. The Baysian estimation assumes that the estimated parameters are random variables with a normal distribution. Thus, the estimated training set distribution will be
\[ P(x / D) = \int p(x / \mu)p(\mu / D) d\mu \sim N(\mu_n, \delta^2 + \delta_n^2) \]

where:

\[
\mu_n = \left( \frac{n\delta_0^2}{n\delta_0^2 + \delta^2} \right) \left( \frac{1}{n} \sum X_i \right) + \left( \frac{\delta^2}{n\delta_0^2 + \delta^2} \right) \mu_0
\]

\[
\delta_n^2 = \left( \frac{\delta_0^2\delta^2}{n\delta_0^2 + \delta^2} \right)
\]

In this equation, the \( \delta_0^2 \) and \( \mu_0 \) are our prior knowledge about the data distribution. The effects of \( \delta_0^2 \) and \( \mu_0 \) on the pattern classification really depend on the accuracy and the value of \( \delta_0^2 \). When \( \delta_0^2 \) is very large, the Bayesian estimation produces similar results as MLE. When \( \delta_0^2 \) is small which infers that \( \mu_0 \) is accurate, the Bayesian estimation can improve the classification results [5].

### 3. Experiments and Results

#### 3.1. The Training Set and the Test Set [3]

Both the training set (synth.tr) and the testing set (synth.te) used in this project are obtained from the dataset provided in [8]. The data sets includes samples of two real-valued coordinates (xs and ys) and a class (xc) which is 0 or 1. The training set contains 125 samples and the testing set contains 500 samples. Figure 1 shows the sample distribution in the two data sets.

![Sample distribution of the training and testing sets](image)

Figure 1. The training set and the test set.

The training set is used to estimate the parameters of the 2D Gaussian distribution. Figure 2 illustrates the derived distribution.
(a) The data distribution using Gaussian (b) the training sets over the top view of (a)

![Image of data distribution and training sets](image)

**Figure 2.** The estimated distribution of the training sets.

### 3.2. Use Two-Modal Gaussian as Distribution Function [3]

By observing the training set and the testing set, we can see that even though the samples are divided into two classes, each class is still composed of two clusters. In another word, a one-modal Gaussian might not be an appropriate estimation of the data distributions. The variance along each direction also supports this observation. The variances of class 0 are $\sigma_x^2 = 0.274595$ and $\sigma_y^2 = 0.0358301$, and the variances of class 1 are $\sigma_x^2 = 0.15847$ and $\sigma_y^2 = 0.0297187$. The variances along the x direction are much larger than that along the y direction. Based on these observations, a two-modal Gaussian is then used to estimate the distribution. We first divide the samples in the same class into two regions by the line $x = a$. The distance between the line and the data sets were evaluated and the smallest points, $x = -0.14$ for the class 0, $x = 0.07$ for the class 1 are used. After we get these points, two different 2D Gaussians can then be derived and combined.

![Diagram showing the absolute distance between sample of the class 0 and the vertical line (x=a)](image)

**Figure 3.** The absolute distance between the $x = a$ for $-1.2 \leq a \leq 1.1$ and the training sets.
The new distribution uses the two Gaussians for one class, so totally four Gaussians are used for representing the two classes as shown in Figure 4.

(a) The data distribution using Gaussian (b) the training sets over the top view of (a)

Figure 4. The estimated distribution using two-modal Gaussian for each class.

### 3.3. Comparison of Different Classification Methods [3]

The performance of different classification algorithms with different parameter estimations is evaluated. The accuracy is shown in Table 1. Totally eight experiments have been performed. For the case 2, two different covariance matrices are used and evaluated separately. Different combinations of the risks, $\lambda_{01}$ and $\lambda_{10}$, are also evaluated for the Maximum likelihood method. The final task using mixture Gaussian distribution uses two 2D Gaussian models for each class.

