CLASSIFICATION OF IMAGE PIXELS BASED ON MINIMUM DISTANCE AND HYPOTHESIS TESTING

by

SANTOSH GHIMIRE

B.Sc., Tribhuvan University, Nepal, 1999
M.Sc., Tribhuvan University, Nepal, 2001
M.S., Kansas State University, 2008

A REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Statistics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas
2011

Approved by:

Major Professor
Prof. Haiyan Wang

Abstract

We introduce a new classification method that is applicable to classify image pixels. This work was motivated by the test-based classification (TBC) introduced by Liao and Akritas (2007). We found that direct application of TBC on image pixel classification can lead to high mis-classcification rate. We propose a method that combines the minimum distance and evidence from hypothesis testing to classify image pixels. The method is implemented in R programming language. Our method eliminates the drawback of Liao and Akritas (2007). Extensive experiments show that our modified method works better in the classification of image pixels in comparison with some standard methods of classification; namely, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Classification Tree (CT), Polyclass classification, and TBC. We demonstrate that our method works well in the case of both grayscale and color images.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my major professor, Dr. Haiyan Wang for her enormous support, close guidance and encouragement throughout this project which helped me accomplish this report. I am also thankful to my mathematics advisor Dr. Charles N. Moore, for his motivation and encouragement for this work. I would like to express my appreciation to Dr. James Neill for his valuable inputs and feedback. Also my gratitude goes to Department of Mathematics, Kansas State University for supporting my study outside the department.

Finally, I am thankful to my mother and wife for their support and encouragement which made this achievement possible.

Dedication

To my father late Madhusudan Upadhyaya.

Chapter 1

Introduction

The aim of this report is to provide a detailed description of a new image classification method supplemented with examples of implementation. We use R programming language, for the implementation of our method. The codes for our method and other methods for comparison are also given in Appendix at the end of the report. We devote chapter 1 to introduce image analysis in general and give a brief review of other existing methods of image pixel classification which motivates our study and provide a background to describe our new method.

1.1 Image

We begin this section with the mathematical definition of an image. Mathematically, an image can be defined by a two dimensional function, say $f(x, y)$, where x and y represent plane coordinates and amplitude of f at (x, y) is called grey level or intensity of image at that point. When the values given by x, y and the amplitude of f are all finite, discrete quantities, the resulting image is called a digital image. These processes of digitizing the coordinates and amplitudes are termed as sampling and quantization respectively. Thus the process of sampling and quantization results in a representation of an image as a matrix of real numbers. More precisely, a rectangular black-and-white image is a matrix of real numbers in which all the entries represent the level of grey at that point. The level of grey ranges from 0 to 255 in which 0 represents the darkest spot and 255 represents the brightest spot. The elements of the digital image are usually called pixels, short for picture elements. Next, with the help of monochromatic images, we can generate color images. Precisely, a color image is generated by taking a tensor product of three matrices which are the decompositions of the original image into blue, green, and red components. These three primary colors, red, blue and green (RGB) are linearly independent which means none of the colors can be obtained by any combination of the other two. The tensor product of three matrices can be viewed as three matrices stacked one on top of the other in the order of blue, green and red from top to bottom. The multi-spectral images which have more than red, green, and blue components can also be visualized as the stack of more than three matrices stacked one on top of the other. Thus individual two-dimensional images (monochromatic images) are combined to form color images so that a color or multi-spectral image is handled by handling each monochromatic image, one at a time and repeating the process for the rest of the color components in the tensor product. Due to this reason, we will mainly consider monochromatic images in the first 3 chapters of the report and give comparisons of classification methods on RGB colored images in chapter 4.

1.2 Image analysis

Simply speaking, image analysis is a study in which we analyze image features and solve the image-related problems using matrix computations or some other mathematical tools. Image analysis is also known as image processing. The objective of image analysis ranges from observing and identifying image features to transforming images into different forms using these features. There are many areas where image analysis is applied, such as remote sensing, medicine, astronomy, space exploration, ultrasonic imaging etc. Now we discuss a core process of the image analysis, known as image classification. Image classification is the process of assigning the pixels of an image to a specific class or category to identify the image features. We begin with the discussion of image pixels classification with the introduction of image classes.

1.2.1 Image classes

We recall that a digital image is a rectangular arrangement of many pixels (picture elements) which are the smallest units of the digital image and that the level of grey or intensity of these pixels ranges from 0 to 255 depending on the brightness of the locations in the image. So the pixels representing a particular feature or a color in an image show more homogeneity in terms of the distribution followed by the data set of pixels. Hence by comparing image pixels with each other, and to pixels of known identity, we can form groups of similar image pixels into different classes. In this way, classes in an image are formed.

Theoretically, these classes are homogeneous in the sense that pixels with classes are spectrally more similar to each other than they are to pixels in the other classes. Although, in practice, pixels in a class will display some variations. These classes then represent different informational categories of interest in the image. Now, in terms of distribution, pixels representing any two classes, so formed, may follow either two different distributions or a same distribution with different parameters. For example, let us consider two classes, say class1 and class2. Then the data in class1 may follow $N(\mu_1, \sigma^2)$ and the data in class2 may follow $N(\mu_2, \sigma^2)$ which is an example of image classes with the same distribution with different parameters. If, on the other hand, the data in class1 follows the $N(\mu, \sigma^2)$ and the data in class2 follows χ^2 we get an example of image classes with two different distributions.

1.2.2 Image classification

Broadly speaking, classification is a multivariate analysis task and as the name suggests, it basically deals with classifying a new observation into one of the classes of interest. Our main focus in this report will be on the classification of image pixels. In the case of images, classification is a process of observing and identifying features of an image. More generally, it is a process of assigning pixels to different classes in the image.

Images can be considered as a finite collection of regions identified as a number of predetermined classes. But these parts or the image itself may not be identifiable to the human eye. In order to view the image parts as something familiar, we need to perform image classification. Thus with the help of image classification, we can observe and identify different image features which have many practical applications. The main part of image classification is to obtain a recognition system that classifies the parts of an image into different identified classes. We use random sample of locations, called the training data, from each class of interest to build the recognition system. All the classification methods assume that the image in context depicts one or more image features on it and that each of these features is from one of exclusive and distinct classes. The numerical properties of image features are analyzed in the classification and it organizes the data into different categories. In classification algorithms there are usually two stages of processing namely training and testing. Characteristic properties of typical image features are separated in the training phase which is then followed by a formation of training classes.

In general, there are two different approaches of image classification: supervised and unsupervised. The supervised classification is based on the idea that a user can select sample pixels in an image that work as representative of classes of interest in the image and then direct the image processing software to use these choices as references for the classification of all other pixels in the image. In the unsupervised classification, as the name suggests, groupings of pixels with common characteristics are based on the software analysis of an image without user providing sample classes for the classification.

1.3 Applications of image classification

Here we present an overview of various applications of image classification techniques mostly taken from Green (1983).

1.3.1 Remote Sensing

Remote sensing refers to collecting information about an object without coming into contact with that object. Observations usually consist of measurements of electromagnetic radiation with different wavelengths of the radiation carrying a variety of information about the earth's surface and atmosphere. There are many applications of multispectral image classification in earth applications remote sensing. The major subdivisons of remote sensing that involve use of image classifications are;

Mineral Exploration

Mineral exploration involves the use of multispectral imagery for analysis of surface composition that may indicate mineral deposition for analysis of surface structure. Image classifications can be applied in these multispectral images which is helpful in the mineral exploration. Along with the classification of images, image enhancment, correlation of image data with geographically referenced data bases are also used in the mineral exploration.

Determination of surface composition

Surface composition can be estimated from a knowledge of the reflectance properties of various types of materials. Multispectral imaging provides one mechanism for determining surface composition. With the help of spectral reflectivity properties of various objects on the surface, we can classify them and thus can determine the surface composition.

Land-Use Analysis

Remotely sensed imagery can be used to determine various categories of land utilization. With the help of multispectral classification, we can get information about the land utilization such as residential region, open space, agricultural, forest, water etc. Classification is also useful to monitor the quality of ocean water, especially near coastlines.

1.3.2 Medical Applications

Digital image processing is becoming an increasingly important tool in medical diagnosis. One of the applications of the image classification in medical is in chromosome karyotyping. Analysis of chromosomes samples can provide important insight into disease and genetic defects. A microscope slide containing a set of randomly oriented chromosome is obtained and converted to digital format. Each of the chromosomes is isolated as a single object, and the object is then classified to type using a variety of pattern-recognition techniques.

As a medical application of classification, we can take classification of database of 10000 grey-level anonymous x-ray images which are arbitrarily selected from clinical routine at some hospital. They represent different anatomic body parts and biological systems. Using some classification technique, we can classify the database of x-ray into some categories. Also comparing normal and abnormal blood vessel structures, via the analysis of cell images is central to pathology and medicine.

1.3.3 Astronomy

Digital techniques are widely used in astronomical applications. Digital techniques and digital image sensors provide improved resolution and dynamic range for astronomical applications. Study of galaxies, stars, planets etc can be helpful using the classification technique.

1.4 Motivation

A classification method based on hypothesis testings was developed by Liao and Akritas (2007). This is a powerful non-parametric classification method which works for data following any distributions. In this report, we employ Liao & Akritas's classification in the context of images and come up with a new method of image classification which works better than any other popular classification methods.

The main objective of the image classification is to obtain a dependable object recognition system that classifies all the locations in the image into a number of identified classes. There are many classification methods such as Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Bayesian method of classification, Classification Tree Method (CTM), Random Forest, Test-based classification, to name just a few.

Although, the non-parametric test-based classification introduced by Liao and Akritas (2007) is a powerful classification method, their implementation in the context of images reveals that their method can completely fail to correctly classify all the image pixels in the given image due to small p-values. This means that the Liao & Akritas's method does not work well for the image classification. A non-parametric test-based classification is an effective method of image pixels classification because image pixel classes, in general, can follow any distributions. So, we look for a test-based classification method that works in the context of images. We eliminate the drawback of Liao & Akritas's classification by introducing the minimum distance classification in it and come up with a new test-based classification of image pixels. Thus Liao & Akritas's classification method was the primary motivation of our work in this report.

The idea of the minimum distance and sample evidence from the hypothesis testings are the main tools of our modified classification method. In our implementation of Liao & Akritas's method, we observe that their method fails to classify the images when the test p-values obtained from the hypothesis testings are very small. But our method works well even if the test p-values are small. Again, we compare our method of classification with some of the standard methods of classifications namely Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Classification Tree Method (CTM), and Polyclass method of image classification. We employ these standard methods of classification and our method in some colored images. We verify with our experiments in different images that our method of classification performs better than Liao & Akritas's method and the standard methods of classification in classifying the image pixels. We first employ our classification method for a binary (two classes) classification of image pixels and then extend it for a multiclass (more than two classes) classification of image pixels. We observe that our method prevails in both cases.

Thus our method is a test-based classification where the minimum distance and sample evidence from the hypothesis testings are combined to build the class recognition system and performs better than other standard methods in the classification of image pixels.

1.5 Different methods of Classification

In this section, we give brief summaries of some popular classification methods.

1.5.1 Bayesian Classification

Bayesian classification is a statistical method for classification which assumes an underlying probabilistic model, the Bayes theorem. Bayesian classification is named after Thomas

Bayes, who proposed the Bayes theorem.

Now, we describe the classification of a pattern vector by the Bayes classifier. Suppose that there are k classes of interest, given by $\omega_j, j = 1, 2, \cdots, k$ and x is a n dimensional pattern vector. The probability that a pattern vector x belongs to a class ω_j is given by $P(\omega_j|x)$. Using the Bayes theorem we have, $P(\omega_j \cap x) = P(x|\omega_j)P(\omega_j)$ where $P(x|\omega_j)$ is the probability density function of the pattern vector x in the class ω_j and $P(\omega_j)$ is the probability of occurrence of class ω_j. The decision function for the Bayesian classification is,

$$d_j(x) = P(\omega_j|x) \propto P(x|\omega_j)P(\omega_j)$$

Thus a pattern vector x belongs to class ω_j if $d_j(x) > d_i(x)$ for $i = 1, 2, \cdots, k; i \neq j$. It is often assumed that the data from a class of interest have Gaussian distribution, i.e.,

$$P(x|\omega_j) = \frac{1}{(2\pi)^{\frac{n}{2}}|C_j|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}[(x - m_j)^T {C_j}^{-1}(x - m_j)]\right)$$

where, C_j, m_j are the covariance matrix and mean vector of class ω_j and $|C_j|$ is the determinant of C_j.

As ln is a monotonic function, decision function remains invariant under the ln transformation. Then our decision function becomes,

$$\begin{aligned} d_j(x) &\propto \ln P(x|\omega_j)P(\omega_j) && (1.5.1) \\ &= \ln P(x|\omega_j) + \ln P(\omega_j) && (1.5.2) \\ &= -\frac{1}{2}\ln|C_j| - \frac{1}{2}[(x - m_j)^T {C_j}^{-1}(x - m_j)] + \ln P(\omega_j) - \frac{n}{2}\ln(2\pi.) && (1.5.3) \end{aligned}$$

Since, the term $-\frac{n}{2}\ln(2\pi)$ is independent of number of classes, the decision function for the Bayesian classification is given by,

$$d_j(x) = P(\omega_j|x) \propto -\frac{1}{2}\ln|C_j| - \frac{1}{2}[(x - m_j)^T {C_j}^{-1}(x - m_j)] + \ln P(\omega_j)$$

1.5.2 Test-Based Classification (TBC)

The test-based classification was introduced by Liao and Akritas (2007). This test-based classification does not need any assumptions on the form of the distribution of classes. We now discuss the main idea behind this test based classification.

Liao and Akritas (2007) employ hypothesis testing in their classification method. The p-values of the hypothesis tests which are essentially the values that provide evidence to reject or fail to reject the null hypothesis is the main idea behind Liao & Akritas's classification method.

Suppose that there are two classes, say class 1 and class 2. Let x_0 be a test point. Suppose that class means from two classes are μ_1 and μ_2. Let us consider training vectors with observations $(x_{11}, x_{12}, ...x_{1n_1})$ and $(x_{21}, x_{22}, ...x_{2n_2})$ from class 1 and class 2 respectively. For the classification of the test point x_0, the following two tests are conducted:

- Test 1: Place x_0 with the observations from class 1 and use $(x_0, x_{11}, x_{12}, x_{13}.........x_{1n_1})$ and $(x_{21}, x_{22}, x_{23}.........x_{1n_2})$ to test the null hypothesis $H_0 : \mu_1 = \mu_2$.

- Test 2: Place x_0 together with the observations from class 2 and use $(x_{11}, x_{12}, x_{13}.........x_{1n_1})$ and $(x_0, x_{21}, x_{22}, x_{23}.........x_{1n_2})$ to test the null hypothesis $H_0 : \mu_1 = \mu_2$.

Then, the decision rule for the classification is that x_0 belongs to class 1 if PV_1 is less than PV_2. Similarly, x_0 belongs to class 2 if PV_2 is less than PV_1. This binary classification is then extended to more than two classes case.

1.5.3 Linear Discriminant Analysis(LDA)

Linear discriminant analysis (LDA) is a method in multivariate analysis and gives us the separation of different classes of objects. It follows the principle of total probability of misclassification and assume the normality distribution for data in each class. We now give a brief overview of binary classification using LDA.

Let p_1, p_2 be the prior probabilities of two classes, say, π_1 and π_2. We would like to assign an object Y to one of the two classes. Let Y be characterized by some vector $X = [x_1, ..., x_p]^T$. Now by using the Bayes's rule the conditional probability of each class is given by:

$$P(\pi_i|X) = \frac{P(X|\pi_i)p_i}{\sum_{i=1}^{2} P(X|\pi_j)p_j}$$

where $P(\pi_i|X)$ is the posterior probability and $P(X|\pi_i)$ is called likelihood function of π_i. The prior probabilities are assumed to be given. If they are not known, then the uniform distribution is used sothat $p_1 = p_2$. We assume that the conditional distributions are multivariate normal, i.e.,

$$P(X|\pi_i) = \frac{1}{(2\pi)^{\frac{p}{2}}|\sum_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}[(X-\mu_i)^T \sum_i^{-1}(X-\mu_i)]\right)$$

where $\mu_i, \sum_i$ are mean and covariance matrices.

$$\log\left[\frac{P(\pi_1|X=x}{P(\pi_2|X=x}\right] = \log\left[\frac{P(X=x|\pi_1)P(\pi_1)}{P(X=x|\pi_2)P(\pi_2)}\right] \tag{1.5.4}$$

$$= \log\left[\frac{p_1\exp(-\frac{1}{2}[(X-\mu_1)^T\sum^{-1}(X-\mu_1)])}{p_2\exp(-\frac{1}{2}[(X-\mu_2)^T\sum^{-1}(X-\mu_2)])}\right] \tag{1.5.5}$$

In LDA, it is assumed that the classes have common covariance matrix, i.e., $\sum_1 = \sum_2$. Thus we have after simplification,

$$\log\left[\frac{P(\pi_1|X=x}{P(\pi_2|X=x}\right] = \log(\frac{p_1}{p_2}) - \frac{1}{2}(\mu_1+\mu_2)^T\sum^{-1}(\mu 1-\mu_2) + x^T\sum^{-1}(\mu_1-\mu_2) \tag{1.5.6}$$

Hence, by minimizing the posterior probability of misclassification, a new observation x_0 belongs to class1 if

$$x_0^T\sum^{-1}(\mu_1-\mu_2) - \frac{1}{2}(\mu_1+\mu_2)^T\sum^{-1}(\mu_1-\mu_2) > \log(\frac{p_1}{p_2})$$

The decision boundary between classes π_1 and π_2, i.e., the set where $P(\pi_1|X=x) = P(\pi_2|X=x)$, is linear in x and is a hyperplane in p-dimension with $p > 1$. In practice, the mean μ_i and covariance matrix $\sum_i$ of classes are unknown and are estimated by using the training data.

1.5.4 Quadratic Discriminant Analysis (QDA)

Quadratic discriminant analysis follows similar principle as LDA and also assume that the distributions are normal. This method is different from the linear discriminant analysis in the sense that it allows the classes to have different covariance matrices. Because of this the decision boundary between the classes is quadratic.

Then using the discussion in the LDA, we have after simplification, an observation x_0 belongs to class π_1 if

$$-\frac{1}{2}x_0^T\left(\sum\nolimits_1^{-1}-\sum\nolimits_2^{-1}\right)x_0+\left(\mu_1^T\sum\nolimits_1^{-1}-\mu_2^T\sum\nolimits_2^{-1}\right)x_0-K>\ln\left[\frac{p_2}{p_1}\right]$$

where

$$K=\frac{1}{2}\log\left[\frac{|\sum_1|}{|\sum_2|}\right]+\frac{1}{2}\left(\mu_1^T\sum\nolimits_1^{-1}\mu_1-\mu_2^T\sum\nolimits_2^{-1}\mu_2\right)$$

Here the surface that separates the classes, is quadratic. Hence, we use the term quadratic in QDA. We estimate the class parameters μ_i , $\sum_i$ by using the training data.

1.5.5 Support Vector Machines (SVM)

Support vector machines are simply a set of related supervised learning methods which analyze data and recognize patterns. The SVM's perform pattern recognition between two point classes with the help of a surface obtained by using certain points of training data and these points are called support vectors. The SVM's is a non-probabilistic binary linear classifier which constructs a hyperplane or a set of hyperplane for the classification. We consider both linearly separable and non-separable data.

The basic idea behind the SVM classification in the linearly separable data is to choose a hyperplane which gives us the maximum separation of two groups of data. In other words, we choose the hyperplane which has the largest margin where margin is the summation of shortest distance from the separating hyperplane to the nearest data of both classes. Such a hyperplane is called maximum-margin hyperplane. In order to address the non-linearly separable data, SVM does a mapping from the input space to a higher dimensional space

where the data is linearly separable and a maximal separating hyperplane is constructed there. Now we give the basic theory of SVM, mostly taken from Vapnik (1982). Suppose that we are given a set S of points x_i, $x_i \in R^n, i = 1, 2, \cdots, N$ and each x_i belongs to either of the two classes. We assign a label $y_i \in \{1, -1\}$. We need to find equation of hyperplane which divides S with all the points of one class in same side and maximizing the minimum distance between either of the two classes and the hyperplane. A hyperplane can be represented by, $W \cdot X - b = 0$ where $\cdot$ represents dot product, W is normal vector and b is the distance from the origin. When the data are linearly separable, W and b are chosen to maximize the distance between two parallel hyperplane which separate the data. These hyperplanes are given by $W \cdot X - b = 1, W \cdot X - b = -1$. But the distance between these hyperplanes is $\frac{2}{\|W\|}$, where, $\|W\|$ is norm of W.
So we minimize $\|W\|$. We need, $W \cdot X - b \geq 1$ for x_i to be in first class and $W \cdot X - b \leq -1$ for x_i to be in second class. Thus we need to minimize $\|W\|$ subjected to the condition $y_i(W \cdot X - b) \geq 1, i = 1, 2, \cdots, N$. After the construction of the hyperplane, it separates the data into two distinct classes.

1.5.6 Classification Tree

Classification tree method (CTM), also known as decision tree method, is an observational method which is used in the classification of explanatory variable. Classification tree method makes no prior assumptions about the data to be classified. Therefore, it is a non-parametric technique. It is simply based on the idea of partition testing. By the means of this method, the input domain of a test object is regarded under various aspects. Then for each such aspect, we form disjoint and complete classification. The stepwise partition of the input domain is represented graphically in the form of a tree. For this reason, it is called classification tree method.
Tree structured classifiers are constructed by repeated splits of subsets of the feature space into two descendent subsets beginning with the feature space itself. More precisely, the decision tree is constructed by recursively partitioning the data set into purer, more homo-

geneous subsets depending on a set of tests applied to one or more attribute values at each node in the tree. All the algorithms developed to split the training data at each internal node of a decision tree into regions that contain examples from just one class, either minimize the impurity of the training data or maximize the goodness of split. The goodness of split is measured by an impurity function defined for each node. The possible impurity functions include entropy, the misclassification rate, and the Gini index. The details of these can be found in Hastie, Tibshirani, and J. (2001). The procedure of creating a tree classifier involves the following three steps:

(1) The selection of splits.

(2) The decisions when to declare a node terminal or to continue splitting it.

(3) The assignment of each terminal node to a class.

The class labels are assigned to terminal nodes based on a majority vote or a weighted vote when it is assumed that certain classes are more likely than others. A tree is composed of a root node which contains all the data, a set of internal nodes (splits), and a set of terminal nodes which are called leaves. Each node in a decision tree has only one parent node and two or more descendent nodes. The data is classified by moving down the tree and sequentially subdividing it according to the decision framework defined by the tree until a leaf is reached. Decision tree classifiers divide the data into subsets, which contain only a single class.

1.5.7 Polyclass

Polyclass model fits a polychotomus logistic regression model using linear splines and their tensor product. It provides estimates for conditional class probabilities which can be estimated to predict class labels. We now give an overview of Polyclass model, most of which has been taken from Stone et al. (1997). Suppose that Y is a qualitative random variable that takes on a finite number $K+1$ of values that we refer to as classes. Depending on a vector of predictors $X \in \mathbb{R}^M$. We would like to predict Y.

As stated earlier, Polyclass uses piecewise linear splines and selected tensor products to model the conditional class probabilities. Precisely, suppose $P(Y = k|X = x) > 0$ for $k \in \mathbf{K} = \{1, \ldots, K+1\}$ and $x \in \mathbf{X}$, where $\mathbf{X}$ is a subset of $\mathbb{R}^M$ over which X ranges. We set,

$$\theta(k|x) = \log \frac{P(Y = k|X = x)}{P(Y = K+1|X = x)}, x \in \mathbf{X} \quad \text{and} \quad k \in \mathbf{K}.$$

Then $\theta(K+1|x) = 0$ for $x \in \mathbf{X}$ and

$$P(Y = k|X = x) = \frac{\exp\theta(k|x)}{\exp\theta(1|x) + \ldots + \exp\theta(K+1|x)}, x \in \mathbf{X} \quad \text{and} \quad k \in \mathbf{K}.$$

This is referred as the polychotomous regression model; when $K = 1$ it is known as the logistic regression model.