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameters</th>
<th>Accuracy for the class 0</th>
<th>Accuracy for the class 1</th>
<th>Overall accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>-</td>
<td>0.68</td>
<td>0.746</td>
<td>0.713</td>
</tr>
<tr>
<td>Case 2</td>
<td>Use the covariance matrix of the class 0</td>
<td>0.896</td>
<td>0.874</td>
<td>0.885</td>
</tr>
<tr>
<td></td>
<td>Use the covariance matrix of the class 1</td>
<td>0.886</td>
<td>0.878</td>
<td>0.882</td>
</tr>
<tr>
<td>Case 3</td>
<td>-</td>
<td>0.908</td>
<td>0.888</td>
<td>0.898</td>
</tr>
<tr>
<td>ML</td>
<td>$\lambda_{01} = 1.0$, $\lambda_{10} = 1.0$</td>
<td>0.908</td>
<td>0.888</td>
<td>0.898</td>
</tr>
<tr>
<td></td>
<td>$\lambda_{01} = 0.5$, $\lambda_{10} = 1.0$</td>
<td>0.956</td>
<td>0.794</td>
<td>0.875</td>
</tr>
<tr>
<td></td>
<td>$\lambda_{01} = 1.0$, $\lambda_{10} = 0.5$</td>
<td>0.834</td>
<td>0.928</td>
<td>0.881</td>
</tr>
<tr>
<td>Mixture Gaussian (Case 3)</td>
<td>2 Gaussian for one class</td>
<td>0.846</td>
<td>0.928</td>
<td>0.887</td>
</tr>
</tbody>
</table>

Table 1 The estimated results. 1.0 represents 100% accuracy. For the maximum likelihood, $\lambda_{11} = \lambda_{00} = 0$ is used for three different risk combinations.
Figure 5 (a) to (c) illustrate the decision regions derived by the three discriminant functions, (d) to (f) show the decision regions with different combinations of risk $\lambda_{01}$ and $\lambda_{10}$ for the maximum likelihood. When $\lambda_{01} = 0.5$ is smaller than $\lambda_{10} = 1.0$, the decision region for class 0 is expanded.

As shown in Table 1, the case 3 shows the best results. The reason the case 1 and case 2 have poor results than the case 3 is that the basic assumption for the class 1 and 2 was not satisfied. Three combination of the risks $\lambda_{01}$ and $\lambda_{10}$ were used for the maximum likelihood method. When $\lambda_{01} = 1.0$, $\lambda_{10} = 1.0$ and $\lambda_{11} = \lambda_{00} = 0$ were used, the result of the maximum likelihood was the same with the case 3. When smaller $\lambda_{01}$ was used, the
accuracy for the class 0 was increased, but the accuracy for the class 1 was decreased because the maximum likelihood minimized the overall risk. In other words, the maximum likelihood method enlarge the decision region 0 because the risk when assigns the class 0 to a sample that belongs to the class 1 is lower than $\lambda_{i0}$.

The performance of the mixture Gaussian, two-modal Gaussian, shows almost same performance with the case 3, but slightly lower. The decision regions, which are shown in Figure 5(c) and (g), shows that the mixture Gaussian gives more reasonable segment than the case 3, but the accuracy for the class 0 was not good.

### 3.4. Comparison of Different Classification Methods [7]

The experiment proceeded as such:

1. Code was written in C++ that would use the data contained in the file synth.tr to train a decision rule, and use synth.te to test that decision rule.

2. Maximum likelihood estimation was used to estimate the parameters of the Gaussian: the mean vector and the covariance matrix.

3. Both likelihood ratio and 3 discriminant functions were used to design a decision rule and the performance was compared.

The classifications from each decision rule were plotted with each feature as an axis. The correctly identified items were marked based on type.
Discriminant Case 1: 71.3% Correct
Discriminant Case 2: 89.1% Correct
Discriminant Case 3: 89.8% Correct

Of the four methods for classifying features, Maximum Likelihood and Discriminant Case 2 produced equal results at 89.1% correct. Discriminant Case 1 yielded the worst results at 71.3% correct and Discriminant Case 3 yielded best results of 89.8% correct. The gradually increasing performance of the discriminant cases was expected since each of the rules gave a gradually more sophisticated boundary between the training features and the actual patterns to be classified.