Let J be a positive integer and G be a J dimensional linear space of functions on $\mathbf{X}$ with basis $B_1, \ldots, B_J$. Let us consider the model

$$\theta(k|x) = \theta(k|x; \beta k) = \sum_{j=1}^{J} \beta_{jk} B_j(x), x \in \mathbf{X} \quad \text{and} \quad k \in \mathbf{K};$$

where β is the $JK-$dimensional column vector consisting of the entries $\beta_1, \ldots, \beta_K$. Then we set,

$$P(Y = k|X = x; \beta) = \frac{\exp\theta(k|x; \beta)}{\exp\theta(1|x; \beta) + \ldots + \exp\theta(K+1|x; \beta)}$$

for $\beta \in \mathbb{R}^{JK}$, $x \in \mathbf{X}$ and $k \in \mathbf{K}$.

The maximum likelihood estimate of $\theta(k|x)$ is given by $\hat{\theta}(k|x) = \theta(k|x; \hat{\beta})$ where $\hat{\beta}$ is the maximum likelihood estimate given by $l(\hat{\beta}) = \max_\beta l(\beta)$. Then the Polyclass rule of classification is to assign a case with $X = x$ to a class k having the maximum value of $\hat{\theta}(k|x)$

In Polyclass, there are K parameters for each basis function which increases the amount of computation needed for large data sets.

1.6 Organization of the report.

The purpose of this report is to introduce and apply a new method of classification in the context of images with detailed theory and illustrated examples. As prior probabilities of classes also play an important role in our classification, we need to take into account of both

equal and unequal prior probabilities. The rest of the report is organized as below.
Chapter 2 begins with the binary classification, i.e., classification of an image into two classes of interest. In this chapter, we first apply Liao & Akritas's classification method in some images considering the case of equal prior probabilities of classes. After the illustration of failure of Liao & Akritas's method in some images, we introduce our modified method. We illustrate that our classification method works in the images. After considering the case of equal prior probabilities of classes, we then take classes with unequal priors in the images. We then discuss Liao & Akritas in the unequal priors case. Finally, the chapter ends with the application of modified classification in the case of unequal prior probabilities.
In chapter 3, we extend the idea of binary classification to the case of multi-classes. The chapter begins with application of Liao & Akritas's multiclass classification in some images by assuming the equal priors of classes. Then the next section deals with the discussion of theory and illustration with examples of application of our modified method in the images. Similarly we discuss Laio & Akritas and the modified method by considering unequal prior probabilities of classes in applications at the end of the chapter.
In chapter 4, we discuss the comparisons of our method of classification with some standard methods of classifications. Comparisons in binary and multiclass classification of image pixels are performed in colored images.

```
0.7137255 0.7176471 0.7215686 0.7254902 0.7019608 0.7215686
0.7098039 0.7098039 0.7137255 0.7137255 0.7215686 0.7372549
0.7294118 0.7176471 0.7294118 0.7490196 0.7254902 0.7411765
0.7294118 0.7137255 0.7176471 0.7372549 0.7215686 0.7333333
0.7294118 0.7137255 0.7098039 0.7215686 0.7098039 0.7215686
```

\\

```
0.1803922 0.2000000 0.2078431 0.2196078 0.2274510 0.1921569 0.1568627
0.1607843 0.1882353 0.2039216 0.2156863 0.2196078 0.1843137 0.1607843
```

0.1647059 0.1843137 0.1882353 0.1882353 0.1882353 0.1725490 0.1803922

0.1921569 0.1764706 0.1490196 0.1529412 0.1764706 0.1725490 0.1764706

0.2235294 0.2117647 0.1921569 0.1960784 0.2196078 0.2117647 0.2078431

0.1960784 0.1960784 0.1843137 0.1960784 0.2156863 0.2078431 0.1960784

0.1568627 0.1686275 0.1686275 0.1803922 0.2039216 0.2000000 0.1843137

\\

0.8627451 0.8588235 0.8509804 0.8509804 0.8509804 0.8431373 0.8313725

0.8627451 0.8588235 0.8549020 0.8549020 0.8509804 0.8392157 0.8313725

0.8666667 0.8588235 0.8549020 0.8549020 0.8549020 0.8470588 0.8352941

0.8666667 0.8588235 0.8588235 0.8588235 0.8588235 0.8509804 0.8392157

0.8666667 0.8627451 0.8588235 0.8588235 0.8627451 0.8549020 0.8470588

0.8666667 0.8627451 0.8627451 0.8627451 0.8666667 0.8666667 0.8509804

0.8705882 0.8666667 0.8627451 0.8666667 0.8666667 0.8627451 0.8549020

0.8745098 0.8666667 0.8627451 0.8666667 0.8705882 0.8666667 0.8549020

\\

0.6039216 0.5764706 0.5882353 0.6039216 0.6117647 0.6313725

0.6039216 0.5725490 0.5843137 0.5960784 0.6000000 0.6156863

0.6078431 0.5764706 0.5843137 0.5921569 0.5921569 0.6039216

0.6196078 0.5882353 0.5921569 0.5960784 0.5882353 0.5960784

0.6274510 0.6039216 0.6078431 0.6078431 0.6000000 0.6039216

Chapter 2

Binary classification of image pixels.

We begin this chapter with the application of test-based classification (TBC) introduced by Liao & Akritas (2007) in the case of images. Liao & Akritas (2007) employ hypothesis testing in their classification method. The p-values of the hypothesis tests which are essentially the values that provide evidence to reject or fail to reject the null hypothesis is the main idea behind Liao & Akritas's classification method. We would like to do a binary classification of image pixels and we use some grey scale images of size 512×512 for our purpose. In this chapter, we consider only two classes of pixels in a given image and our goal is to identify each pixel of the image as of one class or the other. After implementation of Liao and Akritas (2007) for image pixel classification, we found that this method can completely fail to give correct classification of image pixels when both test p-values are small. We will develop a class recognition system or a classification function which assigns a class to every pixel of the image. We will consider both cases of equal and unequal prior probabilities of classes in the image.

2.1 Binary Classification with equal prior using Liao and Akritas (2007).

Here, we give a description of Liao & Akritas's binary classification with equal prior probabilities of classes. Let p_1, p_2 be the prior probabilities of the classes of interest. Equal prior means $p_1 = p_2 = \frac{1}{2}$. We will describe how to apply Liao & Akritas's binary classification method in the context of images below.

2.1.1 Training data, classes and test points.

In this section we give a description about the formation of training data, classes and test points in a given image. This will also be used in later chapters.

We first define our classes of interest in the given image. In an image, we can define our classes of interest by selecting the regions marked with different colors in it. We use some data that is known a priori to belong to the involved classes to train the system about these classes and learn the class parameters. This data is referred to as training data. The training data of each class contains information about that class which is then used to compute class parameters.

Let us now take a rectangular part of a region to acquire training data of that class. We do this by choosing two points in the region which will be the end points of the main diagonal of the rectangle. Then all the pixels in this rectangular region form the training data for its corresponding class. In this way, we select two sets of pixel values in the given image to define our classes and these sets serve as training data for the classes. These training data may follow either two different distributions or the same distribution with different parameters.

In the classification of images, we would like to classify a randomly selected pixel in the images as belonging to one of the defined classes. This randomly selected pixel is called a test point. We then use the information provided by the class parameters to classify the test points in the image.

2.1.2 Description of the method.

We take a standard grey scale image of size 512×512. Let class 1 and class 2 denote the two classes of interest in the image. The training data in class 1 can be obtained by selecting a rectangle from region1 as described in Section 2.1.1 which will be a submatrix of the 512×512 matrix. Next, we put it into a vector form by adjoining each column of the submatrix below its preceding column. We treat this vector of pixels as data from class 1. As the name suggests we need to employ statistical tests for this classification and the type of the test we use in this method strictly depends on how we define the classes of interest. In some cases, the pixel data from a class may resemble a normal random variable while others do not follow the normal distribution. For this reason, we need a flexible test that applies to data with general distributions. We choose Wilcoxon rank sum test and student t-test as our statistical tests. Since Wilcoxon rank sum test does not require assumptions about the form of the distributions of the measurements, we use this test when pixel values do not follow the normal distribution. On the other hand, we use the student t-test when pixel values appear to be reasonably normal.

Let x_0 be a test point. Suppose that class means from the two classes are μ_1 and μ_2. Let us consider training vectors with observations $(x_0, x_{11}, x_{12}, x_{13}.........x_{1n_1})$ and $(x_{21}, x_{22}, x_{23}.........x_{1n_2})$ from class 1 and class 2 respectively. For the classification of the test point x_0, the following two tests are conducted.

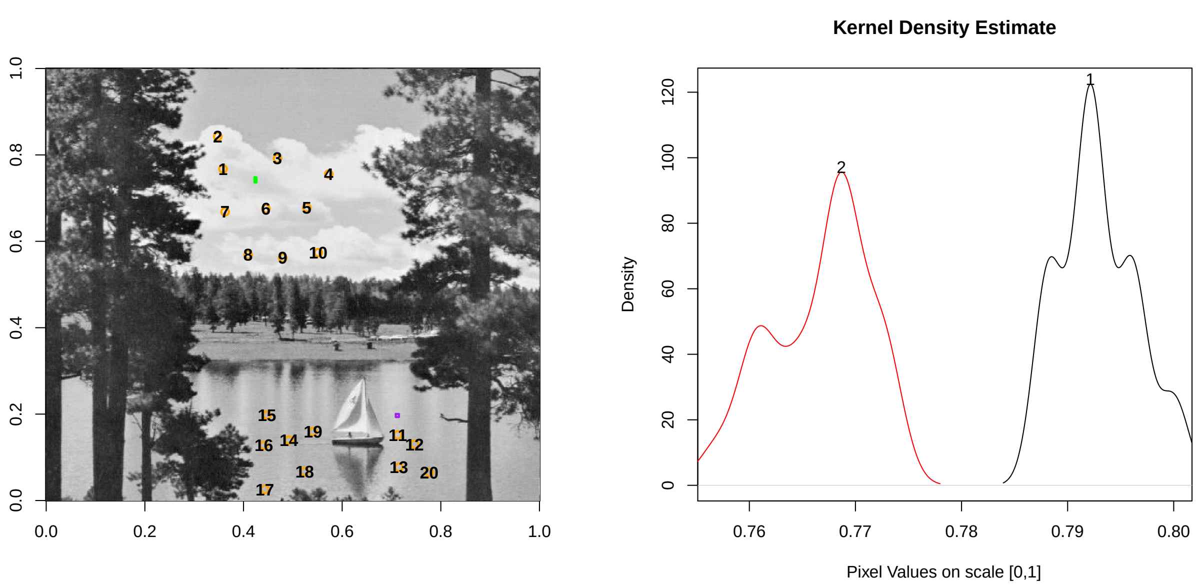
- Test 1: Place x_0 with the observations from class 1 and use $(x_0, x_{11}, x_{12}, x_{13}.........x_{1n_1})$ and $(x_{21}, x_{22}, x_{23}.........x_{1n_2})$ to test the null hypothesis H_0. The H_0 for the Wilcoxon rank sum test is that class 1 and class 2 have identical distribution and the H_0 for the t-test is $\mu_1 = \mu_2$.

- Test 2: Place x_0 with the observation from class 2 and use $(x_{11}, x_{12}, x_{13}.........x_{1n_1})$ and $(x_0, x_{21}, x_{22}, x_{23}.........x_{1n_2})$ to test the null hypothesis H_0. The H_0 for the Wilcoxon rank sum test is that class 1 and class 2 have identical distribution and the H_0 for the t-test is $\mu_1 = \mu_2$.

Let PV_1 and PV_2 be the p-values from Test 1 and Test 2 respectively. In a hypothesis testing, if the p-value is smaller than the significance level, we reject the null hypothesis. A small PV_1 and a large PV_2 suggests that putting this observation in class 1 will maintain the difference of the two classes. On the other hand, putting this observation in class 2 will blur the boundary between the two classes. Therefore, the decision rule for the classification is that x_0 belongs to class 1 if PV_1 is less than PV_2. Similarly, x_0 belongs to class 2 if PV_2 is less than PV_1.

Let us apply this method of classification to a specific image, namely the image in Figure 2.1(a). Classes are formed by choosing some regions which represent different levels of gray-scale so that classes so formed will be different from each other. We choose regions representing sky in the image as our class 1 and water regions as class 2 and form training data for these classes. For the classification purpose, we select 20 observation (test) points labeled with numbers such that first 10 of them are chosen from regions representing class 1 and the rest are taken from class 2 regions as shown in the Figure 2.1(a). Density plot of the classes in Figure 2.1(b) shows that classes so formed are distinct and separated.

Figure 2.1



(a) Image with training data and test points.

(b) Density plot of classes.

We employ Liao & Akritas method of classification as discussed above to classify the selected test points in the given image. The classification of these test points are given in Table 2.1. From the table, we see that all of the test points are classified correctly.

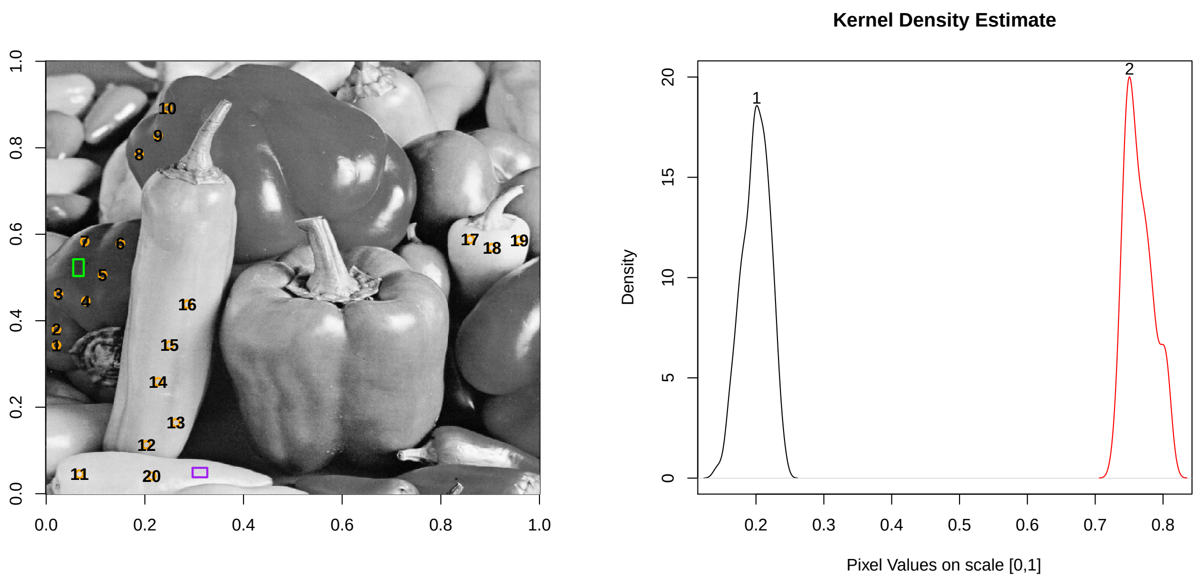
Table 2.1: Classification by Liao-Akritas method in image Figure 2.1(a)

TP	LA	PV1	PV2	Obs
1	class 1	2.281097e-08	4.160598e-07	0.8392157
2	class 1	2.281097e-08	4.160598e-07	0.8784314
3	class 1	2.281097e-08	4.160598e-07	0.8392157
4	class 1	2.281097e-08	4.160598e-07	0.8784314
5	class 1	2.281097e-08	4.160598e-07	0.8156863
6	class 1	2.281097e-08	4.160598e-07	0.8705882
7	class 1	2.281097e-08	4.160598e-07	0.8470588
8	class 1	2.281097e-08	4.160598e-07	0.8078431
9	class 1	2.281097e-08	4.160598e-07	0.8117647
10	class 1	2.281097e-08	4.160598e-07	0.8705882
11	class 2	3.526614e-07	2.19588e-08	0.7568627
12	class 2	3.775556e-07	2.198816e-08	0.6941176
13	class 2	3.775556e-07	2.198816e-08	0.7254902
14	class 2	3.775556e-07	2.198816e-08	0.7058824
15	class 2	3.775556e-07	2.198816e-08	0.7058824
16	class 2	3.775556e-07	2.198816e-08	0.7137255
17	class 2	3.775556e-07	2.198816e-08	0.6941176
18	class 2	3.775556e-07	2.198816e-08	0.682353
19	class 2	3.775556e-07	2.198816e-08	0.717647
20	class 2	3.775556e-07	2.198816e-08	0.6862745

PV1= p-value from test 1, PV2= p-value from test 2,
TP= Test point, Obs= Test point pixel,
LA= Liao & Akritas's classification result

This classification method of Liao and Akritas (2007) works well as long as at least one of the test p-values is large. When both p-values are very small, their method does not work. We illustrate with an experiment that their method fails when both p-values are small. For this experiment, we consider a pepper image and form training data for the classes as given in Figure 2.2(a). In this image, we take white color as our first class (class 1) and dark color as our second class (class 2) so that classes are distinct with each other and form the training data accordingly. Density plot of classes are shown in the Figure 2.2(b), where 1 and 2 represent the plot for class 1 and class 2 respectively. As before, we select 20 test points labeled with numbers in the given image such that half of them are taken from class 1 and the rest are from class 2.

Figure 2.2



(a) Image with training data and test points.

(b) Density plot of classes.

The classification of these test points along with their test p-values are given in Table 2.2.

From the Table 2.2, we observe that the Liao & Akritas method classifies most of the test points as class 2 even though half of the points were chosen from class 1 regions. This means Liao & Akritas's method misclassifies some test points in the given image. We note that for all the test points selected, the corresponding test p-values are very small. This misclassification is due to the fact that for a valid test, the p-values theoretically follow the uniform $(0, 1)$ distribution. In fact, both p-values that are smaller than the significance

Table 2.2: Classification by Liao & Akritas method in image Figure 2.2(a)

TP	LA	PV1	PV2	Obs
1	class 2	2.298138e-76	2.057060e-76	0.2392157
2	class 2	2.298283e-76	1.991224e-76	0.2627451
3	class 2	2.298283e-76	1.991224e-76	0.2823529
4	class 2	2.298283e-76	1.991224e-76	0.2627451
5	class 1	2.291623e-76	1.069815e-75	0.2000000
6	class 1	2.296688e-76	2.960813e-76	0.2235294
7	class 2	2.298138e-76	2.057060e-76	0.2392157
8	class 2	2.298283e-76	1.991224e-76	0.3294118
9	class 2	2.298283e-76	1.991224e-76	0.3176471
10	class 2	2.298283e-76	1.991224e-76	0.3176471
11	class 2	1.791306e-75	1.99047e-76	0.7803922
12	class 2	2.298283e-76	1.991224e-76	0.6862745
13	class 2	2.298283e-76	1.991224e-76	0.717647
14	class 2	2.298283e-76	1.991224e-76	0.7058824
15	class 2	2.298283e-76	1.991224e-76	0.7058824
16	class 2	2.298283e-76	1.991224e-76	0.7098039
17	class 2	1.380736e-75	1.988376e-76	0.772549
18	class 2	2.233405e-75	1.991161e-76	0.7882353
19	class 2	3.423543e-75	1.991224e-76	0.8196078
20	class 2	4.695842e-76	1.987644e-76	0.7490196

PV1= p-value from test 1, PV2= p-value from test 2,
TP= Test point, Obs= Test point pixel,
LA= Liao & Akritas's classification result

level do not give different level of evidence to reject the null hypothesis. Hence, the Liao & Akritas method does not work well when both the test p-values are very small. In the next section, we propose a modified classification method.

2.2 Classification based on combined evidence from minimum distance and hypothesis tests.

In this section, we present a new classification method and apply this method to some images. In this modified method, we consider two different scenarios based on the test p-values. When both the test p-values are very small, we make a decision rule depending on the distance from the class means. When at least one p-value is larger than the significance level, we make the decision rule same as that of Liao & Akritas discussed in Section 2.1. The main idea behind the minimum distance classification method is to calculate the distance of the test point to the classes and then decide the class of the observation with the minimum distance. The following gives the detailed classification rule for the binary classification of image pixels:

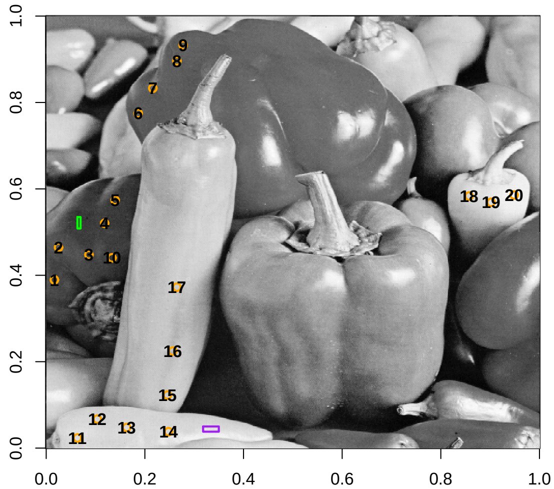
- If PV1 and PV2 are the test p-values obtained in the hypothesis testing discussed in Section 2.1.2 and $\max(PV1, PV2) \geq 0.0001$(threshold), i.e., if at least one of the p-values is larger than the threshold value, then assign the test point x_0 to the class with the smaller p-value.

- If $\max(PV1, PV2) < 0.0001$(threshold), i.e., both p-values are smaller than the threshold value, then assign the test point x_0 to the class 1 if the distance of x_0 to class 1 is less than the distance of x_0 to class 2. If the distance of x_0 to class 1 is larger than that to class 2, we classify x_0 to class 2.

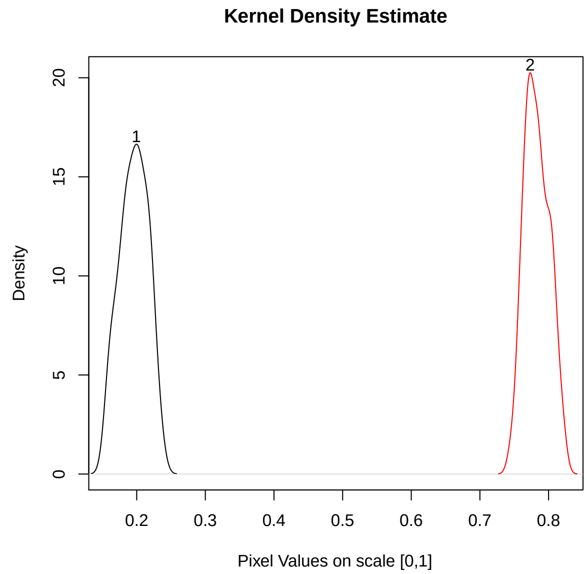
The distance of a point x_0 to a class can take one of the traditional forms such as complete linkage, single linkage, average linkage, etc., or simply, the distance between x_0 and the central tendency of class pixel values. In our experiments, we employ the distance of x_0 to the mean pixel values of each class.

We now apply this modified classification method in the same image, namely, Pepper, where Liao & Akritas's method failed earlier. Similar to the image in the Figure 2.2(a), we define white color and dark color as our class 1 and class 2 and form similar training data for the classes as shown in Figure 2.3(a) which also displays 20 test points. The density plot of the classes show that the classes are distinct in the image.

Figure 2.3



(a) Image with training data and test points.



(b) Density plot of classes.

Table 2.3 compares the classification given by Liao & Akritas method with the modified method discussed in this section. Liao & Akritas's method misclassifies 8 out of 20 test points whereas our method has no misclassifications. We note that for all the test points selected, their corresponding test p-values from both the tests are very small. That is the reason why we have many misclassifications of test points by Liao & Akritas's method.

Finally, we also perform the above comparison between the Liao-Akritas method and modified classification method in another image. We take the image in Figure 2.4(a) and form two classes where pixels representing sky are taken as class 1 and pixels representing vegetation are taken as class 2. These regions represent two different levels of gray-scale in the image. Some test points are selected as before and density plot in Figure 2.4(b) indicates that classes are distinct and well separated. Table 2.4 shows the classification results in this image. We observe from Table 2.4 that modified classification method provides an accurate classification of image pixels.

Table 2.3: Comparison between Liao-Akritas and modified method in image Figure 2.3(a)

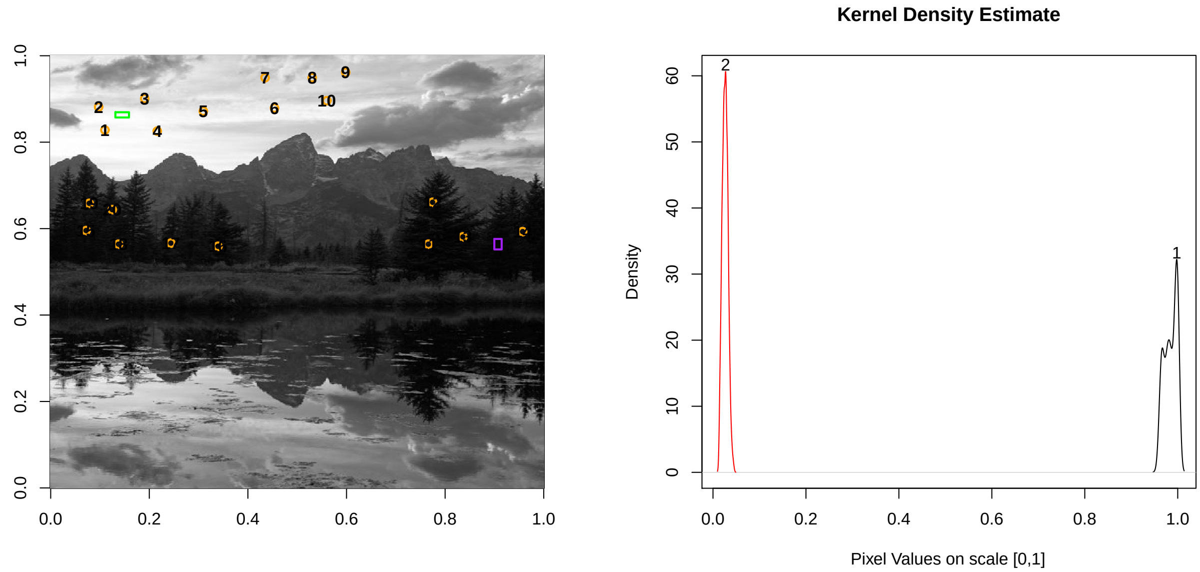
TP	LA	Obs	Our	PV1	PV2	d1	d2
1	class 1	0.294117	class 1	1.440798e-33	2.222016e-33	0.0983006	0.488840
2	class 1	0.309803	class 1	1.440798e-33	2.222016e-33	0.113986	0.473154
3	class 1	0.235294	class 1	1.440732e-33	2.253694e-33	0.0394771	0.547664
4	class 1	0.215686	class 1	1.440398e-33	3.447916e-33	0.0198692	0.5672722
5	class 1	0.223529	class 1	1.440598e-33	2.560586e-33	0.0277124	0.559429
6	class 1	0.309803	class 1	1.440798e-33	2.222016e-33	0.113986	0.473154
7	class 1	0.368627	class 1	1.440798e-33	2.222016e-33	0.172810	0.414331
8	class 1	0.317647	class 1	1.440798e-33	2.222016e-33	0.121830	0.465311
9	class 1	0.364705	class 1	1.440798e-33	2.222016e-33	0.168888	0.418252
10	class 1	0.168627	class 1	1.440398e-33	1.451305e-32	0.0271895	0.614331
11	class 1	0.749019	class 2	1.525001e-33	2.222016e-33	0.553202	0.0339388
12	class 1	0.749019	class 2	1.525001e-33	2.222016e-33	0.553202	0.0339388
13	class 1	0.756862	class 2	1.757262e-33	2.221709e-33	0.561045	0.0260957
14	class 1	0.760784	class 2	2.135574e-33	2.214053e-33	0.564967	0.0221741
15	class 1	0.721568	class 2	1.440798e-33	2.222016e-33	0.525751	0.0613898
16	class 1	0.631372	class 2	1.440798e-33	2.222016e-33	0.435555	0.151585
17	class 1	0.705882	class 2	1.440798e-33	2.222016e-33	0.510065	0.0770761
18	class 1	0.74509	class 2	1.482100e-33	2.221709e-33	0.549281	0.0378604
19	class 2	0.776470	class 2	6.992565e-33	2.216398e-33	0.580653	0.00648788
20	class 2	0.835294	class 2	6.523792e-32	2.222016e-33	0.639477	0.0523356

PVi= p-value for Test i, di= distance of the test point to the mean of class i
Our= Modified method discussed in Section 2.2, LA= Liao & Akritas's method
Obs=Pixel of test point, TP= Test point.

2.3 Binary classification with unequal prior probabilities using Liao and Akritas (2007).

In this section, we give a description of binary classification with unequal prior probabilities of classes introduced by Liao & Akritas. We first discuss the main idea of binary classification with unequal prior probabilities from Liao and Akritas (2007). Let p_1 and p_2 be the prior probabilities of class 1 and class 2 where p_1 and p_2 are not necessarily equal. Then the equal

Figure 2.4



(a) Image with training data and test points. (b) Density plot of classes.

prior probabilities as in Section 2.1 is a particular case of this more general case. Following Bayes theorem, An observational point x_0 belongs to class 1 if

$$\frac{p_1 f_1(x_0)}{p_1 f_1(x_0) + p_2 f_2(x_0)} > \frac{p_2 f_2(x_0)}{p_1 f_1(x_0) + p_2 f_2(x_0)}.$$

where $f_1(x_0)$ represents the density function associated with x_0 for class 1 and $f_2(x_0)$ represents the density function associated with x_0 for class 2. This classifier is the well known Bayes classifier that achieves minimum misclassification error for $0-1$ loss function. We can think of $\dfrac{f_1(x_0)}{f_1(x_0) + f_2(x_0)}$ as the relative probability that the point x_0 is from class 1. Suppose $PV_1(x_0)$ and $PV_2(x_0)$ are the test p-values described in Section 2.1.2. We incorporate the idea of unequal prior probabilities in the test-based classification (TBC) methodology by assuming $\dfrac{PV_1(x_0)}{PV_1(x_0) + PV_2(x_0)}$ as the relative test-based probability that the point x_0 is not from class 1 so that $\left(1 - \dfrac{PV_1(x_0)}{PV_1(x_0) + PV_2(x_0)}\right)$ works as the relative test based probability that the point x_0 is from class 1. Hence the TBC classification rule for the case of unequal prior probabilities is as follows:

Table 2.4: Comparison between Liao-Akritas and modified method in image Figure 2.4(a)

TP	LA	Obs	Our	PV1	PV2	d1	d2
1	class 1	0.9686275	class 1	3.470301e-41	6.497954e-41	0.0160014	0.9424326
2	class 1	0.909804	class 1	3.475206e-41	3.842559e-41	0.07482493	0.883609
3	class 1	0.8745098	class 1	3.475206e-41	3.842559e-41	0.1101190	0.848315
4	class 1	0.972549	class 1	3.473843e-41	7.558878e-41	0.01207983	0.9463542
5	class 1	0.9254902	class 1	3.475206e-41	3.842559e-41	0.05913866	0.8992953
6	class 1	0.9372549	class 1	3.475206e-41	3.842559e-41	0.04737395	0.91106
7	class 1	0.7960784	class 1	3.475206e-41	3.842559e-41	0.1885504	0.7698836
8	class 1	0.8392157	class 1	3.475206e-41	3.842559e-41	0.1454132	0.8130208
9	class 1	0.8000000	class 1	3.475206e-41	3.842559e-41	0.1846289	0.7738051
10	class 1	0.8117647	class 1	3.475206e-41	3.842559e-41	0.1728641	0.7855699
11	class 1	0.04313725	class 2	3.518989e-41	3.842408e-41	0.9414916	0.01694240
12	class 2	0.02745098	class 2	1.300644e-40	3.768573e-41	0.9571779	0.001256127
13	class 2	0.01960784	class 2	5.152598e-40	3.807927e-41	0.965021	0.00658701
14	class 1	0.04313725	class 2	3.518989e-41	3.842408e-41	0.9414916	0.01694240
15	class 1	0.05490196	class 2	3.475206e-41	3.842559e-41	0.9297269	0.02870711
16	class 1	0.05098039	class 2	3.475206e-41	3.842559e-41	0.9336485	0.02478554
17	class 1	0.04705882	class 2	3.475206e-41	3.842559e-41	0.93757	0.02086397
18	class 1	0.08235294	class 2	3.475206e-41	3.842559e-41	0.902276	0.05615809
19	class 1	0.05490196	class 2	3.475206e-41	3.842559e-41	0.9297269	0.02870711
20	class 1	0.06666667	class 2	3.475206e-41	3.842559e-41	0.9179622	0.04047181

PVi= p-value for Test i, di= distance of the test point to the mean of class i
Our= Modified method discussed in Section 2.2, LA= Liao & Akritas's method
Obs=Pixel of test point, TP= Test point.

We classify observation x_0 to class 1 if

$$p_1\left(1-\frac{PV_1(x_0)}{PV_1(x_0)+PV_2(x_0)}\right) > p_2\left(1-\frac{PV_2(x_0)}{PV_1(x_0)+PV_2(x_0)}\right).$$

More precisely, x_0 is classified to class 1 if

$$(1-p_2)PV_2(x_0) > (1-p_1)PV_1(x_0).$$

Similarly, the test point x_0 belongs to class 2 if

$$(1 - p_1)PV_1(x_0) > (1 - p_2)PV_2(x_0).$$

Next, we describe the Liao & Akritas method with unequal prior probabilities in the case of images.

2.3.1 Application of Liao & Akritas method with unequal prior in image pixel classification.

Here, we describe the method by taking a grey scale image of 512×512. The classes are defined in the given image and then training data and test points are formed by following the procedure discussed in Section 2.1.1. Let $(x_{11}, x_{12}, ...x_{1n_1})$ and $(x_{21}, x_{22}, ...x_{2n_2})$ be the observations for the training vectors for class 1 and class 2. Suppose that μ_1 and μ_2 are the class means and x_0 is a randomly chosen test point. We use Wilcoxon rank sum test or student t-test as our statistical tests depending on the form of the distribution of data set for the classes. Let $\lambda = \dfrac{\mu_1 + \mu_2}{2}$. Using this λ, we define the prior probabilities of the classes in images as follows:

- If μ_1 is less than μ_2, then

  $$\text{Prior of class } 1 = \frac{\text{Number of pixels in the training data which are less than the } \lambda \text{ entrywise}}{\text{Number of pixels in the training data.}}$$

  $\therefore$ Prior of class $2 = 1 -$ Prior of class 1.

- If μ_2 is less than μ_1, then

  $$\text{Prior of class } 2 = \frac{\text{Number of pixels in the training data which are less than the } \lambda \text{ entrywise}}{\text{Number of pixels in the training data}}$$

  $\therefore$ Prior class $1 = 1 -$ Prior of class 2.

We can also define prior probabilities of classes as:

$$\text{Prior of class } 1 = \frac{N_1}{N_1 + N_2} \quad \text{and} \quad \text{Prior of class } 2 = \frac{N_2}{N_1 + N_2}$$

where N_1 and N_2 are the number of pixel values in the training data for the classes 1 and 2 respectively. For the classification of the test point x_0, we perform the following two tests:

- Test 1: Place x_0 with the observations from class 1 and use $(x_0, x_{11}, x_{12}, x_{13}.........x_{1n_1})$ and $(x_{21}, x_{22}, x_{23}.........x_{1n_2})$ to test the null hypothesis H_0. The H_0 for the Wilcoxon rank sum test is that class 1 and class 2 have identical distribution and the H_0 for the t-test is $\mu_1 = \mu_2$.

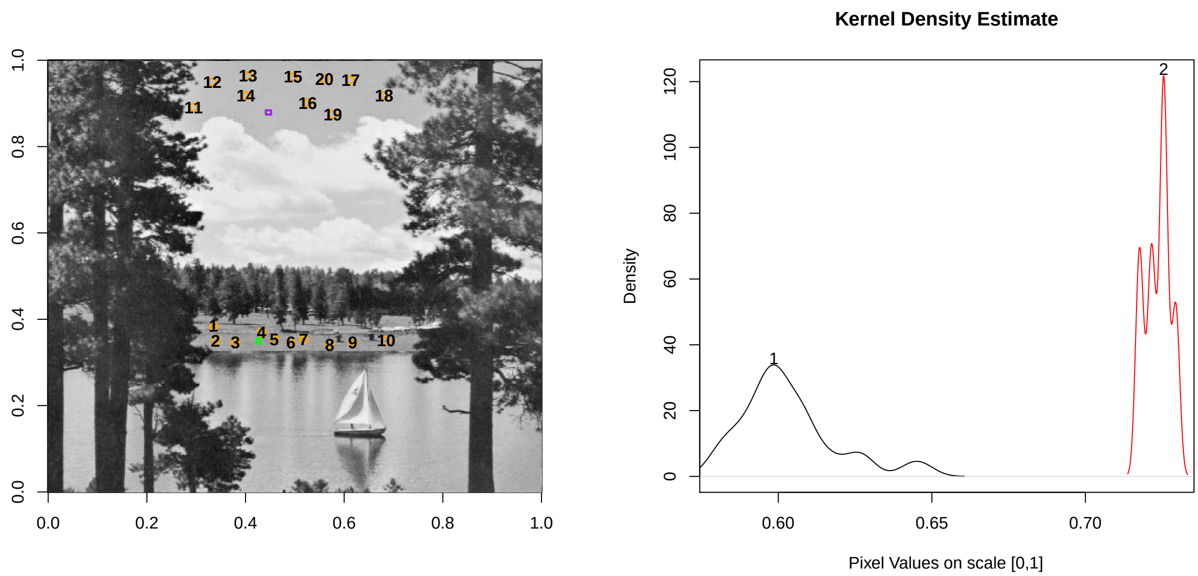
- Test 2: Place x_0 with the observation from class 2 and use $(x_{11}, x_{12}, x_{13}.........x_{1n_1})$ and $(x_0, x_{21}, x_{22}, x_{23}.........x_{1n_2})$ to test the null hypothesis H_0. The H_0 for the Wilcoxon rank sum test is that class 1 and class 2 have identical distribution and the H_0 for the t-test is $\mu_1 = \mu_2$.

Then the decision rule for the classification is that the test point x_0 belongs to class 1 or class 2 depending on $PV_1(1 - \text{prior of class 1})$ is smaller or greater than $PV_2(1 - \text{prior of class 2})$. Hence, in terms of prior probabilities, we observe that the higher the prior probability of a class, the higher the chance for a text point to be in that class.

Next, we apply this method to an image given in Figure 2.5(a). We define the pixels defining vegetation (grass) as class 1 and that of sky as class 2. We then form training data for the classes and select some test points in the image as shown in Figure 2.5(a). Density plot of the classes in Figure 2.5(b) show that the classes are distinct with each other. The prior probabilities of classes are calculated by the method discussed earlier and are displayed in Table 2.5 which also exhibits test p-values and classification of selected test points. All the test points are classified correctly.

As observed in the case of equal prior probabilities, Liao & Akritas method works well for the images when at least one of the test p-values is large and that their method does not work when both the p-values are small. In order to illustrate this, we consider an image given in Figure 2.6(a). In this image we select sky region and vegetation region to represent two different levels of gray-scale to form two distinct classes. The classes so formed are distinct and separated as shown by Figure 2.6(a). Then training data are formed and test points are selected from both regions for classification as shown in Figure 2.6(a). Using the

Figure 2.5



(a) Image with training data and test points.

(b) Density plot of classes.

decision rule of Liao & Akritas, we classify these points and are shown in Table 2.6 which also shows the test p-values and prior probabilities of classes.

From the Table 2.6, we observe that 10 test points are misclassified by Liao & Akritas's method. Since p-values in a valid test follow the uniform $(0, 1)$ distribution, p-values smaller than the significance level do not provide differential evidence in rejecting the null hypothesis. Hence, we conclude that Liao & Akritas method for the unequal prior probabilities does not work for the images when both p-values are small. In the next section, we propose a modified classification method.

2.4 Modified classification based on combined evidence from minimum distance and hypothesis tests.

In the modified method, classification decision depends upon both the p-values obtained from the hypothesis tests and the distance of the test point to each class. The prior probabilities of the classes are obtained by following the method in Section 2.3.1. When both

Table 2.5: Classification by Liao-Akritas in image Figure 2.5(a)

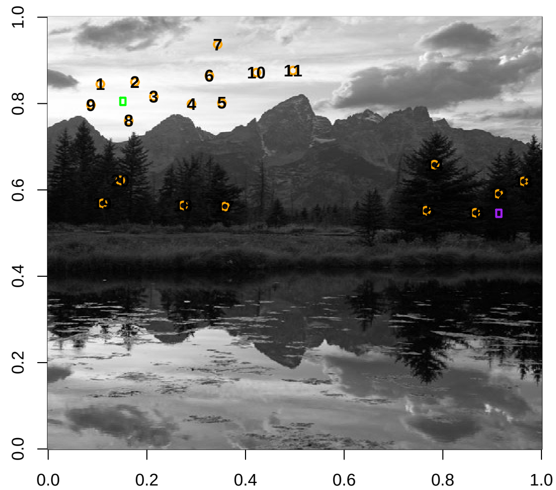
TP	LA	PV1	PV2	Pr1	Pr2	Obs
1	class 1	2.162039e-13	2.615103e-13	0.4929577	0.5070423	0.6313725
2	class 1	2.162039e-13	2.615103e-13	0.4929577	0.5070423	0.6313725
3	class 1	2.162039e-13	3.981934e-13	0.4929577	0.5070423	0.6117647
4	class 1	2.159146e-13	3.647074e-12	0.4929577	0.5070423	0.5803922
5	class 1	2.162039e-13	3.956792e-12	0.4929577	0.5070423	0.5568627
6	class 1	2.147611e-13	1.549740e-12	0.4929577	0.5070423	0.5960784
7	class 1	2.161074e-13	3.226799e-13	0.4929577	0.5070423	0.6235294
8	class 1	2.159146e-13	3.656894e-13	0.4929577	0.5070423	0.6156863
9	class 1	2.135179e-13	9.418029e-13	0.4929577	0.5070423	0.6000000
10	class 1	2.162039e-13	3.956792e-12	0.4929577	0.5070423	0.5058824
11	class 2	4.197354e-12	2.208323e-13	0.4929577	0.5070423	0.7647059
12	class 2	3.270928e-12	2.187733e-13	0.4929577	0.5070423	0.7294118
13	class 2	2.162039e-13	2.208323e-13	0.4929577	0.5070423	0.7098039
14	class 2	5.826353e-13	2.173134e-13	0.4929577	0.5070423	0.7215686
15	class 2	2.162039e-13	2.208323e-13	0.4929577	0.5070423	0.7137255
16	class 2	5.826353e-13	2.173134e-13	0.4929577	0.5070423	0.7215686
17	class 2	5.826353e-13	2.173134e-13	0.4929577	0.5070423	0.7215686
18	class 2	2.162039e-13	2.208323e-13	0.4929577	0.5070423	0.7098039
19	class 2	5.826353e-13	2.173134e-13	0.4929577	0.5070423	0.7215686
20	class 2	2.162039e-13	2.208323e-13	0.4929577	0.5070423	0.7137255

PVi= P-value for Test i, Pri= Prior probability of class i
LA= Liao & Akritas's method, Obs=Pixel of test point, TP= Test point.

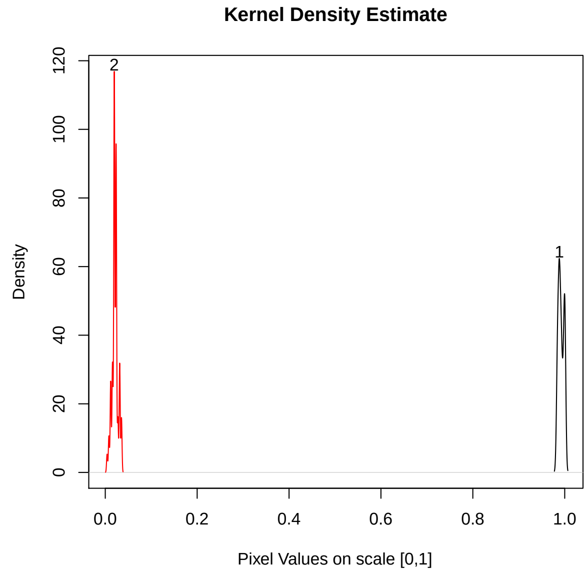
p-values are larger than the significance level, we follow the method of Liao & Akritas discussed in Section 2.3.1. On the other hand when test p-values are small, we make a decision depending on the distance of the test point to the classes. For the distance of the test point to the classes, we employ the distance between the mean of pixel values of the class and the test point. The detailed classification rule is given below:

- If PV1 and PV2 are test p-values in the hypothesis testings discussed in 2.3.1 and $\max(PV1, PV2) \geq 0.0001$(threshold), i.e., at least one of the test p-value is larger

Figure 2.6



(a) Image with training data and test points.



(b) Density plot of classes.

than the threshold value, then a test point x_0 belongs to class 1 or class 2 depending on $PV_1(1 - \text{prior of class 1})$ is smaller or greater than $PV_2(1 - \text{prior of class 2})$.

- If $\max(PV1, PV2) < 0.0001(\text{threshold})$, i.e., both the test p-values are smaller than the threshold value, then a test point x_0 belongs to class 1 if the distance of x_0 to the mean of class 1 is less than distance of x_0 to the mean of class 2. We classify x_0 as coming from the class 2 if the distance of x_0 to the mean of class 2 is less than distance of x_0 to the mean of class 1.

Next, we exemplify our modified method by applying it to the same image where Liao & Akritas's method failed to classify image pixels. As earlier, we choose pixels representing sky and vegetation as our class 1 and class 2 respectively and form training data and test points as shown in Figure 2.7(a). Classification of test points can be observed from Table 2.7 which shows that there are many misclassifications of test points by Laio & Akritas's method while the modified method classifies all the selected test points accurately. Hence, the modified method works well for the case of unequal priors.

Table 2.6: Classification by Liao-Akritas method in image Figure 2.6(a)

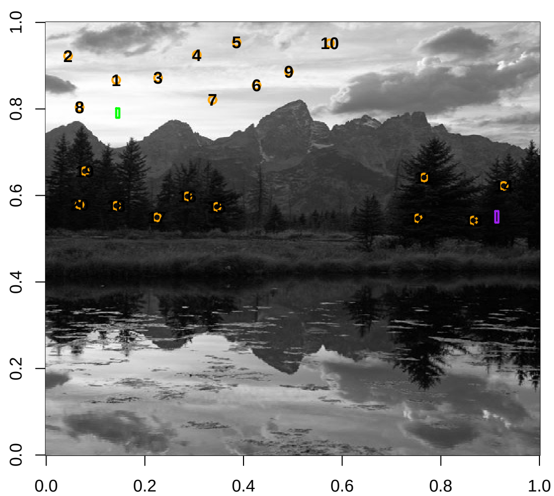
TP	LA	PV1	PV2	Pr1	Pr2	Obs
1	class 2	1.288141e-23	1.288141e-23	0.3376992	0.6623008	0.9803922
2	class 1	1.259595e-23	3.284515e-23	0.3376992	0.6623008	0.9882353
3	class 2	1.288141e-23	1.288141e-23	0.3376992	0.6623008	0.9803922
4	class 2	1.288141e-23	1.288141e-23	0.3376992	0.6623008	0.945098
5	class 2	1.288141e-23	1.288141e-23	0.3376992	0.6623008	0.9254902
6	class 2	1.288141e-23	1.288141e-23	0.3376992	0.6623008	0.9490196
7	class 2	1.288141e-23	1.288141e-23	0.3376992	0.6623008	0.8352941
8	class 2	1.288141e-23	1.288141e-23	0.3376992	0.6623008	0.9764706
9	class 1	1.275044e-23	6.554281e-23	0.3376992	0.6623008	0.9921569
10	class 2	1.288141e-23	1.288141e-23	0.3376992	0.6623008	0.9411765
11	class 2	1.288141e-23	1.288141e-23	0.3376992	0.6623008	0.945098
12	class 2	2.080374e-23	1.287129e-23	0.3376992	0.6623008	0.02745098
13	class 2	8.030967e-23	1.246122e-23	0.3376992	0.6623008	0.01960784
14	class 2	1.288141e-23	1.288141e-23	0.3376992	0.6623008	0.04313725
15	class 2	8.030967e-23	1.246122e-23	0.3376992	0.6623008	0.01960784
16	class 2	1.288141e-23	1.288141e-23	0.3376992	0.6623008	0.03921569
17	class 2	1.288141e-23	1.288141e-23	0.3376992	0.6623008	0.05882353
18	class 2	1.288141e-23	1.288141e-23	0.3376992	0.6623008	0.05882353
19	class 2	1.288141e-23	1.288141e-23	0.3376992	0.6623008	0.03921569
20	class 2	2.080374e-23	1.287129e-23	0.3376992	0.6623008	0.02745098

PVi= P-value for Test i, Pri= Prior probability of class i
LA= Liao & Akritas's method, Obs=Pixel of test point, TP= Test point.