3.5. Comparison of Different Classification Methods [5]

It is very clear there are two clusters in each group. Therefore, using one Gaussian function to summarize the data in each group will bring an extra error. Suppose we describe the two clusters in a group separately and then add them together, the classification error should be decreased. With the help of Fuzzy c_cluster and MLE technique, we can estimate the parameters for the clusters in each group and summarize as below. These parameters will be utilized in two-modal Gaussian classification, the results are shown in the table.

Table 2: Tow modal Gaussian function in order to improve the accuracy of classification

<table>
<thead>
<tr>
<th>Group 1</th>
<th>Group 2</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster 1</td>
<td>Cluster 2</td>
<td>Cluster 1</td>
</tr>
<tr>
<td>( \mu )</td>
<td>( \delta^2 )</td>
<td>( \delta^n^2 )</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
</tbody>
</table>
| \[
\begin{bmatrix}
0.4374 \\
0.6351
\end{bmatrix}
\] | \[
\begin{bmatrix}
0.0286.0038 \\
0.0038.03158
\end{bmatrix}
\] | \[
\begin{bmatrix}
0.00035 \\
0.000111
\end{bmatrix}
\] | 92 | 9.2% |
| \[
\begin{bmatrix}
-0.2913 \\
0.7316
\end{bmatrix}
\] | \[
\begin{bmatrix}
0.0228.0004 \\
0.0004.0232
\end{bmatrix}
\] | \[
\begin{bmatrix}
0.000111 \\
0.00035
\end{bmatrix}
\] | | |
| \[
\begin{bmatrix}
0.2705 \\
0.3500
\end{bmatrix}
\] | \[
\begin{bmatrix}
0.0313 - 0.0042 \\
-0.0042 0.0240
\end{bmatrix}
\] | \[
\begin{bmatrix}
0.000111 \\
0.00035
\end{bmatrix}
\] | | |
| \[
\begin{bmatrix}
-0.7214 \\
0.3012
\end{bmatrix}
\] | \[
\begin{bmatrix}
0.0260 .0023 \\
0.0230 .0467
\end{bmatrix}
\] | \[
\begin{bmatrix}
0.000111 \\
0.00035
\end{bmatrix}
\] | | |

Table 3: Baysien estimation for classification, with different \( \delta_0^2 \) and \( \mu_0 \)

<table>
<thead>
<tr>
<th>( \delta_0^2 ) and ( \mu_0 )</th>
<th>( \delta_n^2 ) and ( \mu_n )</th>
<th>Number of misclassification</th>
<th>Error rate</th>
</tr>
</thead>
</table>
| Group 1: \( \delta_0^2 = 
\begin{bmatrix}
0.002746 & 0.000111 \\
0.000111 & 0.00035
\end{bmatrix}
\) | Group 1: \( \delta_n^2 =
\begin{bmatrix}
0.00122043 & 4.94282e - 05 \\
4.94282e - 05 & 0.000157584
\end{bmatrix}
\) | 100 | 10.0% |
| \( \mu_0 =
\begin{bmatrix}
-0.2050 \\
0.2960
\end{bmatrix}
\) | \( \mu_n =
\begin{bmatrix}
-0.214159 \\
0.312353
\end{bmatrix}
\) | | |
| Group 2: \( \delta_0^2 = 
\begin{bmatrix}
0.002746 & 0.000111 \\
0.000111 & 0.000358
\end{bmatrix}
\) | Group 2: \( \delta_n^2 =
\begin{bmatrix}
0.000843312 & -3.6899e - 05 \\
-3.6899e - 05 & 0.00139366
\end{bmatrix}
\) | | |
| \( \mu_0 =
\begin{bmatrix}
0.0567 \\
0.6932
\end{bmatrix}
\) | \( \mu_n =
\begin{bmatrix}
0.0678304 \\
0.687614
\end{bmatrix}
\) | | |
3.6. **Other Comparisons [4]**

![Accuracy vs. lambda values](image1)

Figure 6. Accuracy vs. lambda values.