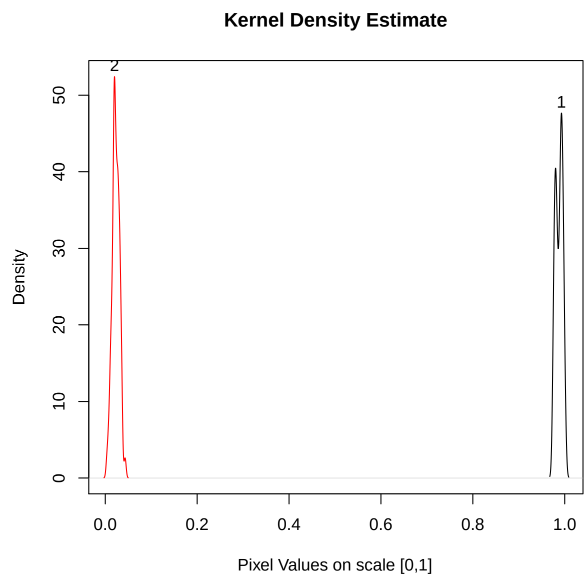
2.5 Classification of a block of pixels

After the classification of a test point, we would like to classify a part of the given image, i.e., a block of pixels. A block is a collection of many pixels in the image or it can be thought as a collection of many test points in the image. Then each of the test points is classified using the modified method of classification discussed in Section 2.4.

Figure 2.7



(a) Image with training data and test points.



(b) Density plot of classes.

Table 2.7: Comparison between Liao-Akritas and modified method in image Figure 2.7(a)

TP	LA	Obs	Our	Pr1	Pr2	PV1	PV2	d1	d2
1	class 1	0.97	class 1	0.46875	0.53125	9.701176e-23	1.285448e-22	0.011241	0.953460
2	class 2	0.94	class 1	0.46875	0.53125	9.749115e-23	1.070870e-22	0.038692	0.926009
3	class 2	0.93	class 1	0.46875	0.53125	9.749115e-23	1.070870e-22	0.054379	0.910323
4	class 2	0.86	class 1	0.46875	0.53125	9.749115e-23	1.070870e-22	0.124967	0.839734
5	class 2	0.82	class 1	0.46875	0.53125	9.749115e-23	1.070870e-22	0.164183	0.800519
6	class 2	0.96	class 1	0.46875	0.53125	9.749115e-23	1.070870e-22	0.019084	0.945617
7	class 2	0.89	class 1	0.46875	0.53125	9.749115e-23	1.070870e-22	0.093594	0.871107
8	class 1	1.00	class 1	0.46875	0.53125	9.735776e-23	1.566885e-21	0.012287	0.976989
9	class 2	0.92	class 1	0.46875	0.53125	9.749115e-23	1.070870e-22	0.058300	0.906401
10	class 2	0.82	class 1	0.46875	0.53125	9.749115e-23	1.070870e-22	0.160261	0.804440
11	class 2	0.06	class 2	0.46875	0.53125	9.749115e-23	1.070870e-22	0.924967	0.039734
12	class 2	0.02	class 2	0.46875	0.53125	4.191465e-22	1.061253e-22	0.964183	0.000519
13	class 2	0.03	class 2	0.46875	0.53125	1.584094e-22	1.064304e-22	0.956339	0.008362
14	class 2	0.08	class 2	0.46875	0.53125	9.749115e-23	1.070870e-22	0.905359	0.059342
15	class 2	0.03	class 2	0.46875	0.53125	1.584094e-22	1.064304e-22	0.956339	0.008362
16	class 2	0.07	class 2	0.46875	0.53125	9.749115e-23	1.070870e-22	0.909281	0.055420
17	class 2	0.05	class 2	0.46875	0.53125	9.749115e-23	1.070870e-22	0.932810	0.031891
18	class 2	0.03	class 2	0.46875	0.53125	1.021741e-22	1.070870e-22	0.948496	0.016205
19	class 2	0.07	class 2	0.46875	0.53125	9.749115e-23	1.070870e-22	0.913202	0.051499
20	class 2	0.03	class 2	0.46875	0.53125	1.146434e-22	1.068677e-22	0.952418	0.012283

PVi= P-value for Test i, di=distance of test point to the mean of class i,
Pri= Prior probability of class i, TP= Test points, Obs= Test point pixel
LA= Liao & Akritas's classification, Our= Modified method discussed in Section 2.4.

Chapter 3

Multiclass classification of image pixels.

In this chapter, we consider more than two classes of image pixels in the given image and focus on the classification of a randomly selected pixel in one of the defined classes. We begin with the extension of binary classification discussed in Section 2.1 to multiclass classification given by Liao and Akritas (2007). Then we apply their method of multiclass classification in the context of images. As was already discussed in Section 2.1, Liao and Akritas (2007) use hypotheses testings in their classification. In their classification, hypotheses testings are done as many times as the number of classes by placing the test observation in one of the classes every time. The null hypothesis then is that these new classes have the same distribution when the test observation is placed in only one of the classes and remaining classes are left intact. The p-values from these hypotheses testings which provide evidence to reject or fail to reject the null hypotheses, are the main tools of Liao and Akritas (2007) multiclass classification.

In our implementation of the Liao and Akritas (2007) for multiclass classification of image pixels, we observe that their classification method can completely fail to classify the image pixels when all the test p-values are small. Then we introduce a modified multiclass classification method which eliminates this drawback of the Liao & Akritas classification method. This modified classification is based on combined evidence from the minimum distance and the hypothesis testings. We consider both cases of equal and unequal prior probabilities of classes.

3.1 Multiclass classification of image pixels with equal priors using Liao and Akritas (2007).

In this section, we give a description of Liao & Akritas's multiclass classification with equal prior probabilities of classes in the case of images. We take a standard grey scale image of size 512×512. Then we define classes of pixels in the selected image and form training data for these classes by following the method discussed in Section 2.1.1.

Let $\pi_1, \pi_2, ..., \pi_k$ be the classes of pixels with class means $\mu_1, \mu_2, ..., \mu_k$ and prior probabilities $p_1, p_2, ..., p_k$. In this case, we have $p_1 = p_2 = ... = p_k = \frac{1}{k}$. Suppose that we have training data with observations $(x_{11}, x_{12}, ..., x_{1n_1})$, $(x_{21}, x_{22}, ..., x_{2n_2}), ..., (x_{k1}, x_{k2}, ..., x_{kn_k})$ from the classes $\pi_1, \pi_2, ..., \pi_k$ respectively. Next, we take a randomly selected pixel value x_0 which we would like to classify in one of the classes. For the classification of x_0, we perform a series of hypothesis testings in which we test to see the sample evidence that the observation x_0 belongs to each of the classes based on the training data. For this purpose, we need to determine the statistical tests to perform. The choice of statistical tests for the hypothesis testings mainly depends on the form of the distributions of pixel values representing classes. We note that the pixel values in some classes may behave like normal distributions where as others may not. So, we choose Kruskal-Wallis and ANOVA as our statistical tests. We use Kruskal-Wallis test when the pixel values do not follow normal distribution. Kruskal-Wallis test does not assume a normal population and is used to test the equality of class distributions. On the other hand, when pixel values appear reasonably normal, we use ANOVA which provides a statistical test of whether or not means of several classes are equal. Let f_i and F_i denote the probability density and cumulative density function of the class π_i. The k tests are as follows:

- Test 1: Place x_0 with the observation from class 1 and assume,
  $(x_0, x_{11}, x_{12}, ..., x_{1n_1}) \sim f_1(x)$
  $(x_{21}, x_{22}, ..., x_{2n_2}) \sim f_2(x)$
  $\vdots$
  $(x_{k1}, x_{k2}, ..., x_{kn_k}) \sim f_k(x)$
  to test the null hypothesis, H_0. The H_0 for Kruskal-Wallis test is that all the distribution functions are identical and H_0 for ANOVA is $\mu_1 = \mu_2 = \ldots = \mu_k$.

- Test 2: Place x_0 with the observation from class 2 and assume,
  $(x_{11}, x_{12}, ..., x_{1n_1}) \sim f_1(x)$
  $(x_0, x_{21}, x_{22}, ..., x_{2n_2}) \sim f_2(x)$
  $\vdots$
  $(x_{k1}, x_{k2}, ..., x_{kn_k}) \sim f_k(x)$
  to test the null hypothesis, H_0. The H_0 for Kruskal-Wallis test is that all the distribution functions are identical and H_0 for ANOVA is $\mu_1 = \mu_2 = \ldots = \mu_k$.
  and similarly,

- Test k: Place x_0 with the observation from class k and assume,
  $(x_{11}, x_{12}, ..., x_{1n_1}) \sim f_1(x)$
  $(x_{21}, x_{22}, ..., x_{2n_2}) \sim f_2(x)$
  $\vdots$
  $(x_0, x_{k1}, x_{k2}, ..., x_{kn_k}) \sim f_k(x)$
  to test the null hypothesis, H_0. The H_0 for Kruskal-Wallis test is that all the distribution functions are identical and H_0 for ANOVA is $\mu_1 = \mu_2 = \ldots = \mu_k$.

Let $PV_1(x_0), PV_2(x_0), ..., \& PV_k(x_0)$ denote the p-values of the Test 1, Test 2,...., & Test k. Then the multiclass classification introduced by Liao and Akritas (2007) can be described in the following four steps:

- Step 1: We obtain p-values $PV_1(x_0), PV_2(x_0), ..., PV_k(x_0)$ from Test 1, Test 2,....,Test k as discussed above.

- Step 2: We compare the p-values and eliminate the class with the largest $(1 - \frac{1}{k}) \times PV_i(x_0)$.

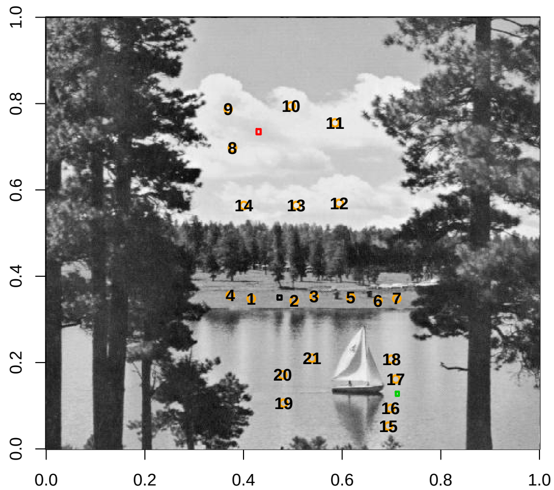
- Step 3: We repeat Step 1 and Step 2 until there are two classes of pixels left.

- Step 4: We then repeat Step 1 for the remaining two classes and we classify x_0 as the one with the smaller $(1 - \frac{1}{k}) \times PV_i(x_0)$ which is exactly the binary classification discussed in Section 2.1.

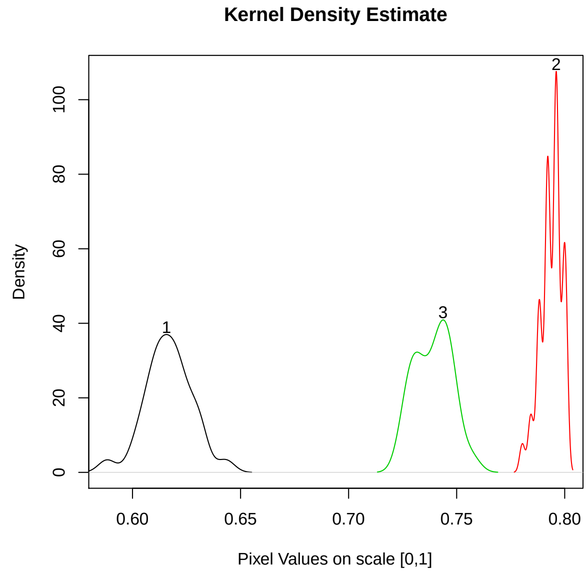
Now, we apply the Liao & Akritas multiclass classification to a specific image, as given in Figure 3.1(a). Let us consider three different classes in the given 512×512 image. We choose regions with different levels of gray-scale to form distinct classes. We choose grass region as our first class (class 1), cloud region (white color) as our second class (class 2) and water region (grey color) as our third class (class 3). Then, using the method discussed in Section 2.1.1, we form training data for the classes. We use these training data to estimate the class parameters. We select 21 test points labeled with numbers in such a way that first seven of them are chosen from regions representing class 1, next seven test points are from class 2 and the rest are from the class 3 regions as shown in Figure 3.1(a). Density plot of the classes are shown in the Figure 3.1(b) in which 1, 2, and 3 denote density plot of classes 1, 2 and 3 respectively. Density plot shows that classes so formed are distinct and separated. Then using decision rule discussed earlier, we classify selected test points. Classification of the selected test points are given in Table 3.1 and all the test points are classified correctly.

We observe that the Liao and Akritas (2007) multiclass classification is applicable to classify image pixels when all the test p-values from the hypotheses testings are not very small. Their method of classification does not work when the p-values are small. With the image shown in Figure 3.2, we show the failure of their classification method. Figure 3.2(a) is an image in which we choose three regions with different levels of gray-scale to form classes. We choose sky region as our class 1, mountain region as class 2, and vegetation

Figure 3.1



(a) Image with training data and test points.



(b) Density plot of classes.

as class 3 and form training data for the classes. The classes so formed are distinct and well separated as shown by the their density plot. For classification purpose, we select 21 observational points as shown in the Figure 3.2(a).

Classification of these selected test points are shown in Table 3.2 in which we see that Liao & Akritas's method misclassifies test points 7-14. Hence, we have many misclassification of test points.

The reason behind these misclassifications are the p-values which are very small as shown in the table. The p-values in a valid test, theoretically follow the uniform $(0, 1)$ distribution so that two p-values that are both smaller than the level of significance do not provide different level of evidence to reject the null hypothesis. Consequently, Liao & Akritas's method fails when all the p-values are small. Hence, we conclude that Liao & Akritas's multiclass classification is not applicable in the context of images when the test p-values are small.

Table 3.1: Classification by Laio-Akritas method in image Figure 3.1(a)

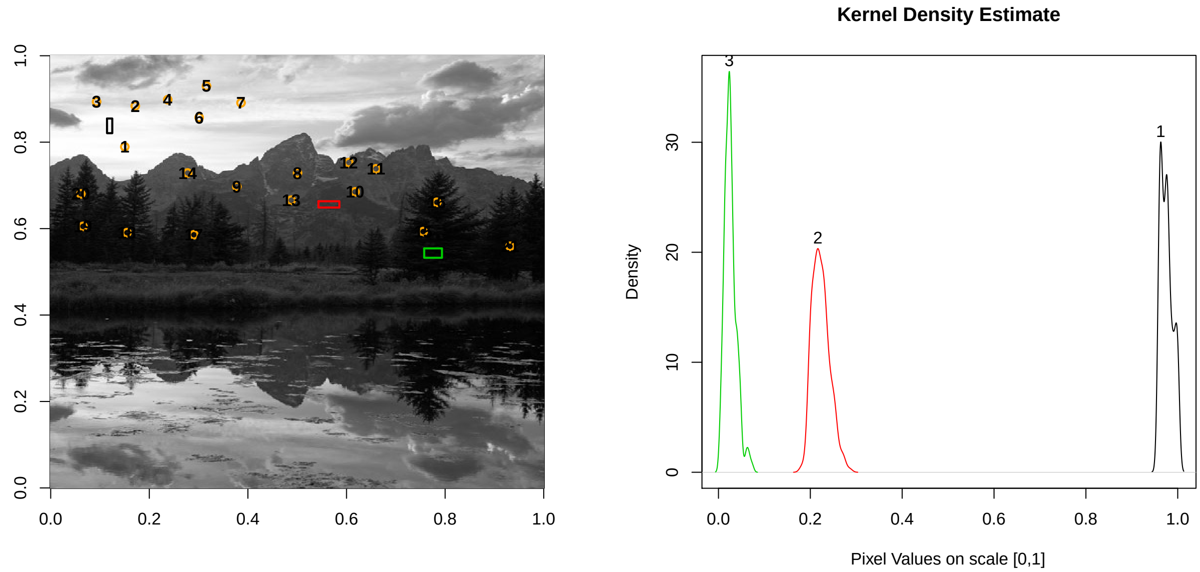
TP	PV1	PV2	PV3	LA	Obs
1	5.383e-19	1.426e-17	1.331e-18	class 1	0.6000000
2	5.376e-19	2.958e-18	6.612e-19	class 1	0.6235294
3	5.362e-19	3.991e-18	7.557e-19	class 1	0.6196078
4	5.376e-19	2.026e-18	5.565e-19	class 1	0.6313725
5	5.385e-19	1.480e-17	1.353e-18	class 1	0.5921569
6	5.376e-19	1.058e-17	1.167e-18	class 1	0.6078431
7	5.370e-19	5.604e-18	8.805e-19	class 1	0.6156863
8	2.752e-17	6.188e-19	2.840e-18	class 2	0.8705882
9	2.752e-17	6.188e-19	2.840e-18	class 2	0.8509804
10	2.752e-17	6.188e-19	2.840e-18	class 2	0.8705882
11	2.752e-17	6.188e-19	2.840e-18	class 2	0.8784314
12	2.752e-17	6.188e-19	2.840e-18	class 2	0.8549020
13	2.752e-17	6.188e-19	2.840e-18	class 2	0.8823529
14	2.752e-17	6.188e-19	2.840e-18	class 2	0.8235294
15	1.212e-18	6.188e-19	5.110e-19	class 3	0.7725490
16	8.389e-19	9.805e-19	5.096e-19	class 3	0.7411765
17	1.212e-18	6.188e-19	5.110e-19	class 3	0.7764706
18	1.409e-18	6.183e-19	5.558e-19	class 3	0.7843137
19	5.571e-19	1.610e-18	5.106e-19	class 3	0.7254902
20	5.385e-19	1.678e-18	5.110e-19	class 3	0.7098039
21	7.594e-19	1.113e-18	5.106e-19	class 3	0.7372549

PVi= p-value from test i, TP=Test point, Obs= Test point pixel
LA=Liao & Akritas's method.

3.2 Multiclass classification based on combined evidence from minimum distance and hypothesis tests.

In the last section, we observe that Liao & Akritas multiclass classification fails when the hypotheses tests produce small p-values. In this section, we propose a modified multiclass classification method which also works in the case of small p-values from the hypotheses tests described in Section 3.1. We discuss multiclass classification rules for the modified

Figure 3.2



(a) Image with training data and test points.

(b) Density plot of classes.

classification method and employ this method in the context of images. Here, the classification is based on both the p-values obtained from the hypothesis testings and distance of test observation from the classes. When all the test p-values are large (larger than some threshold value), we follow the decision rule given by Liao-Akritas (2007) as discussed in Section 3.1. When at least one of the test p-values is small (smaller than the threshold), we use the distance of the observation to the classes in our decision rule. The detailed classification rule for the multiclass classification based on the combined evidence from minimum distance and hypotheses tests are as follows:

- If $\max_{1\leq i\leq k} PV_i \leq 0.0001$ (threshold), i.e., all the test p-values are small, then we use minimum distance classification. We assign x_0 to the class with the smallest d_i where $d_i = \min_{1\leq i\leq k} D_i$ and D_i is the distance of the observation x_0 to the class π_i.

- If $\min_{1\leq i\leq k} PV_i \geq 0.0001$ (threshold), i.e., all the test p-values are large, then the decision function follows the Liao & Akritas's classification method discussed in Section 3.1.

- If m $(1 < m < k)$ of the test p-values are less than or equal to the threshold (0.0001),

Table 3.2: Classification by Laio-Akritas method in image Figure 3.2(a)

TP	PV1	PV2	PV3	LA	Obs
1	3.228e-121	5.621e-121	6.093e-120	class 1	0.97647059
2	3.233e-121	3.758e-121	2.358e-120	class 1	0.86666667
3	3.233e-121	3.758e-121	2.358e-120	class 1	0.80784314
4	3.233e-121	3.758e-121	2.358e-120	class 1	0.86274510
5	3.233e-121	3.758e-121	2.358e-120	class 1	0.84705882
6	3.233e-121	3.779e-121	2.388e-120	class 1	0.95686275
7	3.233e-121	3.758e-121	2.358e-120	class 1	0.92941176
8	3.242e-121	3.758e-121	2.349e-120	class 1	0.28627451
9	3.646e-121	3.757e-121	2.004e-120	class 1	0.25098039
10	4.413e-121	3.757e-121	1.547e-120	class 2	0.23529412
11	3.251e-121	3.758e-121	2.340e-120	class 1	0.28235294
12	3.304e-121	3.758e-121	2.289e-120	class 1	0.27058824
13	3.233e-121	3.758e-121	2.358e-120	class 1	0.29803922
14	3.233e-121	3.758e-121	2.358e-120	class 1	0.32156863
15	1.183e-119	1.408e-120	3.921e-121	class 3	0.01960784
16	2.438e-120	5.652e-121	3.929e-121	class 3	0.03529412
17	1.352e-120	4.018e-121	3.929e-121	class 3	0.05490196
18	2.062e-120	5.128e-121	3.925e-121	class 3	0.03921569
19	1.378e-120	4.063e-121	3.929e-121	class 3	0.05098039
20	1.204e-120	3.758e-121	3.929e-121	class 2	0.09411765
21	2.438e-120	5.652e-121	3.929e-121	class 3	0.03529412

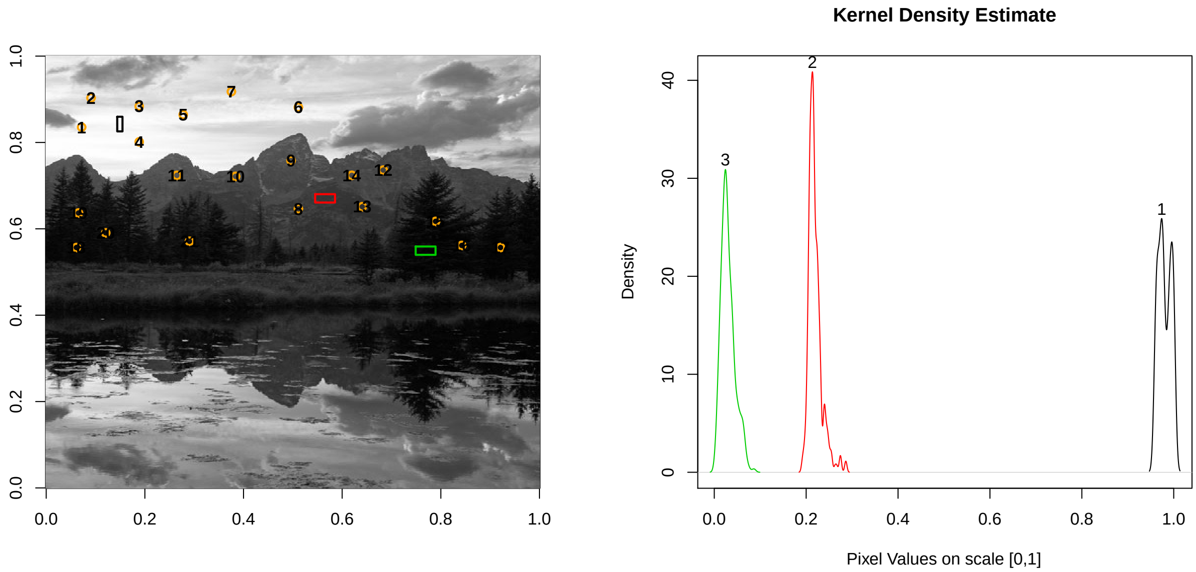
PVi= p-value from test i, TP=Test point, Obs= Test point pixel
LA=Liao & Akritas's method.

then we eliminate $k - m$ classes which have p-values larger than the threshold. We use the minimum distance method to determine suitable class from these m classes.

- If $m = 1$, we assign the observation to that class with the p-value less than the threshold.

Now, we employ this modified classification method in the same image where the Liao & Akritas classification was observed to fail in Section 3.1. We define 3 classes as it was defined earlier in the image 3.2(a) and form training data. Figure 3.3(a) shows the image with training data and some test points. We classify these selected test points by employing Liao and Akritas (2007) classification and modified classification method. The classification results of both methods are shown in Table 3.3 which also shows the test p-values and distances of the test points to their corresponding classes. We observe that the modified method classifies all the test points accurately while the Liao & Akritas's method have some misclassifications.

Figure 3.3



(a) Image with training data and test points.

(b) Density plot of classes.

Finally, we employ the modified classification method in pepper image and define three distinct classes in it. Figure 3.4(a) shows the training data for three image classes along with some test points. Figure 3.4(b) shows that the classes so formed are well separated and distinct. Employing the modified method and Liao & Akritas's method, we classify the selected test points and are shown in Table 3.4. From the table, we observe that the modified classification method based on the combined evidence from minimum distance and hypotheses tests provides an accurate multiclass classification of image pixels.

Table 3.3: Comparison between Liao-Akritas and modified methods in image Figure 3.3(a)

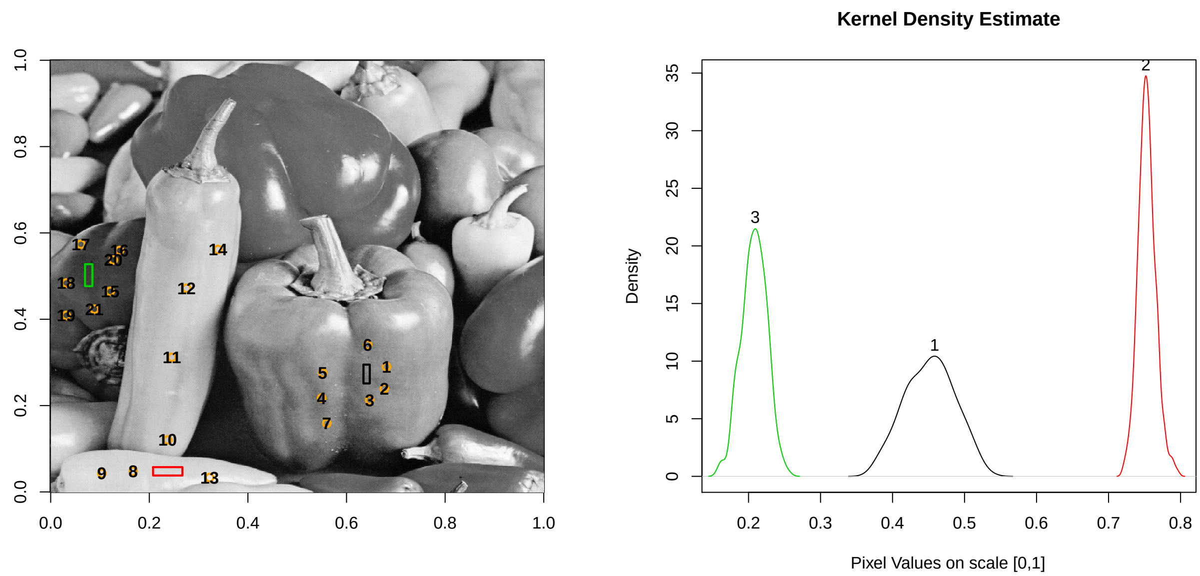
TP	LA	PV1	PV2	PV3	d1	d2	d3	Our	Obs
1	class 1	6.488e-128	7.816e-128	5.420e-127	0.027	0.733	0.922	class 1	0.949
2	class 1	6.488e-128	7.816e-128	5.420e-127	0.118	0.643	0.831	class 1	0.858
3	class 1	6.488e-128	7.816e-128	5.420e-127	0.106	0.655	0.843	class 1	0.870
4	class 1	6.486e-128	1.217e-127	1.530e-126	0.012	0.773	0.961	class 1	0.988
5	class 1	6.488e-128	7.816e-128	5.420e-127	0.02	0.741	0.929	class 1	0.956
6	class 1	6.488e-128	7.816e-128	5.420e-127	0.035	0.725	0.914	class 1	0.941
7	class 1	6.488e-128	7.816e-128	5.420e-127	0.145	0.616	0.804	class 1	0.831
8	class 1	6.732e-128	7.816e-128	5.159e-127	0.718	0.043	0.231	class 2	0.258
9	class 1	6.488e-128	7.816e-128	5.420e-127	0.671	0.09	0.278	class 2	0.305
10	class 1	6.488e-128	7.816e-128	5.420e-127	0.667	0.094	0.282	class 2	0.309
11	class 1	6.488e-128	7.816e-128	5.420e-127	0.631	0.129	0.318	class 2	0.345
12	class 1	6.714e-128	7.816e-128	5.177e-127	0.714	0.047	0.235	class 2	0.262
13	class 2	2.605e-127	7.815e-128	8.273e-128	0.776	0.016	0.173	class 2	0.200
14	class 1	6.679e-128	7.816e-128	5.214e-127	0.71	0.051	0.239	class 2	0.266
15	class 3	3.907e-127	9.640e-128	7.815e-128	0.925	0.165	0.024	class 3	0.050
16	class 3	3.298e-126	3.298e-127	7.803e-128	0.957	0.196	0.008	class 3	0.019
17	class 3	1.377e-126	1.991e-127	7.793e-128	0.949	0.188	0.000	class 3	0.027
18	class 3	5.858e-127	1.217e-127	7.802e-128	0.937	0.176	0.012	class 3	0.039
19	class 3	2.767e-127	7.900e-128	7.816e-128	0.902	0.141	0.047	class 3	0.074
20	class 3	4.338e-127	1.024e-127	7.814e-128	0.929	0.169	0.020	class 3	0.047
21	class 3	3.072e-127	8.391e-128	7.814e-128	0.914	0.153	0.035	class 3	0.062

PVi= P-value from test i, di=distance of test pt from mean of class i, TP= Test point, LA= Liao & Akritas's method , Our= Modified method, Obs= Test point pixel.

3.3 Multiclass classification of image pixels with unequal prior probabilities of classes, using Liao and Akritas (2007).

This section illustrates Liao and Akritas (2007) multiclass classification in the context of unequal prior probabilities of classes. The idea of classification in this case is similar to the

Figure 3.4



(a) Image with training data and test points.

(b) Density plot of classes.

case of equal priors as discussed in Section 3.1.

We begin by taking a standard grey-scale image of size 512×512 and forming training data for the classes as described in Section 2.1.1. Let $\pi_1, \pi_2, ..., \pi_k$ be the classes of image pixels and let μ_i and p_i be the mean and prior probability of the class π_i respectively for $i = 1, 2, \ldots, k$. Unequal prior probability means that these classes are not equally likely. So, we have $p_i \neq p_j$ for some i, j where $1 \leq i, j \leq n$. In order to define the prior of classes, we first order the class-means. Let $\mu_1, \mu_2, ..., \mu_k$ be the ordered means of the classes, i.e., $\mu_1 \leq \mu_2 \leq ... \leq \mu_k$. We define the prior probability of class i as follows:

$$\text{Prior of class i} = \frac{\text{Number}\left(\dfrac{\mu_{(i-1)} + \mu_{(i)}}{2} < \text{pixels in training data} < \dfrac{\mu_{(i)} + \mu_{(i+1)}}{2}\right)}{\text{Number of pixels in training data.}} \tag{3.3.1}$$

This means, the prior of class i is the ratio of the number of pixels greater than the average of means of classes (i-1)and i, and smaller than the average of means of classes i and (i+1) entry-wise to the total number of pixels in the given image.

Table 3.4: Comparison between Liao-Akritas and modified methods in image Figure 3.4(a)

TP	LA	PV1	PV2	PV3	d1	d2	d3	Our	Obs
1	class 1	1.239e-135	1.971e-135	2.265e-135	0.027	0.271	0.275	class 1	0.482
2	class 1	1.239e-135	1.691e-135	2.533e-135	0.059	0.239	0.306	class 1	0.513
3	class 1	1.239e-135	3.021e-135	1.656e-135	0.02	0.318	0.227	class 1	0.435
4	class 1	1.239e-135	4.041e-135	1.336e-135	0.071	0.369	0.176	class 1	0.384
5	class 1	1.239e-135	1.638e-135	2.592e-135	0.078	0.22	0.325	class 1	0.533
6	class 1	1.239e-135	2.028e-135	2.217e-135	0.024	0.275	0.271	class 1	0.478
7	class 1	1.239e-135	1.871e-135	2.353e-135	0.035	0.263	0.282	class 1	0.490
8	class 2	3.469e-135	1.623e-135	1.555e-134	0.298	0.000	0.545	class 2	0.752
9	class 2	3.469e-135	1.623e-135	1.555e-134	0.298	0.000	0.545	class 2	0.752
10	class 1	1.239e-135	1.633e-135	2.598e-135	0.259	0.039	0.506	class 2	0.713
11	class 1	1.239e-135	1.633e-135	2.598e-135	0.231	0.067	0.478	class 2	0.686
12	class 1	1.239e-135	1.633e-135	2.598e-135	0.255	0.043	0.502	class 2	0.709
13	class 2	9.380e-135	1.633e-135	8.733e-134	0.373	0.075	0.62	class 2	0.827
14	class 2	2.035e-135	1.628e-135	6.153e-135	0.29	0.008	0.537	class 2	0.745
15	class 1	1.293e-135	4.571e-135	1.313e-135	0.22	0.518	0.027	class 3	0.235
16	class 3	2.911e-135	3.119e-134	1.313e-135	0.275	0.573	0.027	class 3	0.180
17	class 1	1.239e-135	4.135e-135	1.313e-135	0.129	0.427	0.118	class 3	0.325
18	class 1	1.239e-135	4.135e-135	1.313e-135	0.169	0.467	0.078	class 3	0.286
19	class 1	1.239e-135	4.135e-135	1.313e-135	0.188	0.486	0.059	class 3	0.266
20	class 3	1.982e-135	1.255e-134	1.312e-135	0.247	0.545	0	class 3	0.207
21	class 1	1.239e-135	4.135e-135	1.313e-135	0.192	0.49	0.055	class 3	0.262

PVi= P-value from test i, di=distance of test pt from mean of class i, TP= Test point, LA= Liao & Akritas's method , Our= Modified method, Obs= Test point pixel.

Another way to define prior probabilities of classes as are follows: Prior probability of i^{th} class :

$$\text{Prior of class i} = \frac{N_i}{N_1 + \ldots + N_k}$$

where N_i represents the number of pixel values in the training data for the i^{th} class and k is the total number of classes in the image.

Suppose that the training vectors for the classes $\pi_1, \pi_2, ..., \pi_k$ are given by $(x_{11}, x_{12}, ..., x_{1n_1})$,

$(x_{21}, x_{22}, ..., x_{2n_2})$, ..., $(x_{k1}, x_{k2}, ..., x_{kn_k})$, respectively. Let x_0 denote a randomly selected pixel in the given image. Then a series of hypotheses tests similar to the equal prior case discussed in Section 3.1 is performed in order to classify the test point x_0. These hypotheses testings are done as many times as the number of classes by assuming that the test point belongs to one of the classes each time. Each of these hypotheses testings will test that the new classes have same the distribution under the assumption that the test point is in one of the classes while the rest of the classes are kept as they are. Regarding the statistical tests, when image pixels are normally distributed, we chose the ANOVA test whereas Kurskal-Wallis test is preferred in the case of non-normal distributions in the hypotheses testings. Then similar to the equal prior case, we perform Test 1, Test 2,... ,and Test k as described in Section 3.1 which will provide us k p-values. Let us denote the p-values of these tests by $PV_1(x_0), PV_2(x_0), ..., \text{and} PV_k(x_0)$ respectively. Next, we summarize the Liao & Akritas's multiclass classification with unequal prior probabilities in the following four steps:

- Step 1: We obtain p-values $PV_1(x_0), PV_2(x_0), ..., PV_k(x_0)$ from Test 1, Test 2,....,Test k as discussed above.

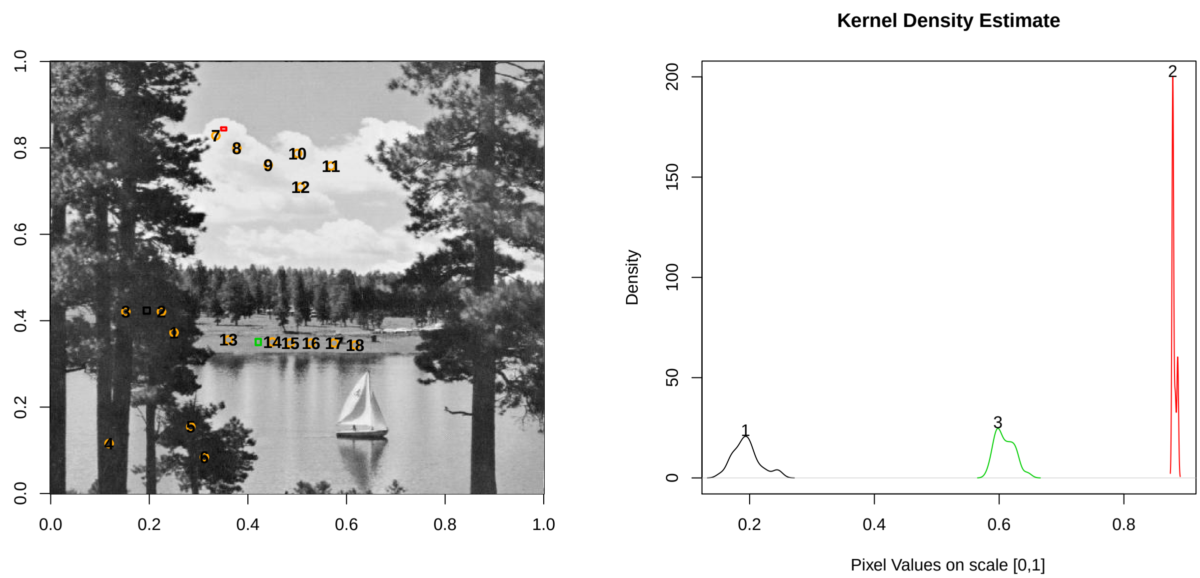
- Step 2: We compare the p-values and eliminate the class with the largest $(1 - p_i) \times PV_i(x_0)$.

- Step 3: We repeat Step 1 and Step 2 until there are two classes of pixels left.

- Step 4: We then repeat Step 1 for the remaining two classes and we classify x_0 as the one with the smaller $(1 - p_i) \times PV_i(x_0)$.

Now, we employ this multiclass classification of Liao & Akritas with unequal priors of classes to an image as shown in Figure 3.5(a). We define three classes in the given image depending on the gray-scale levels. We choose tree region, sky region, and grass region for class 1, class 2 and class 3 respectively and form training data accordingly. These classes are distinct and well separated with each other as shown by Figure 3.5(b) and Figure 3.5(a) shows the training data and some test points for the classification. After computing class means from the training data, prior probabilities of these classes are obtained by using the formula given in (3.3.1). Then the classification of these test points by the Liao & Akritas's method discussed above are shown in Table 3.5, which shows the classification result of the selected test points along with p-values of the test and prior probabilities of the classes. All the test points are classified accurately except test point 7.

Figure 3.5



(a) Image with training data and test points.

(b) Density plot of classes.

As in the case of equal prior case, the Liao & Akritas's classification does not classify image pixels correctly when all the test p-values are small. This is shown in the classification of the test points considered in the image in Figure 3.6 where we define four classes of image pixels depending on the levels of gray-scale in the image. Then we calculate the prior probabilities of the classes as defined by the formula earlier. The selected test points are classified by employing Liao & Akritas's method and are tabulated in Table 3.6. We observe that half of the test points are misclassified. The main reasons for these misclassifications are the small p-values which are used to reject or fail to reject the null hypothesis. These p-values theoretically follow the uniform $(0, 1)$ distribution so that the small p-values (smaller than the significance level) are not helpful in distinguishing among the classes. So, from

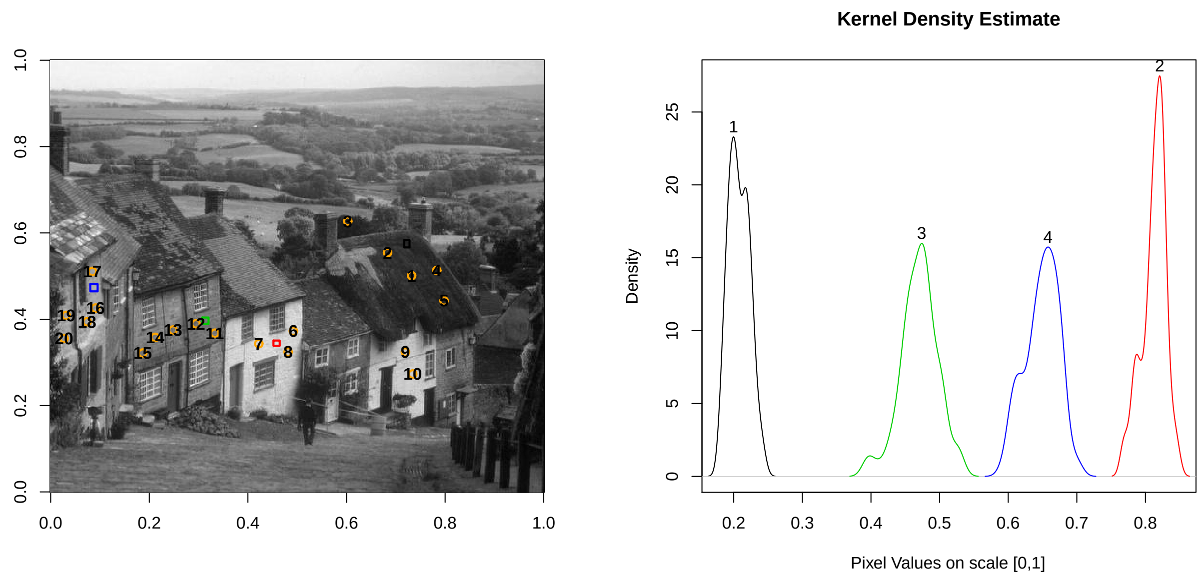
Table 3.5: Classification by Laio-Akritas method in image Figure 3.5(a)

TP	LA	PV1	PV2	PV3	Pr1	Pr2	Pr3
1	class 1	3.618e-27	2.380e-25	2.328e-26	0.4472618	0.1961899	0.3565292
2	class 1	3.618e-27	2.380e-25	2.328e-26	0.4472618	0.1961899	0.3565292
3	class 1	3.615e-27	1.798e-26	4.988e-27	0.4472618	0.1961899	0.3565292
4	class 1	3.618e-27	1.366e-26	4.233e-27	0.4472618	0.1961899	0.3565292
5	class 1	3.618e-27	2.380e-25	2.328e-26	0.4472618	0.1961899	0.3565292
6	class 1	3.618e-27	2.650e-25	2.481e-26	0.4472618	0.1961899	0.3565292
7	class 3	7.760e-26	2.867e-27	5.928e-27	0.4472618	0.1961899	0.3565292
8	class 2	1.738e-26	2.875e-27	3.194e-27	0.4472618	0.1961899	0.3565292
9	class 2	1.738e-26	2.875e-27	3.194e-27	0.4472618	0.1961899	0.3565292
10	class 2	1.738e-26	2.875e-27	3.194e-27	0.4472618	0.1961899	0.3565292
11	class 2	1.738e-26	2.875e-27	3.194e-27	0.4472618	0.1961899	0.3565292
12	class 2	1.738e-26	2.875e-27	3.194e-27	0.4472618	0.1961899	0.3565292
13	class 3	1.603e-26	3.047e-27	3.194e-27	0.4472618	0.1961899	0.3565292
14	class 3	3.618e-27	8.536e-27	3.194e-27	0.4472618	0.1961899	0.3565292
15	class 3	9.082e-27	4.558e-27	3.190e-27	0.4472618	0.1961899	0.3565292
16	class 3	7.471e-27	5.228e-27	3.193e-27	0.4472618	0.1961899	0.3565292
17	class 3	4.334e-27	7.549e-27	3.188e-27	0.4472618	0.1961899	0.3565292
18	class 3	1.140e-26	3.883e-27	3.190e-27	0.4472618	0.1961899	0.3565292

PVi= P-value from test i; Pri=Prior probability class i
LA= Liao & Akritas's method; TP= Test point.

this illustration, we conclude that in the case of unequal prior probabilities of classes, Liao & Akritas multiclass classification does not work well when all the test p-values are small. Next, we introduce a modified classification method which eliminates this drawback of the Liao & Akrita's method.

Figure 3.6



(a) Image with training data and test points.

(b) Density plot of classes.

3.4 Multiclass classification, with unequal priors, based on combined evidence from minimum distance and hypothesis tests.

In this section, we give a modified multiclass classification method for unequal priors. The decision rule for the classification in this modified method is similar to that of the modified method discussed in Section 3.2. It depends both on the p-values from the hypotheses testings and distances of the test points to the defined classes. Use of the distances in the decision rule will overcome the drawback of the Liao & Akritas's classification method when all the p-values from the hypotheses testings are small. The detailed classification rule for the modified method in the context of unequal priors of the classes based on the combined evidence from minimum distance and hypotheses tests are as follows:

- If $\max\limits_{1\le i\le k} PV_i \le 0.0001$ (threshold), i.e., all the test p-values are small, then we use minimum distance classification. We assign x_0 to the class with the smallest d_i where $d_i = \min\limits_{1\le i\le k} D_i$ and D_i is the distance of the observation x_0 to the class π_i.