![Accuracy vs. disc. function](image2)

Figure 7. Accuracy vs. disc. function

The highest accuracy rate from the Maximum Likelihood classification method was obtained when $\lambda_{21} = \lambda_{12}$. This equality simply means that there is equal risk in misclassifying a sample in either category. Classification accuracy decreases for these data sets when loss is biased in either direction. When a loss of zero is assigned to either $\lambda_{21}$ or $\lambda_{12}$, all of the samples are classified into one category, which results in a very low 50% accuracy rating.

Of the three discriminant functions, the Case 3 function performed the best. The greatest drawback of the Case 3 function is its complexity and thus the time needed to compute its value. In only two dimensions, the time needed for a modern computer to compute the Case 3 discriminant function value is negligible. However, in systems with more features, the complexity of the function may be of greater concern.

The two different types of decision rules performed equally well in their best cases. The highly simplified Case 1 discriminant function performed relatively poorly; however, the somewhat simplified Case 2 function performed very much like the Case 3 function. Although a 90% accuracy rate might be acceptable for some applications, it might be
quite unacceptable for an application such as detecting cancer. In many medical applications, biased loss values might need to be assigned so that safety takes precedence over accuracy.

3.7. Other Comparisons [6]

The accuracy of case 3 was seen to be the highest. This makes sense because it is simply based on the Guassian distribution of each type without any imposed losses or false assumptions taken into account. Next, the likelihood ratio shows an accuracy almost as high as case 3. Since the loss ratio used is so close to unity and likelihood ratio and case 3 are identical for a loss of unity, this also makes sense. Case 2 shows an accuracy that is lower by only 1% to 2%. Apparently, the covariance matrices for each of the two types are approximately equal, and the assumption that they are equal does not affect the results to a large degree. Case 1 shows the poorest results. The assumption that the variances of each type in the training set are equal is not an accurate assumption. In general, the assumptions made in cases 1 and 2 should not be made unless they are truly the case.

4. Summary

A two-category classification algorithm has been designed using likelihood ratios and discriminant functions. Evaluating the performance of each method suggests that using the Case-III discriminant function is the best decision rule for these given data sets because it always provided the lowest error rate. It is important to note that the method with the lowest error rate is not always the most preferable, since over-training of the data may result. However, the Case-III discriminant function also offers another advantage in that it works independently of the arbitrarily chosen losses $\lambda_{ij}$[1].
The objectives of this project were met with no unreasonable difficulty. The methods used to develop an appropriate decision rule have been more fully understood by myself, and they are ready to be employed in future course projects. These methods could be used in a myriad of classification problems. Speech recognition, image recognition, and medical diagnosis are but a few. These techniques could be extremely useful to any scientist, engineer, or mathematician wishing to bring order to some apparently random data. This project also served as useful programming practice and helped to better familiarize myself with the LINUX operating system. The method of Bayesian parameter estimation is perhaps a better method of estimation than maximum likelihood. Bayesian estimation was not explored in this project due to time constraints; however, in the future this method should be employed since it allows the parameters to learn as each test sample is evaluated [6].

The development of the code necessary to implement the various decision rules was not an overly difficult process. The data acquisition and matrix operations were greatly simplified by the use of the provided libraries. The techniques required to realize the complex mathematical formulas provided good practice for creating sophisticated statistical applications. Given more time several improvements could be made to the decision rule algorithm. Bayesian parameter estimation could be used to find the component mean vector and covariance matrix from the training set. The use of a two-modal Gaussian to measure the data set would probably increase performance due to the distribution of data into two clusters. Also, the code could be expanded to work with a wider range of features and types. There are several real-world examples where the code developed in this project might be able to be used help make decisions. People could be classified as at risk for hypertension based on weight and age. It could be used politically to predict candidate support based on geographic location and age. Students could be grouped as engineering or non-engineering based on amount of homework and hours of sleep at night [7].

5. Reference