Table 3.6: Classification by Laio-Akritas method in image Figure 3.6(a)

TP	LA	PV1	PV2	PV3	PV4	Pr1	Pr2	Pr3	Pr4
1	class 1	1.125e-59	7.776e-58	1.986e-59	8.672e-59	0.33155	0.09981	0.44617	0.12247
2	class 1	1.126e-59	1.343e-58	1.148e-59	2.685e-59	0.33155	0.09981	0.44617	0.12247
3	class 1	1.125e-59	8.921e-58	2.073e-59	9.505e-59	0.33155	0.09981	0.44617	0.12247
4	class 1	1.125e-59	9.922e-58	2.142e-59	1.020e-58	0.33155	0.09981	0.44617	0.12247
5	class 1	1.126e-59	1.071e-57	2.194e-59	1.074e-58	0.33155	0.09981	0.44617	0.12247
6	class 4	1.647e-58	1.110e-59	3.378e-59	1.258e-59	0.33155	0.09981	0.44617	0.12247
7	class 4	2.105e-58	1.110e-59	4.004e-59	1.369e-59	0.33155	0.09981	0.44617	0.12247
8	class 4	1.502e-58	1.110e-59	3.171e-59	1.220e-59	0.33155	0.09981	0.44617	0.12247
9	class 4	1.391e-58	1.110e-59	3.007e-59	1.189e-59	0.33155	0.09981	0.44617	0.12247
10	class 4	1.526e-58	1.110e-59	3.205e-59	1.226e-59	0.33155	0.09981	0.44617	0.12247
11	class 1	1.515e-59	6.684e-59	1.131e-59	1.861e-59	0.33155	0.09981	0.44617	0.12247
12	class 1	1.404e-59	7.912e-59	1.131e-59	2.031e-59	0.33155	0.09981	0.44617	0.12247
13	class 1	1.257e-59	1.007e-58	1.131e-59	2.300e-59	0.33155	0.09981	0.44617	0.12247
14	class 1	2.173e-59	2.994e-59	1.131e-59	1.228e-59	0.33155	0.09981	0.44617	0.12247
15	class 1	1.748e-59	4.870e-59	1.131e-59	1.580e-59	0.33155	0.09981	0.44617	0.12247
16	class 1	2.400e-59	2.709e-59	1.175e-59	1.188e-59	0.33155	0.09981	0.44617	0.12247
17	class 4	3.229e-59	2.338e-59	1.378e-59	1.189e-59	0.33155	0.09981	0.44617	0.12247
18	class 4	7.742e-59	1.502e-59	2.201e-59	1.188e-59	0.33155	0.09981	0.44617	0.12247
19	class 1	2.235e-59	2.809e-59	1.131e-59	1.189e-59	0.33155	0.09981	0.44617	0.12247
20	class 1	2.235e-59	2.809e-59	1.131e-59	1.189e-59	0.33155	0.09981	0.44617	0.12247

PVi= P-value from test i, Pri=Prior probability class i
LA= Liao & Akritas's method, TP= Test point.

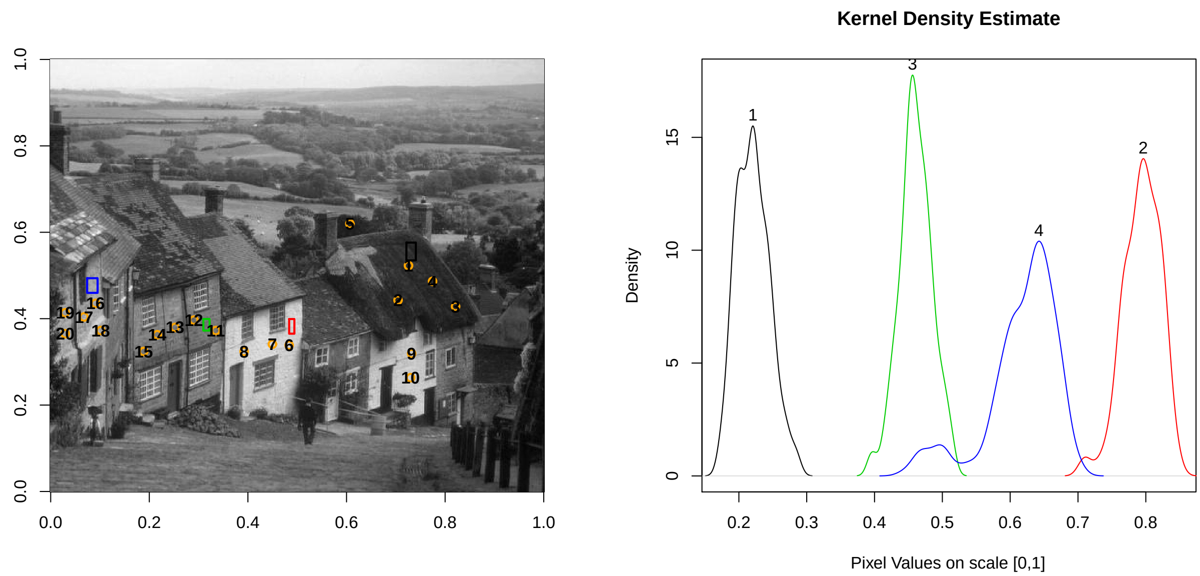
- If $\min_{1 \leq i \leq k} PV_i \geq 0.0001$ (threshold), i.e., all the test p-values are large, then the decision function follows the Liao & Akritas's classification method discussed in Section 3.3.

- If m $(1 < m < k)$ of the test p-values are less than or equal to the threshold (0.0001), then we eliminate $k - m$ classes which have p-values larger than the threshold. We use the minimum distance method to determine suitable class from these m classes.

- If $m = 1$, we assign the observation to that class with the p-value less than the threshold.

Now, we apply this modified method to the image where Liao & Akritas's method failed to classify image pixels. We choose classes as it was chosen in Figure 3.7(a) and form training data. As before, we choose some test points representing the classes. Figure 3.7(b) shows that defined class 4 has slight overlap with class 3 and 2. The prior probabilities of classes are shown in Table 3.7.

Figure 3.7



(a) Image with training data and test points.

(b) Density plot of classes.

Table 3.7: Prior probabilities of classes in image Figure 3.7(a)

Prior1	Prior2	Prior3	Prior4
0.3315468	0.1094818	0.4273758	0.1315956

The classification of these test points by this modified method are shown in Table 3.8. The table also shows the classification given by Liao & Akritas's method and distances of the test points to the classes. All the selected test points are classified accurately by the modified method while there are some misclassifications by Liao & Akritas.

Table 3.8: Classification of test points in image Figure 3.7(a)

TP	LA	Our	PV1	PV2	PV3	PV4	d1	d2	d3	d4
1	class 1	class 1	2.7e-135	2.8e-134	2.9e-135	6.3e-135	0.023	0.552	0.215	0.388
2	class 1	class 1	2.7e-135	1.4e-133	5.2e-135	1.9e-134	0.015	0.592	0.254	0.427
3	class 1	class 1	2.7e-135	4.8e-134	3.5e-135	9.1e-135	0.007	0.568	0.231	0.403
4	class 1	class 1	2.7e-135	4.8e-134	3.5e-135	9.1e-135	0.007	0.568	0.231	0.403
5	class 1	class 1	2.7e-135	3.2e-133	6.9e-135	3.2e-134	0.039	0.615	0.278	0.450
6	class 4	class 2	4.7e-134	2.3e-135	7.7e-135	3.0e-135	0.556	0.019	0.317	0.145
7	class 4	class 2	1.0e-133	2.3e-135	1.3e-134	4.0e-135	0.603	0.027	0.364	0.192
8	class 4	class 2	3.3e-134	2.3e-135	6.1e-135	2.7e-135	0.505	0.070	0.266	0.094
9	class 4	class 2	5.1e-134	2.3e-135	8.1e-135	3.1e-135	0.564	0.011	0.325	0.152
10	class 4	class 2	7.8e-134	2.3e-135	1.0e-134	3.6e-135	0.584	0.007	0.345	0.172
11	class 1	class 3	4.0e-135	8.3e-135	2.4e-135	3.1e-135	0.250	0.325	0.011	0.160
12	class 1	class 3	3.9e-135	8.8e-135	2.4e-135	3.2e-135	0.247	0.329	0.007	0.164
13	class 1	class 3	4.2e-135	7.8e-135	2.4e-135	3.0e-135	0.254	0.321	0.015	0.156
14	class 1	class 3	4.9e-135	6.3e-135	2.5e-135	2.8e-135	0.278	0.298	0.039	0.133
15	class 1	class 3	3.0e-135	1.3e-134	2.4e-135	3.9e-135	0.223	0.352	0.015	0.188
16	class 4	class 4	1.5e-134	3.5e-135	4.2e-135	2.7e-135	0.419	0.156	0.180	0.007
17	class 4	class 4	1.5e-134	3.5e-135	4.2e-135	2.7e-135	0.419	0.156	0.180	0.007
18	class 4	class 4	2.3e-134	2.7e-135	5.2e-135	2.7e-135	0.443	0.133	0.203	0.031
19	class 4	class 4	9.5e-135	4.3e-135	3.3e-135	2.7e-135	0.392	0.184	0.152	0.019
20	class 4	class 4	1.9e-134	3.0e-135	4.7e-135	2.7e-135	0.431	0.145	0.192	0.019

PVi= p-value from test i, di=distance of test pt from mean of class i
LA= Liao & Akritas's method, Our= Modified method, TP= Test point.

3.5 Classification of a block of pixels

In Sections 3.1-3.4, we discussed the classification of randomly chosen pixels, i.e., test points in the images. The same idea used to classify the test points can be used to classify test blocks. A block of pixel can be thought like a collection of many test points in the given image. Then we apply the modified method of classification discussed in Section 3.4 to each of the test point in the block. The classification of each of the test point will result in the classification of the whole block.

Chapter 4

Comparisons among classification methods on colored images.

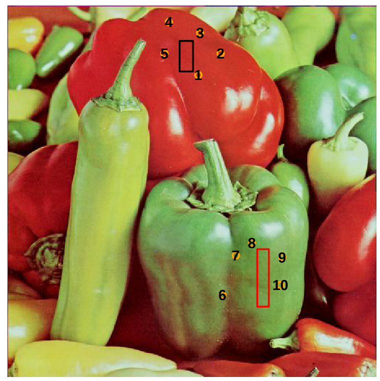
In this chapter, we employ some of the standard classification methods in the context of images and compare their classification results with our modified classification method. We consider some standard classification methods, namely, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Classification Tree Method (CTM), Test-based classification (Liao & Akritas) and Polyclass (PC) method which were introduced in Section 1.5 for comparison with our own method. We begin the chapter with the comparison in the case of binary classification of images.

4.1 Classification comparison in Binary Classification

In this section, we compare our modified method of classification with the other methods by constructing two classes in the image (binary classification). With some standard colored and grayscale images, we demonstrate that our method works better than the other classification methods in the binary classification.

We begin the comparison with a standard color image as shown in Figure 4.1. In the given image, we choose red pepper region as our first class and green pepper region as our second class. The training data for the classes are formed as discussed in Section 2.1.1 and are shown in the Figure 4.1. For the sake of convenience, we select 10 test points labeled with the numbers in such a way that first five of them are taken from class 1 regions and the rest of the points are taken from class 2 regions.

Figure 4.1: Pepper image with training data and test points



We consider the grayscale images of the three primary components of the given image, namely, red, green and blue. We employ our modified method and the other methods of classification methods to classify the selected test points in each of the components. Componentwise classification of these test points are shown in Table 4.1 along with pixel values of the test points which is scaled to $[0, 1]$.

After the classification of a test point in each of the grayscale images corresponding to red, green, and blue components, we use the majority of votes of classification to obtain the final classification of the test point. For example, if a test point is classified as coming from class 1 in the red component and as belonging to class 2 in the green and blue components, the final classification for it will be in class 2. Table 4.2 gives us the final classification of the selected test points.

From the Table 4.2, we observe that our method of classification has one misclassification, namely test point 4, while the other methods have more misclassifications. Hence, in this experiment, our method of classification works better than other methods.

Table 4.1: Classification of test points in RGB components in image Figure 4.1

Compt	TP	LA	NEW	LDA	QDA	TREE	POLY	Value
Red	1	class 1	class 1	class 1	class 1	class 1	class 1	0.45
	2	class 1	class 1	class 1	class 1	class 1	class 1	0.52
	3	class 1	class 1	class 1	class 1	class 1	class 1	0.44
	4	class 1	class 2	class 2	class 2	class 2	class 2	0.71
	5	class 1	class 1	class 1	class 1	class 1	class 1	0.44
	6	class 1	class 2	class 2	class 2	class 2	class 2	0.85
	7	class 1	class 2	class 2	class 2	class 2	class 2	0.81
	8	class 1	class 2	class 2	class 2	class 2	class 2	0.90
	9	class 1	class 2	class 2	class 2	class 2	class 2	0.81
	10	class 1	class 2	class 2	class 2	class 2	class 2	0.74
Green	1	class 1	class 1	class 2	class 2	class 1	class 2	0.51
	2	class 1	class 1	class 2	class 2	class 1	class 1	0.49
	3	class 2	class 2	class 2	class 2	class 2	class 2	0.20
	4	class 1	class 1	class 1	class 2	class 2	class 2	0.64
	5	class 2	class 2	class 2	class 2	class 2	class 2	0.18
	6	class 2	class 2	class 2	class 2	class 2	class 2	0.22
	7	class 2	class 2	class 2	class 2	class 2	class 2	0.22
	8	class 2	class 2	class 2	class 1	class 1	class 1	0.30
	9	class 2	class 2	class 2	class 2	class 2	class 2	0.25
	10	class 2	class 2	class 2	class 2	class 2	class 2	0.26
Blue	1	class 1	class 2	class 2	class 2	class 1	class 1	0.27
	2	class 1	class 2	class 2	class 2	class 2	class 2	0.28
	3	class 2	class 1	class 2	class 2	class 2	class 2	0.14
	4	class 1	class 2	class 2	class 2	class 2	class 2	0.40
	5	class 2	class 1	class 2	class 1	class 1	class 1	0.18
	6	class 2	class 1	class 2	class 1	class 1	class 1	0.18
	7	class 2	class 1	class 2	class 2	class 2	class 2	0.15
	8	class 1	class 1	class 2	class 1	class 1	class 1	0.21
	9	class 1	class 1	class 2	class 1	class 1	class 1	0.20
	10	class 2	class 1	class 2	class 2	class 2	class 2	0.12

Next, we compare the classifications in another image as displayed in Figure 4.2 in which we take sky region as our first class and vegetation region as our second class. After the classes are defined, we form training data for the classes and select some test points as shown in the Figure 4.2.

As in previous application, these selected test points in each of the grayscale images corresponding to RGB components are classified applying all the methods and are shown in Table 4.3.

After the classification of the test points in each component, we obtain their final clas-

Table 4.2: Final classification of test points in Figure 4.1

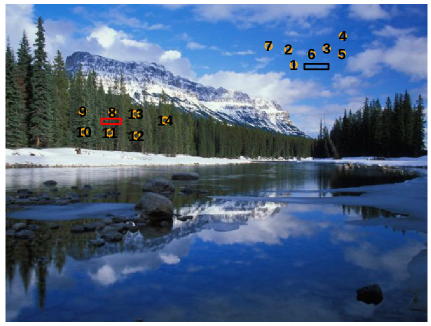
TP	LA	NEW	LDA	QDA	TREE	POLY
1	class 1	class 1	class 2	class 2	class 1	class 1
2	class 1	class 1	class 2	class 2	class 1	class 1
3	class 2	class 1	class 2	class 2	class 2	class 2
4	class 1	class 2	class 2	class 2	class 2	class 2
5	class 2	class 1	class 2	class 1	class 1	class 1
6	class 2	class 2	class 2	class 2	class 2	class 2
7	class 2	class 2	class 2	class 2	class 2	class 2
8	class 1	class 2	class 2	class 1	class 1	class 1
9	class 1	class 2	class 2	class 2	class 2	class 2
10	class 2	class 2	class 2	class 2	class 2	class 2

LA= Liao & Akritas's method; NEW= Our method;
LDA=Linear Discriminant Analysis; QDA=Quadratic Discriminant Analysis
TREE=Classification Tree; POLY=Polyclass method
TP=Test Point; Value=Pixel value of test point.

sification using majority of votes described earlier. The final classification of test points are displayed in Table 4.4. As we have 2 classes and 3 components, we will not have any tie while employing the majority of votes.

We note that first 7 of the 14 selected test points in the given image were taken from class 1 (sky) and the rest of the test points were from class 2 (vegetation). Table 4.4 shows that our method of classification has 1 misclassification, namely test points 11. Since the other methods have even more misclassifications than our methods, we could say that our method works better than the other methods in the given image.
From the above illustrations, we conclude that our method of classification works better than the other methods in the case of binary classification of image pixels.

Figure 4.2: Image with training data and test points



4.2 Classification-comparison in Multiclass Classification

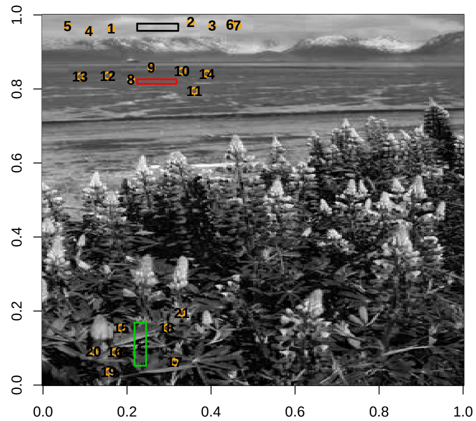
In this section, we consider more than two classes (multiclass) of image pixels and compare different methods of classifications with our method. We support our method of classification with the help of some standard images.

We first carry out comparisons in a color image by considering the grayscale images of RGB components one by one. We allow different training data and test data be used for Red, Green and Blue components. We begin with the gray scale image of red component which is shown in Figure 4.3 and specify three classes in it. Class 1 is sky region, class 2 is river region and class 3 is vegetation and the training data are formed as described in Section 2.1.1 and are displayed in the Figure 4.3.

We now apply different methods of classification and our modified method to classify the selected test points and present the results in Table 4.5.

We remark that the first seven test points are taken from class 1, another seven (from 8-14) test points are from class 2, and the last seven (from 15-21) are chosen from class 3. Test point 19 is the only misclassified test point by our method of classification while we see many misclassified test points by other methods as shown in the Table 4.3. This illustration

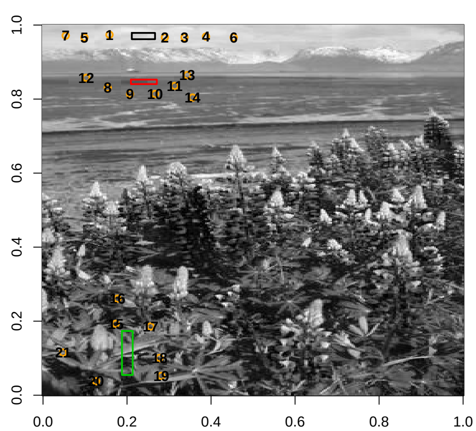
Figure 4.3: Image with training data and test points, red component.



proves that our method of classification works well in the grayscale image of red component of the given image.

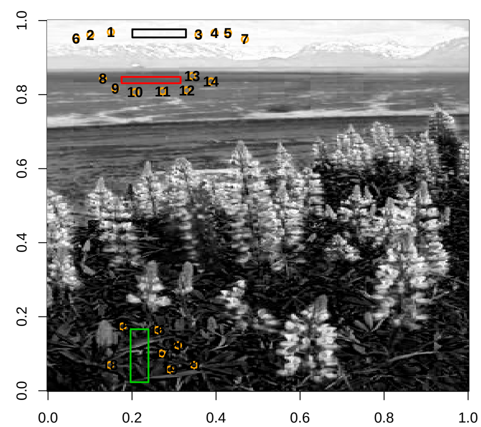
Likewise, we perform the comparisons in the grayscale images of green and blue components of the original image. The same three classes and similar training data that were used in the red component are used in the green component as displayed in Figure 4.4 and in the blue component as displayed in Figure 4.5.

Figure 4.4: Image with training data and test points, green component



From Table 4.6, we see that only two test points (19 an 21) are misclassified in the green component whereas there are more misclassified test points by other methods. Table 4.7 shows the classification result by different methods in the grayscale image of the blue component and shows that our method classifies all the test points correctly and there are some misclassifications by some of the methods.

Figure 4.5: Image with training data and test points, blue component



From the classification comparisons done in the three grayscale images of the original image, we observe that our method of classification has fewer misclassifications than the other methods. Hence, we infer that our method works better than the other methods in the given image.

We do some further comparisons on more images to see if our method of classification does work better than the other methods. We begin with a color image as shown in Figure 4.6. Let us take region of snow as class 1, vegetation as class 2, and sky as class 3. After the regions are selected, we form training data for these classes and choose some test points.

For the sake of convenience, we select first 5 of the test points from the class 1, another 5 test points (test point 6 to 10) from class 2 and the last 5 points from the class 3. The classification of these test points in all the primary components are shown in Table 4.8.

Figure 4.6: Image with training data and test points

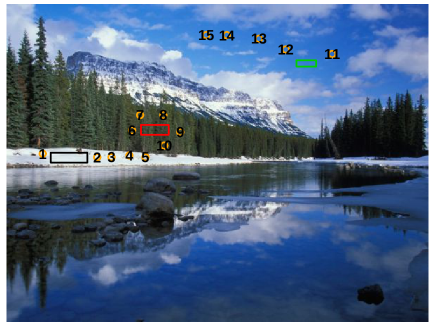


Table 4.8 gives us the classification results of the selected test points in the grayscale images of RGB components. To get the final classification, we use majority of votes as explained earlier. Since we have only three components, agreed classification by two components is the final decision. The classification of test points using the majority of votes is shown in Table 4.9.

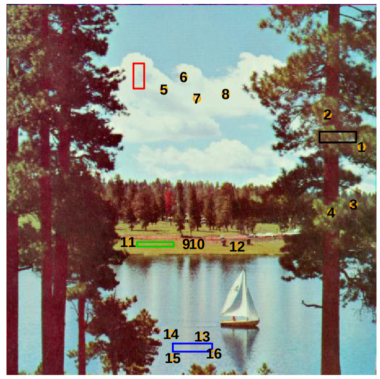
We observe that our method of classification has one misclassification where as other methods have more misclassifications. Thus, in the given image, our method is relatively better than the other methods of classification.

Finally, we close this section with the comparison of methods in another image as shown in Figure 4.7. In this image we define four classes which are tree region as class 1, river as class 2, grass as class 3, and sky as class 4. The training data are formed and some test points are selected in the image. As shown in the image, first five test points are selected from class 1, the next five from class 2 and so on.

The componentwise classification of these test points is displayed in Table 4.10.

Using the rule of majority of votes, the final classification of the selected test points are given in Table 4.11.

Figure 4.7: Image with training data and test points



From Table 4.11, we observe that test points 10 and 12 are misclassified by our method. But when we look at the decisions by the other methods, we see far more misclassified test points. Thus, in this image too, our method beats other methods.

Hence, from all the above comparisons, we conclude that our modified method works better than Liao & Akritas method, QDA method, LDA method, Classification tree method, and Polyclass method in the context of binary and multiclass classification of image pixels classification.

Table 4.3: Classification of test points in RGB components in image Figure 4.2

Compt	TP	LA	NEW	LDA	QDA	TREE	POLY	Value
	1	class 1	class 1	class 1	class 1	class 1	class 1	0.08
	2	class 1	class 1	class 1	class 1	class 1	class 1	0.09
	3	class 1	class 1	class 1	class1	class 1	class 1	0.07
	4	class 1	class 1	class 1	class 1	class 1	class1	0.07
	5	class 1	class 1	class 1	class 1	class 1	class 1	0.09
	6	class 1	class 1	class 1	class 1	class 1	class 1	0.08
Red	7	class 1	class 1	class 1	class 1	class 1	class 1	0.09
	8	class 2	class 2	class 2	class 2	class 2	class 2	0.20
	9	class 2	class 2	class 2	class 2	class 2	class 2	0.16
	10	class 2	class 2	class 2	class 2	class 2	class 2	0.18
	11	class 2	class 1	class 1	class 2	class 2	class 1	0.11
	12	class 2	class 2	class 2	class 2	class 2	class 2	0.18
	13	class 2	class 2	class 2	class 2	class 2	class 2	0.18
	14	class 2	class 2	class 2	class 2	class 2	class 2	0.27
	1	class 1	class 1	class 1	class 1	class 1	class 1	0.24
	2	class 1	class 1	class 1	class 2	class 2	class 2	0.21
	3	class 1	class 1	class 1	class 2	class 2	class 2	0.19
	4	class 1	class 1	class1	class 2	class 2	class 2	0.16
	5	class 1	class 1	class 1	class 2	class 2	class 2	0.21
	6	class 1	class 1	class 1	class 1	class 1	class 1	0.22
Green	7	class 1	class 1	class 1	class 1	class 1	class1	0.22
	8	class 2	class 2	class 2	class 2	class 2	class 2	0.29
	9	class 2	class 2	class 1	class 2	class 2	class 2	0.28
	10	class 1	class 1	class 1	class 2	class 2	class 2	0.20
	11	class 1	class 1	class 1	class 2	class 2	class 2	0.13
	12	class 1	class 1	class 1	class 2	class 2	class 2	0.20
	13	class 2	class 1	class 1	class 1	class 1	class 1	0.25
	14	class 2	class 1	class 1	class 2	class 2	class 2	0.27
	1	class 2	class 1	class 1	class 1	class 1	class 1	0.51
	2	class 1	class 2	class 1	class 2	class 1	class 1	0.45
	3	class 1	class 2	class 1	class 2	class 2	class 2	0.40
	4	class 1	class 2	class 1	class 2	class 2	class 2	0.33
	5	class 1	class 2	class 1	class 2	class 2	class 2	0.42
	6	class 1	class 2	class 1	class 2	class 1	class 1	0.45
Blue	7	class 1	class 2	class 1	class 1	class 1	class 1	0.46
	8	class 1	class 2	class 1	class 2	class 2	class 2	0.42
	9	class 1	class 2	class 1	class 2	class 2	class 2	0.43
	10	class 1	class 2	class 2	class 2	class 2	class 2	0.21
	11	class 1	class 2	class 2	class 2	class 2	class2	0.16
	12	class 1	class 2	class 2	class 2	class 2	class 2	0.16
	13	class 1	class 2	class 1	class 2	class 2	class 2	0.35
	14	class 1	class 2	class 1	class 2	class 2	class 2	0.28

Table 4.4: Final classification of test points in Figure 4.2

TP	LA	NEW	LDA	QDA	TREE	POLY
1	class 1	class 1	class 1	class 1	class 1	class 1
2	class 1	class 1	class 1	class 2	class 1	class 1
3	class 1	class 1	class 1	class 2	class 2	class 2
4	class 1	class 1	class 1	class 2	class 2	class 2
5	class 1	class 1	class 1	class 2	class 2	class 2
6	class 1	class 1	class 1	class 1	class 1	class 1
7	class 1	class 1	class 1	class 1	class 1	class 1
8	class 2	class 2	class 2	class 2	class 2	class 2
9	class 2	class 2	class 1	class 2	class 2	class 2
10	class 1	class 2	class 2	class 2	class 2	class 2
11	class 1	class 1	class 1	class 2	class 2	class 2
12	class 1	class 2	class 2	class 2	class 2	class 2
13	class 2	class 2	class 1	class 2	class 2	class 2
14	class 2	class 2	class 1	class 2	class 2	class 2

LA= Liao & Akritas's method; NEW= Our method;
LDA=Linear Discriminant Analysis; QDA=Quadratic Discriminant Analysis
TREE=Classification Tree; POLY=Polyclass method
TP=Test Point; Value=Pixel value of test point.

Table 4.5: Classification of test points in image Figure 4.3, Red component.

TP	LA	NEW	LDA	QDA	TREE	POLY	Value
1	class 1	class 1	class 1	class 1	class 1	class 1	0.68
2	class 1	class 1	class 1	class 1	class 1	class 1	0.80
3	class 1	class 1	class 1	class 1	class 1	class 1	0.75
4	class 1	class 1	class 1	class 1	class 1	class 1	0.72
5	class 2	class 1	class 1	class 1	class 1	class 1	0.65
6	class 1	class 1	class 1	class 1	class 1	class 1	0.71
7	class 1	class 1	class 1	class 1	class 1	class 1	0.70
8	class 3	class 2	class 3	class 2	class 2	class 3	0.37
9	class 3	class 2	class 3	class 3	class 2	class 3	0.37
10	class 3	class 2	class 3	class 3	class 3	class 3	0.35
11	class 3	class 2	class 3	class 3	class 2	class 3	0.37
12	class 2	class 2	class 3	class 2	class 2	class 2	0.39
13	class 3	class 2	class 3	class 3	class 3	class 3	0.35
14	class 3	class 2	class 3	class 3	class 3	class 3	0.34
15	class 3	class 3	class 3	class 3	class 3	class 3	0.30
16	class 3	class 3	class 3	class 3	class 3	class 3	0.25
17	class 3	class 3	class 3	class 3	class 3	class 3	0.06
18	class 3	class 3	class 3	class 3	class 3	class 3	0.25
19	class 3	class 2	class 3	class 3	class 3	class 3	0.34
20	class 3	class 3	class 3	class 3	class 3	class 3	0.29
21	class 3	class 3	class 3	class 3	class 3	class 3	0.14

Table 4.6: Classification of test points in Figure 4.4, green component.

TP	LA	NEW	LDA	QDA	TREE	POLY	Value
1	class 1	class 1	class 1	class 1	class 1	class 1	0.86
2	class 1	class 1	class 1	class 1	class 1	class 1	0.84
3	class 1	class 1	class 1	class 1	class 1	class 1	0.86
4	class 1	class 1	class 1	class 1	class 1	class 1	0.85
5	class 1	class 1	class 1	class 1	class 1	class 1	0.85
6	class 1	class 1	class 1	class 1	class 1	class 1	0.85
7	class 2	class 1	class 1	class 1	class 1	class 1	0.82
8	class 2	class 2	class 3	class 3	class 3	class 3	0.43
9	class 2	class 2	class 3	class 3	class 3	class 3	0.48
10	class 2	class 2	class 3	class 3	class 3	class 3	0.46
11	class 2	class 2	class 3	class 3	class 3	class 3	0.43
12	class 2	class 2	class 3	class 3	class 3	class 3	0.44
13	class 2	class 2	class 3	class 3	class 3	class 3	0.60
14	class 2	class 2	class 3	class 3	class 3	class 3	0.45
15	class 3	class 3	class 3	class 3	class 3	class 3	0.10
16	class 3	class 3	class 3	class 3	class 2	class 3	0.36
17	class 3	class 3	class 3	class 3	class 3	class 3	0.21
18	class 3	class 3	class 3	class 3	class 3	class 3	0.13
19	class 3	class 2	class 3	class 2	class 3	class 3	0.38
20	class 3	class 3	class 3	class 3	class 3	class 3	0.18
21	class 2	class 2	class 3	class 3	class 3	class 3	0.58

Table 4.7: Classification of test points in Figure 4.5, blue component.

Obs	LA	NEW	LDA	QDA	TREE	POLY	Value
1	class 1	class 1	class 1	class 1	class 1	class 1	0.99
2	class 1	class 1	class 1	class 1	class 1	class 1	1.00
3	class 1	class 1	class 1	class 1	class 1	class 1	0.98
4	class 1	class 1	class 1	class 1	class 1	class 1	0.98
5	class 1	class 1	class 1	class 1	class 1	class 1	0.99
6	class 1	class 1	class 1	class 1	class 1	class 1	0.99
7	class 1	class 1	class 1	class 1	class 1	class 1	0.97
8	class 2	class 2	class 2	class 2	class 2	class 2	0.38
9	class 2	class 2	class 2	class 3	class 3	class 3	0.48
10	class 2	class 2	class 2	class 3	class 3	class 3	0.49
11	class 2	class 2	class 2	class 3	class 3	class 3	0.49
12	class 2	class 2	class 2	class 3	class 3	class 3	0.49
13	class 2	class 2	class 2	class 2	class 2	class 2	0.44
14	class 2	class 2	class 2	class 2	class 2	class 2	0.36
15	class 3	class 3	class 3	class 3	class 3	class 3	0.16
16	class 3	class 3	class 3	class 3	class 3	class 3	0.11
17	class 3	class 3	class 3	class 3	class 3	class 3	0.00
18	class 3	class 3	class 3	class 3	class 3	class 3	0.22
19	class 3	class 3	class 3	class 3	class 3	class 3	0.04
20	class 3	class 3	class 3	class 3	class 3	class 3	0.01
21	class 3	class 3	class 3	class 3	class 3	class 3	0.17

Table 4.8: Classification of test points in image Figure 4.6

Comp	TP	LA	NEW	LDA	QDA	TREE	POLY	Value
	1	class 1	class 1	class 1	class 1	class 1	class 1	0.64
	2	class 1	class 1	class 1	class 1	class 1	class 1	0.79
	3	class 1	class 1	class 1	class 1	class 1	class 1	0.74
	4	class 2	class 1	class 1	class 1	class 1	class 1	0.51
	5	class 2	class 1	class 1	class 1	class 1	class 1	0.57
	6	class 2	class 2	class 2	class 2	class 2	class 2	0.20
	7	class 3	class 3	class 2	class 2	class 3	class 3	0.15
Red	8	class 2	class 2	class 2	class 2	class 2	class 2	0.19
	9	class 2	class 2	class 2	class 2	class 2	class 2	0.19
	10	class 2	class 2	class 2	class 2	class 2	class 2	0.33
	11	class 3	class 3	class 3	class 3	class 3	class 3	0.09
	12	class 3	class 3	class 3	class 3	class 3	class 3	0.09
	13	class 3	class 3	class 3	class 3	class 3	class 3	0.08
	14	class 3	class 3	class 3	class 3	class 3	class 3	0.09
	15	class 3	class 3	class 3	class 3	class 3	class 3	0.09
	1	class 1	class 1	class 1	class 1	class 1	class 1	0.67
	2	class 1	class 1	class 1	class 1	class 1	class 1	0.82
	3	class 1	class 1	class 1	class 1	class 1	class 1	0.76
	4	class 2	class 1	class 1	class 1	class 1	class 1	0.53
	5	class 1	class 1	class 1	class 1	class 1	class 1	0.58
	6	class 3	class 3	class 2	class 2	class 3	class 2	0.21
	7	class 3	class 3	class 2	class 2	class 2	class 2	0.17
Green	8	class 3	class 3	class 2	class 2	class 2	class 2	0.18
	9	class 3	class 3	class 2	class 3	class 3	class 3	0.24
	10	class 2	class 2	class 2	class 2	class 2	class 2	0.38
	11	class 3	class 3	class 2	class 2	class 2	class 2	0.20
	12	class 3	class 3	class 2	class 3	class 3	class 3	0.22
	12	class 3	class 3	class 2	class 2	class 2	class 2	0.20
	14	class 3	class 3	class 2	class 2	class 3	class 2	0.21
	15	class 3	class 3	class 2	class 2	class 3	class 3	0.22
	1	class 1	class 1	class 1	class 1	class 1	class 1	0.73
	2	class 1	class 1	class 1	class 1	class 1	class 1	0.86
	3	class 1	class 1	class 1	class 1	class 1	class 1	0.80
	4	class 3	class 3	class 3	class 1	class 1	class 1	0.57
	5	class 3	class 3	class 1	class 1	class 1	class 1	0.61
	6	class 2	class 2	class 2	class 2	class 2	class 2	0.17
	7	class 2	class 2	class 2	class 2	class 2	class 2	0.20
Blue	8	class 2	class 2	class 2	class 2	class 2	class 2	0.16
	9	class 2	class 2	class 2	class 2	class 2	class 2	0.26
	10	class 3	class 3	class 2	class 2	class 2	class 2	0.42
	11	class 3	class 3	class 2	class 2	class 2	class 2	0.42
	12	class 3	class 3	class 3	class 3	class 3	class 3	0.46
	13	class 3	class 3	class 2	class 2	class 2	class 2	0.41
	14	class 3	class 3	class 3	class 2	class 2	class 2	0.43
	15	class 3	class 3	class 3	class 2	class 2	class 3	0.44

Table 4.9: Final classification of test points in image Figure 4.6

TP	LA	NEW	LDA	QDA	TREE	POLY
1	class 1	class 1	class 1	class 1	class 1	class 1
2	class 1	class 1	class 1	class 1	class 1	class 1
3	class 1	class 1	class 1	class 1	class 1	class 1
4	class 2	class 1	class 1	class 1	class 1	class 1
5	class 2	class 1	class 1	class 1	class 1	class 1
6	class 2	class 2	class 2	class 2	class 2	class 2
7	class 3	class 3	class 2	class 2	class 2	class 2
8	class 2	class 2	class 2	class 2	class 2	class 2
9	class 2	class 2	class 2	class 2	class 2	class 2
10	class 2	class 2	class 2	class 2	class 2	class 2
11	class 3	class 3	class 2	class 2	class 2	class 2
12	class 3	class 3	class 3	class 3	class 3	class 3
13	class 3	class 3	class 2	class 2	class 2	class 2
14	class 3	class 3	class 3	class 2	class 3	class 2
15	class 3	class 3	class 3	class 2	class 3	class 3

LA= Liao & Akritas's method; NEW= Our method;
LDA=Linear Discriminant Analysis; QDA=Quadratic Discriminant Analysis
TREE=Classification Tree; POLY=Polyclass method
TP=Test Point.

Table 4.10: Classification of test points in image Figure 4.7

Compt	TP	LA	NEW	LDA	QDA	TREE	POLY	Value
	1	class 1	class 1	class 1	class 1	class 1	class 1	0.32
	2	class 1	class 1	class 1	class 1	class 1	class 1	0.36
	3	class 2	class 1	class 1	class 1	class 1	class 1	0.42
	4	class 1	class 1	class 1	class 1	class 1	class 1	0.34
	5	class 2	class 2	class 4	class 2	class 2	class 2	0.62
	6	class 3	class 2	class 2	class 2	class 2	class 2	0.67
	7	class 2	class 2	class 4	class 2	class 2	class 2	0.62
Red	8	class 2	class 4	class 4	class 2	class 2	class 2	0.55
	9	class 3	class 3	class 3	class 3	class 3	class 3	0.76
	10	class 3	class 3	class 3	class 2	class 2	class 2	0.72
	11	class 3	class 3	class 3	class 2	class 3	class 3	0.85
	12	class 2	class 2	class 2	class 2	class 2	class 2	0.65
	13	class 2	class 4	class 4	class 4	class 4	class 4	0.59
	14	class 2	class 4	class 4	class 2	class 2	class 2	0.60
	15	class 2	class 4	class 4	class 4	class 4	class 4	0.58
	16	class 2	class 4	class 4	class 4	class 4	class 4	0.58
	1	class 1	class 1	class 1	class 1	class 1	class 1	0.26
	2	class 1	class 1	class 1	class 1	class 1	class 1	0.11
	3	class 2	class 2	class 2	class 2	class 2	class 2	0.49
	4	class 1	class 1	class 1	class 1	class 1	class 1	0.10
	5	class 2	class 2	class 4	class 2	class 2	class 2	0.72
	6	class 2	class 2	class 4	class 2	class 2	class 2	0.75
	7	class 2	class 2	class 2	class 2	class 2	class 2	0.69
Green	8	class 2	class 2	class 2	class 2	class 2	class 2	0.65
	9	class 3	class 3	class 4	class 3	class 3	class 3	0.84
	10	class 3	class 4	class 4	class 2	class 2	class 2	0.80
	11	class 3	class 3	class 3	class 2	class 3	class 3	0.91
	12	class 3	class 4	class 4	class 2	class 2	class 2	0.82
	13	class 2	class 4	class 4	class 4	class 4	class 4	0.78
	14	class 3	class 4	class 4	class 4	class 4	class 4	0.79
	15	class 2	class 4	class 4	class 4	class 4	class 4	0.78
	16	class 2	class 4	class 4	class 4	class 4	class 4	0.77
	1	class 2	class 1	class 1	class 1	class 1	class 1	0.18
	2	class 1	class 1	class 1	class 1	class 1	class 1	0.10
	3	class 2	class 1	class 2	class 2	class 2	class 2	0.41
	4	class 1	class 1	class 1	class 1	class 1	class 1	0.13
	5	class 2	class 2	class 2	class 2	class 2	class 2	0.70
	6	class 2	class 2	class 4	class 2	class 2	class 2	0.71
	7	class 2	class 2	class 2	class 2	class 2	class 2	0.68
Blue	8	class 2	class 2	class 2	class 2	class 2	class 2	0.67
	9	class 3	class 3	class 4	class 4	class 4	class 4	0.8 1
	10	class 3	class 4	class 4	class 4	class 4	class 4	0.79
	11	class 3	class 3	class 4	class 2	class 3	class 3	0.85
	12	class 3	class 3	class 4	class 3	class 3	class 3	0.82
	13	class 3	class 4	class 4	class 4	class 4	class 4	0.81
	14	class 3	class 3	class 4	class 4	class 4	class 4	0.81
	15	class 3	class 4	class 4	class 4	class 4	class 4	0.80
	16	class 3	class 4	class 4	class 4	class 4	class 4	0.80

Table 4.11: Final classification of test points in image Figure 4.7

Obs	LA	NEW	LDA	QDA	TREE	POLY
1	class 1	class 1	class 1	class 1	class 1	class 1
2	class 1	class 1	class 1	class 1	class 1	class 1
3	class 2	class 1	class 2	class 2	class 2	class 2
4	class 1	class 1	class 1	class 1	class 1	class 1
5	class 2	class 2	class 4	class 2	class 2	class 2
6	class 2	class 2	class 4	class 2	class 2	class 2
7	class 2	class 2	class 2	class 2	class 2	class 2
8	class 2	class 2	class 2	class 2	class 2	class 2
9	class 3	class 3	class 4	class 3	class 3	class 3
10	class 3	class 4	class 4	class 2	class 2	class 2
11	class 3	class 3	class 3	class 2	class 3	class 3
12	class 3	class 2	class 4	class 2	class 2	class 2
13	class 2	class 4	class 4	class 4	class 4	class 4
14	class 3	class 4	class 4	class 4	class 4	class 4
15	class 2	class 4	class 4	class 4	class 4	class 4
16	class 2	class 4	class 4	class 4	class 4	class 4

LA= Liao & Akritas's method; NEW= Our method;
LDA=Linear Discriminant Analysis; QDA=Quadratic Discriminant Analysis
TREE=Classification Tree; POLY=Polyclass method
TP=Test Point; Value=Pixel value of test point.

Conclusion

In this report, we considered image pixels classification problem. A test-based classification method introduced by Liao and Akritas (2007) was first implemented in some grayscale images to classify image pixels. Liao and Akritas employ hypothesis testings in their method where the main idea is to use p-values obtained from the hypothesis testings as a distance measure. We apply their method in binary and multiclass classification of image pixels and observe that it fails to perform well in some images. Particularly, their method did not work well in the case when the p-values from the hypothesis testings were very small. This feature of Liao and Akritas's method motivated us to introduce a new classification method which make conclusions based on combined evidence from the minimum distance and hypothesis testing. This method eliminated the drawback of Liao and Akritas's method. We implemented our method in several grayscale images and found in extensive experiments that our method consistently worked well in the classification of image pixels for both binary and multiclass cases.

Performance of the modified method was also compared with some standard classification methods, namely, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Classification Tree and Polyclass Classification. We also compared their performances in color images and confirmed that the modified method gave less mis-classification than the other methods in both binary and multiclass settings.

Bibliography

Green, B. W. (1983). *Digital Image Prcessing, A Systems Approach*. Van Nostrand Reinhold Company.

Hastie, T., R. Tibshirani, and F. J. (2001). *The Elements of Statistical Learning*. Spring-Verlag, New York.

Liao, S.-M. and M. Akritas (2007). Test-based classification: A linkage between classification and statistical testing. *Statistics & Probability Letters 77*, 1269–1281.

Stone, J. C., H. M. Hansen, C. Kooperberg, and K. Y. Truong (1997). Polynomail splines and their tensor products in extended linear modeling. *Annals of Statistics 25*, 1371–1470.

Vapnik, V. (1982). *The nature of Statistical Learning Theroy*. Spring-Verlag, New York.

Appendix A

R codes for Binary Classification.

```
library("rimage")
#A=read.jpeg("brain.jpg")
A=read.jpeg("goldhill.jpg")
Adat=imagematrix(A, type="grey")

library("rimage")
#A=read.jpeg("brain.jpg")
A=read.jpeg("goldhill.jpeg")
Adat=imagematrix(A, type="grey")

class.data.fun=function(Adat, histogram=T){
#if (histogram ==T)  {par(mfrow=c(1,3)); scan(what="character", nmax=1)  }
image(Adat, col=gray( (0:254)/255 )   )
find.data=list()
for ( i in 1:2){
z=unlist(locator(2) )
lines(x=c(z[1],z[2], z[2], z[1],z[1]),y=c(z[3], z[3],
z[4], z[4], z[3]), col=ifelse(i==1, "green", "purple"), lwd=2  )
class1.x=round(z[1:2]*nrow(Adat))
```

```
class1.y=round(z[3:4]*ncol(Adat)  )
class1.dat=Adat[class1.x[1]:   class1.x[2],   class1.y[1]: class1.y[2] ]
find.data[[i]]=class1.dat
}
if (histogram ==T) { hist(find.data[[1]]); hist(find.data[[1]]) }
find.data
}
#LIAO AND AKRITAS
new.obs.class.T.aka=function(Adat, class1, class2,p1,p2, method="Wilcox" ,
 click=T, obsgiven=mean(Adat) ){
class1=classes[[1]]; class2=classes[[2]]
if (click==T){
image(Adat, col=gray( (0:254)/255 )   )
z=unlist(locator(1) )
obs=Adat[round(z[1]*nrow(Adat)),  round(z[2]*ncol(Adat)  )   ]
} else obs= obsgiven

#obs=new.obs(Adat)
temp1=c(class1, obs)
if (method=="Wilcox") mytest=wilcox.test else mytest=t.test
p1=mytest(temp1, c(class2) )$p.value
temp2=c(class2, obs)
p2=mytest( c(class1), temp2 )$p.value
result=ifelse(p1<p2,"class 1","class 2")
#result
list (result=result,p1,p2)
}
#MODIFIED METHOD
```

```
new.obs.class.T=function(Adat, d1,d2, method="Wilcox" ,
click=T, obsgiven=mean(Adat) ){
class1=classes[[1]]; class2=classes[[2]]
if (click==T){
image(Adat, col=gray( (0:254)/255 )   )
z=unlist(locator(1) )
obs=Adat[round(z[1]*nrow(Adat)),  round(z[2]*ncol(Adat)  )   ]
} else obs= obsgiven

#obs=new.obs(Adat)
temp1=c(class1, obs)
if (method=="Wilcox") mytest=wilcox.test else mytest=t.test
p1=mytest(temp1, c(class2) )$p.value
temp2=c(class2, obs)
p2=mytest( c(class1), temp2 )$p.value
d1=abs(obs-mean(class1))
d2=abs(obs-mean(class2))


ind=abs(obs-mean(class1)) < abs(obs-mean(class2))
if (ind){
if ((p1 <1e-3)&(p2<1e-3) ) { result="class 1"}
 else{ result= ifelse( (p1<p2), "class 1", "class2") }
} else {
if ((p1 <1e-3)&(p2<1e-3) ) { result="class 2"}
else{ result= ifelse( (p1<p2), "class 1", "class2") }
 }
result
```

```
list(result=result, p1,p2, d1,d2)
}
# with prior probability
new.obs.class.T.with.prior=function(Adat,  method="Wilcox",
click=T, obsgiven=mean(Adat), no.prior=F ){
# calculate prior prob. If no.prior is necessary, set 1/2 for both prior prob
if (no.prior==T){ prior1=prior2=1/2} else {
mc1=mean(class1);   mc2=mean(class2)
if (mc1<mc2) {prior1=mean(Adat< (mc1+mc2)/2); prior2=1-prior1      }
else {prior2=mean(Adat< (mc1+mc2)/2); prior1=1-prior2      }
}


# choose a point to classify
if (click==T){
image(Adat, col=gray( (0:254)/255 )   )
z=unlist(locator(1) )
obs=Adat[round(z[1]*nrow(Adat)),  round(z[2]*ncol(Adat)  )   ]
} else obs=obsgiven

# decide which class the point belongs to
temp1=c(class1, obs)
if (method=="Wilcox") mytest=wilcox.test else mytest=t.test
p1=mytest(temp1, c(class2) )$p.value
temp2=c(class2, obs)
p2=mytest( c(class1), temp2 )$p.value

ind=abs(obs-mean(class1)) < abs(obs-mean(class2))
```

```
if (ind){
if ((p1 <1e-3)&(p2<1e-3) ) { result="class 1"} else{ result=
ifelse( (p1*(1-prior1) <p2*(1-prior2)), "class 1", "class2") }
} else {
if ((p1 <1e-3)&(p2<1e-3) ) { result="class 2"} else{ result=
ifelse( (p1*(1-prior1)<p2*(1-prior2)), "class 1", "class2") }
 }
result
list(result=result, prior1, prior2)
}

## classify for a block of pixels
new.obs.class.T.block=function(Adat, click=F, blockpixels=Adat, use.prior=T){

if (click==T){
image(Adat, col=gray( (0:254)/255 )   )
z=unlist(locator(2) )
obs.x=round(z[1:2]*nrow(Adat))
obs.y=round(z[3:4]*ncol(Adat)  )
obs=Adat[obs.x[1]:   obs.x[2],   obs.y[1]: obs.y[2] ]
} else obs= blockpixels

classres=matrix("NA", nrow=nrow(obs), ncol=ncol(obs) )
for ( i in 1:nrow(obs) ){
 for (j in 1:ncol(obs) ) {
if (use.prior ==T){
classres[i,j]=new.obs.class.T.with.prior(Adat, class1, class2,
method="Wilcox", click=F, obsgiven=obs[i,j], no.prior=F ) } else {
```

```
classres[i,j]=new.obs.class.T.with.prior(Adat, class1, class2,
method="Wilcox", click=F, obsgiven=obs[i,j], no.prior=T ) }
}
}
list(individual.class=classres, summarys=table(c(classres) ))
}

# for comparing Liao-Aka and our method
common.obs.Liao.Aka.ours=function(Adat,labeluse=1){
z=unlist(locator(1))
points(z[1],z[2],lwd=2,col="orange")
text(x=z[1],y=z[2],labels=labeluse,font=2)
obs=Adat[round(z[1]*nrow(Adat)),round(z[2]*ncol(Adat))]
obs
}
```

```
# Define two classes classes=class.data.fun(Adat,histogram=F)

class1=class.data.fun(Adat) class2=class.data.fun(Adat)


fortable=character() for (i in 1:6){
commonobs=common.obs.Liao.Aka.ours(Adat,i)
akares=new.obs.class.T.aka(Adat,class1,class2, method="Wilcox" ,
click=F, obsgiven=commonobs) ours=new.obs.class.T(Adat,
method="Wilcox" ,  click=F, obsgiven=commonobs )
fortable=rbind(fortable, c(i, akares,ours)) }
```

fortable

Appendix B

R codes for Multiclass Classification.

```
class.data.fun.poly=function(Adat, histogram=T){
if (histogram ==T)  {par(mfrow=c(1,3)); scan(what="character", nmax=1)  }
image(Adat, col=gray( (0:254)/255 )   )
find.data=list()
for ( i in 1:2){
library(splancs)
n=scan(what=numeric(),nmax=1)
z=unlist(locator(n) )
polygon(x=z[1:n],y=z[(n+1):(2*n)])
myp=data.frame(x=z[1:n],y=z[(n+1):(2*n)])
x0=seq(nrow(Adat))/nrow(Adat)
y0=seq(ncol(Adat))/ncol(Adat)
indicator.map=inout(data.frame(expand.grid(x0, y0)), myp)
keeps=expand.grid(seq(nrow(Adat)), seq(ncol(Adat)) )[indicator.map,]
class1.dat=Adat[unlist(keeps[,1]), unlist(keeps[,2]) ]
find.data[[i]]=class1.dat
}
if (histogram ==T) { hist(find.data[[1]]); hist(find.data[[1]]) }
find.data
```

```
}

class.data.fun=function(Adat, histogram=T){
if (histogram ==T)  {par(mfrow=c(1,3)); scan(what="character", nmax=1)  }
 image(Adat, col=gray( (0:254)/255 )   )
find.data=list()
for ( i in 1:2){
z=unlist(locator(2) )
lines(x=c(z[1],z[2], z[2], z[1],z[1]),y=c(z[3], z[3], z[4], z[4], z[3]),
col=ifelse(i==1, "green", "purple"), lwd=2  )
class1.x=round(z[1:2]*nrow(Adat))
class1.y=round(z[3:4]*ncol(Adat)  )
class1.dat=Adat[class1.x[1]:   class1.x[2],   class1.y[1]: class1.y[2] ]
find.data[[i]]=class1.dat
}
if (histogram ==T) { hist(find.data[[1]]); hist(find.data[[1]]) }
find.data
}


#consider multiple classes
class.data.fun.multiclass=function(Adat,k=3, histogram=T){
if (histogram ==T)  {par(mfrow=c(1,k+1)); scan(what="character", nmax=1)  }
image(Adat, col=gray( (0:254)/255 )   )
find.data=list()
for ( i in 1:k){
z=unlist(locator(2) )
lines(x=c(z[1],z[2], z[2], z[1],z[1]),y=c(z[3], z[3],
```

```
z[4], z[4], z[3]), col=i, lwd=2  )
class1.x=round(z[1:2]*nrow(Adat))
class1.y=round(z[3:4]*ncol(Adat)  )
class1.dat=Adat[class1.x[1]:   class1.x[2],   class1.y[1]: class1.y[2] ]
find.data[[i]]=class1.dat
}
if (histogram ==T) { for (i in 1:k){ hist(find.data[[i]]) }}
find.data
}



#consider multiple classes from colored image
# Adat is directly from read.jpeg command
class.3d.data.fun.multiclass=function(Adat,k=3, histogram=T){
if (histogram ==T)  {par(mfrow=c(1,k+1)); scan(what="character", nmax=1)  }
plot(Adat  )
find.data=list()
for ( i in 1:k){
z=unlist(locator(2) )
lines(x=c(z[1],z[2], z[2], z[1],z[1]),y=c(z[3],
z[3], z[4], z[4], z[3]), col=i, lwd=2  )
class1.x=round(z[1:2])
class1.y=round(z[3:4] )

class1.dat=Adat[class1.y[1]: class1.y[2], class1.x[1]:   class1.x[2],     ]
find.data[[i]]=class1.dat
}
if (histogram ==T) { for (i in 1:k){ hist(find.data[[i]]) }}
```

```
find.data
}


plot.classes=function(classes){
k=length(classes)
b=lapply(classes,density)
ally=numeric()
for (i in 1:k){
ally=c(ally, b[[i]]$y)
}

for (i in 1:k){
  di=density(c(classes[[i]]))
  if (i==1) {plot(di,xlim=range(classes),ylim=range(ally),col=1,
  xlab="Pixel Values on scale [0,1]" ,main="Kernel Density Estimate")} else {
lines(di$x,di$y, col=i) }
  ordy=(1:length(di$y))[di$y==max(di$y)]
  text((di$x)[ordy],(di$y)[ordy]+1.5,labels=i)

 }
}


#classification rule for multi classes
Aka.new.obs.class.T.multiclasses=function(Adat, classes,  method="rank" ,
 click=T, obsgiven=mean(Adat) ){
#class1=classes[[1]]; class2=classes[[2]]
```

```
if (click==T){
#image(Adat, col=gray( (0:254)/255 )   )
z=unlist(locator(1) )
points(z[1], z[2], lwd=2, col="orange")
obs=Adat[round(z[1]*nrow(Adat)),  round(z[2]*ncol(Adat)  )   ]
} else obs= obsgiven
temp=calculate.p.and.dist(classes, obs, method, 1:length(classes) )
p=temp$p
result= paste('class', Liao.Aka(p, classes, obs, method))
list(result=result,pvalue=p)
}




#classification rule for 3 classes
new.obs.class.T.3classes=function(Adat, classes,  method="rank" ,
click=T, obsgiven=mean(Adat),threshold=1e-3 ){
#class1=classes[[1]]; class2=classes[[2]]
if (click==T){
#image(Adat, col=gray( (0:254)/255 )   )
z=unlist(locator(1) )
points(z[1], z[2], lwd=2, col="orange")
obs=Adat[round(z[1]*nrow(Adat)),  round(z[2]*ncol(Adat)  )   ]
} else obs= obsgiven

k=length(classes)
temp=calculate.p.and.dist(classes, obs, method, 1:k)
```

```
p=temp$p
all.abs.dist=temp$all.abs.dist

if (max(p)<threshold)  result=paste('class', (seq(k))[order(all.abs.dist)[1] ] )
if (min(p)>threshold) result= paste('class', Liao.Aka(p, classes, obs, method))
if (sum(p<threshold)==1) result=paste('class',(seq(k))[p<threshold])
if (sum(p<threshold)==2) {whichclass=(seq(k))[p<threshold]
   result=decide.2(p[whichclass[1]], p[whichclass[2]],
   classes[[whichclass[1]]], classes[[whichclass[2]]], obs, threshold)
   }
list(result=result,pvalue=p,distance=all.abs.dist)
}

# for comparing Liao-Aka and our method
#common.obs.Liao.Aka.ours=function(Adat,labeluse=1){
#z=unlist(locator(1))
#points(z[1],z[2],lwd=2,col="orange")
#text(x=z[1],y=z[2],labels=labeluse,font=2)
#obs=Adat[round(z[1]*nrow(Adat)),round(z[2]*ncol(Adat))]
#obs
#}

#for colored image
common.obs.Liao.Aka.ours=function(Adat,labeluse=1){
z=unlist(locator(1))
points(z[1],z[2],lwd=2,col="orange")
text(x=z[1],y=z[2],labels=labeluse,font=2)
if (length(dim(Adat))>2) obs=Adat[round(z[2]),round(z[1]),] else
```

```
obs= Adat[round(z[1]*nrow(Adat)),round(z[2]*ncol(Adat))]
obs
}

#classification rule for multi classes for one obs without prior
new.obs.class.T.multiclasses=function(Adat, classes,classlab=1:length(classes),
method="rank" ,  click=T, obsgiven=mean(Adat),threshold=1e-3 ){
#class1=classes[[1]]; class2=classes[[2]]
if (click==T){
#image(Adat, col=gray( (0:254)/255 )   )
z=unlist(locator(1) )
points(z[1], z[2], lwd=2, col="orange")
obs=Adat[round(z[1]*nrow(Adat)),  round(z[2]*ncol(Adat)  )   ]
} else obs= obsgiven

k=length(classlab)
temp=calculate.p.and.dist(classes, obs, method, classlab)
p=temp$p
all.abs.dist=temp$all.abs.dist

if (max(p)<threshold)  result=paste('class', classlab[order(all.abs.dist)[1] ] )
if (min(p)>threshold) result= paste('class', Liao.Aka(p, classes, obs, method))
if (sum(p<threshold)==1) result=paste('class',classlab[p<threshold])
if (sum(p<threshold)==2) {whichclass=classlab[p<threshold]
   result=decide.2(p[whichclass[1]], p[whichclass[2]],classes[[whichclass[1]]],
    classes[[whichclass[2]]], obs, threshold)
   }
if((sum(p<threshold)>2)&(sum(p<threshold)<k)){
```

```
currentlab=classlab[p<threshold]
result=new.obs.class.T.multiclasses(Adat, classes,classlab=currentlab,
  method,  click, obsgiven,threshold)
}
list(result=result,pvalue=p,distance=all.abs.dist)
}

##############classification rule for multi classes for one obs with prior
multiclass.new.obs.class.T.with.prior=function(Adat, classes,
classlab=1:length(classes), method, click=F, obsgiven=0, no.prior=F,threshold=1e-3 ){
if (click==T){
#image(Adat, col=gray( (0:254)/255 )   )
z=unlist(locator(1) )
points(z[1], z[2], lwd=2, col="orange")
obs=Adat[round(z[1]*nrow(Adat)),  round(z[2]*ncol(Adat)  )   ]
} else obs= obsgiven

k=length(classlab)
temp=calculate.p.and.dist(classes, obs, method, classlab)
p=temp$p
all.abs.dist=temp$all.abs.dist

if (max(p)<threshold)  result=paste('class', classlab[order(all.abs.dist)[1] ] )
if (min(p)>threshold) result= paste('class', Liao.Aka.prior(p, classes, obs, method))
if (sum(p<threshold)==1) result=paste('class',classlab[p<threshold])
if (sum(p<threshold)==2) {whichclass=classlab[p<threshold]
   result=decide.2(p[whichclass[1]], p[whichclass[2]],
   classes[[whichclass[1]]], classes[[whichclass[2]]], obs, threshold)
```

```
   }
if((sum(p<threshold)>2)&(sum(p<threshold)<k)){
currentlab=classlab[p<threshold]
result=new.obs.class.T.multiclasses(Adat, classes,classlab=currentlab,
method,  click, obsgiven,threshold)
}

result
}
```

```
#order(tt) gives 2, 3,1 for tt=c(1, 0.3, 0.4)
# it means second element gives smallest value; third element gives
#second smallest and first element gives largest.

Liao.Aka=function(p, classes, obs, method){
klist=1: length(classes)
while(length(klist)>1) {

  eliminate=(seq(length(p)))[order(p )[length(klist) ] ]
  klist=klist[klist!=eliminate]
  if (length(klist)>1 ) {temp=calculate.p.and.dist(classes, obs, method, klist)
  p=temp$p  }
}
klist
```

```
}

Liao.Aka.prior=function(p, classes, obs, method){
priors=numeric()
# calculate prior prob. If no.prior is necessary,
# set 1/k for each  prior prob
k=length(classes)
if (no.prior==T){ priors=1/k} else {
mc=unlist(lapply(classes,median));
mc.order=order(mc)
sorted.mc=sort(mc)
LMC=c(-sorted.mc[1],sorted.mc[-k])
RMC=c(sorted.mc[-1],1e+16)
for (j in 1:k){
#when the variances of the different classes  are different then
#the two terms average in the line below  does not give the  correct partition.
priors[mc.order[j]]=mean((Adat>(LMC[j]+sorted.mc[j])/2)&
(Adat<=(RMC[j]+sorted.mc[j])/2))
 }
}


klist=1: length(classes)
while(length(klist)>1) {

  eliminate=(seq(length(p)))[order(p*(1-priors) )[length(klist) ] ]
  klist=klist[klist!=eliminate]
  if (length(klist)>1 ) {temp=calculate.p.and.dist(classes, obs, method, klist)
```

```
  p=temp$p  }
}
klist
}




#### calculate the pvalues for testing that all classes have
#identical distribution (or mean)
#    when assigning the new obs to each of the classes
# If the method is 'rank', use Kruskal Wallis test; otherwise, use anova
# Also calculate distance of the new obs to each class median

calculate.p.and.dist=function(classes, obs, method, klist){
 #datv=as.numeric(c( unlist(obs), unlist(classes)))
datv=unlist(obs)
 oldgroup=character()
 for ( j in klist){ oldgroup=c(oldgroup, rep(paste("class", j),
  nrow(classes[[j]])*ncol(classes[[j]]) ) )
datv=c(datv,unlist(classes[[j]]))
}
 p=numeric(); all.abs.dist=numeric()
 for ( i in klist){
      all.abs.dist[i]= abs(obs-median(classes[[i]]) ) #absolute distance
group=c(paste("class", i), oldgroup )
if (method=="rank")  {
p[i]=kruskal.test(datv~group, data=data.frame(datv, group) )$p.value
```

```
} else {
p[i]=anova(lm(datv~group, data=data.frame(datv, group) ))$'Pr(>F)'[1]
}
  }
list(p=p, all.abs.dist=all.abs.dist)
}



#  Decciding two clsses
decide.2=function(p1, p2, class1,class2, obs,threshold=1e-3){
ind=abs(obs-mean(class1)) < abs(obs-mean(class2))
if (ind){
if ((p1 <threshold)&(p2<threshold) ) { result="class 1"}
else{ result= ifelse( (p1<p2), "class 1", "class2") }
} else {
if ((p1 <threshold)&(p2<threshold) ) { result="class 2"}
else{ result= ifelse( (p1<p2), "class 1", "class2") }
 }
result
}



#  Decciding two clsses incorporating prior
decide.2.prior=function(p, classes, obs,threshold=1e-3){
priors=numeric()
# calculate prior prob. If no.prior is necessary, set 1/k for each  prior prob
```

```
k=length(classes)
if (no.prior==T){ priors=1/k} else {
mc=unlist(lapply(classes,median));
mc.order=order(mc)
sorted.mc=sort(mc)
LMC=c(-sorted.mc[1],sorted.mc[-k])
RMC=c(sorted.mc[-1],1e+16)
for (j in 1:k){
#when the variances of the different classes  are different then the two
#terms average in the line below  does not give the  correct partition.
priors[mc.order[j]]=mean((Adat>(LMC[j]+sorted.mc[j])/2)
&(Adat<=(RMC[j]+sorted.mc[j])/2))
 }
}

#ind tells us class that this obs is closet to in terms of no.
#of std.dev. away from each center.
ind=(order(abs(obs-mc)/(unlist(lapply(classes,function(xs)sd(c(xs))))+1e-5)))[1]
if (max(p) <threshold ) { result=paste("class",ind)} else{ result=(order(p*(1-priors))
result
}

#classification of block of pixel on multiclasses with prior
new.obs.class.T.block.multiclass=function(Adat, classes, click=F,
blockpixels=Adat, use.prior=T,threshold=1e-3,Method="rank"){

if (click==T){
#image(Adat, col=gray( (0:254)/255 )   )
```

```
z=unlist(locator(2) )
lines(x=c(z[1],z[2], z[2], z[1],z[1]),y=c(z[3], z[3],
z[4], z[4], z[3]), col="orange", lwd=2  )

obs.x=round(z[1:2]*nrow(Adat))
obs.y=round(z[3:4]*ncol(Adat)  )
obs=Adat[obs.x[1]:   obs.x[2],   obs.y[1]: obs.y[2] ]
} else obs= blockpixels

classres=matrix("NA", nrow=nrow(obs), ncol=ncol(obs) )
for ( i in 1:nrow(obs) ){
 for (j in 1:ncol(obs) ) {
if (use.prior ==T){
classres[i,j]=multiclass.new.obs.class.T.with.prior(Adat, classes,
 method=Method, click=F, obsgiven=obs[i,j], no.prior=F ) } else {
classres[i,j]=multiclass.new.obs.class.T.with.prior(Adat, classes,
method=Method, click=F, obsgiven=obs[i,j], no.prior=T ) }
}
}
list(individual.class=classres, summarys=table(c(classres) ))
}

# classify a point with no prior prob.
# This function is a special case of the other function though command
# new.obs.class.T.with.prior(Adat, class1, class2, method="Wilcox",
#click=T, obsgiven=mean(Adat), no.prior=T )

new.obs.class.T=function(Adat, classes,  method="Wilcox" ,
```

```
click=T, obsgiven=mean(Adat) ){
class1=classes[[1]]; class2=classes[[2]]
if (click==T){
image(Adat, col=gray( (0:254)/255 )   )
z=unlist(locator(1) )
points(z[1], z[2], lwd=2, col="orange")
obs=Adat[round(z[1]*nrow(Adat)),  round(z[2]*ncol(Adat)  )   ]
} else obs= obsgiven

#obs=new.obs(Adat)
temp1=c(class1, obs)
if (method=="Wilcox") mytest=wilcox.test else mytest=t.test
p1=mytest(temp1, c(class2) )$p.value
temp2=c(class2, obs)
p2=mytest( c(class1), temp2 )$p.value

ind=abs(obs-mean(class1)) < abs(obs-mean(class2))
if (ind){
if ((p1 <1e-3)&(p2<1e-3) ) { result="class 1"}
else{ result= ifelse( (p1<p2), "class 1", "class2") }
} else {
if ((p1 <1e-3)&(p2<1e-3) ) { result="class 2"}
else{ result= ifelse( (p1<p2), "class 1", "class2") }
 }
result
}
```

```
new.obs.class.T.with.prior=function(Adat, classes, method="Wilcox",
click=T, obsgiven=mean(Adat), no.prior=F ){

class1=classes[[1]]; class2=classes[[2]]

# calculate prior prob. If no.prior is necessary, set 1/2 for both prior prob
if (no.prior==T){ prior1=prior2=1/2} else {
mc1=mean(class1);   mc2=mean(class2)
if (mc1<mc2) {prior1=mean(Adat< (mc1+mc2)/2); prior2=1-prior1      }
 else {prior2=mean(Adat< (mc1+mc2)/2); prior1=1-prior2      }
}


# choose a point to classify
if (click==T){
image(Adat, col=gray( (0:254)/255 )   )
z=unlist(locator(1) )
points(z[1], z[2], lwd=2, col="orange")
obs=Adat[round(z[1]*nrow(Adat)),  round(z[2]*ncol(Adat)  )   ]
} else obs=obsgiven

# decide which class the point belongs to
temp1=c(class1, obs)
if (method=="Wilcox") mytest=wilcox.test else mytest=t.test
p1=mytest(temp1, c(class2) )$p.value
temp2=c(class2, obs)
p2=mytest( c(class1), temp2 )$p.value
```

```
ind=abs(obs-mean(class1)) < abs(obs-mean(class2))
if (ind){
if ((p1 <1e-3)&(p2<1e-3) ) { result="class 1"} else{ result=
ifelse( (p1*(1-prior1) <p2*(1-prior2)), "class 1", "class2") }
} else {
if ((p1 <1e-3)&(p2<1e-3) ) { result="class 2"} else{ result=
ifelse( (p1*(1-prior1)<p2*(1-prior2)), "class 1", "class2") }
 }
result
}



## classify for a block of pixels
new.obs.class.T.block=function(Adat, classes, click=F,
blockpixels=Adat, use.prior=T){

if (click==T){
image(Adat, col=gray( (0:254)/255 )   )
z=unlist(locator(2) )
lines(x=c(z[1],z[2], z[2], z[1],z[1]),y=c(z[3], z[3],
z[4], z[4], z[3]),
 col="orange", lwd=2  )

obs.x=round(z[1:2]*nrow(Adat))
obs.y=round(z[3:4]*ncol(Adat)  )
obs=Adat[obs.x[1]:   obs.x[2],   obs.y[1]: obs.y[2] ]
```

```
} else obs= blockpixels

classres=matrix("NA", nrow=nrow(obs), ncol=ncol(obs) )
for ( i in 1:nrow(obs) ){
 for (j in 1:ncol(obs) ) {
if (use.prior ==T){
classres[i,j]=new.obs.class.T.with.prior(Adat, classes,
method="Wilcox", click=F, obsgiven=obs[i,j], no.prior=F ) } else {
classres[i,j]=new.obs.class.T.with.prior(Adat, classes,
method="Wilcox", click=F, obsgiven=obs[i,j], no.prior=T ) }
}
}
list(individual.class=classres, summarys=table(c(classres) ))
}



new.obs.class.T.block.poly=function(Adat, classes, click=F,
blockpixels=Adat, use.prior=T){

if (click==T){
image(Adat, col=gray( (0:254)/255 )   )

library(splancs)
n=scan(what=numeric(),nmax=1)
z=unlist(locator(n) )
polygon(x=z[1:n],y=z[(n+1):(2*n)])
myp=data.frame(x=z[1:n],y=z[(n+1):(2*n)])
```

```
x0=seq(nrow(Adat))/nrow(Adat)
y0=seq(ncol(Adat))/ncol(Adat)
indicator.map=inout(data.frame(expand.grid(x0, y0)), myp)
keeps=expand.grid(seq(nrow(Adat)), seq(ncol(Adat)) )[indicator.map,]
obs=Adat[unlist(keeps[,1]), unlist(keeps[,2]) ]

} else obs= blockpixels

classres=matrix("NA", nrow=nrow(obs), ncol=ncol(obs) )
for ( i in 1:nrow(obs) ){
 for (j in 1:ncol(obs) ) {
if (use.prior ==T){
classres[i,j]=new.obs.class.T.with.prior(Adat, classes,
 method="Wilcox", click=F, obsgiven=obs[i,j], no.prior=F ) } else {
classres[i,j]=new.obs.class.T.with.prior(Adat, classes,
method="Wilcox", click=F, obsgiven=obs[i,j], no.prior=T ) }
}
}
list(individual.class=classres, summarys=table(c(classres) ))
}


library("rimage")
#A=read.jpeg("brain.jpg")
A=read.jpeg("C:\\Users\\Santosh\\Documents\\Test images
\\test images\\coloredimage.jpg")
Adat=imagematrix(A, type="grey")
Adat0=imagematrix(A)
```

```
Adat=Adat0[,,3]
#image(Adat)
plot(A)



par(mfrow=c(1,2))
classes=class.data.fun.multiclass(Adat, k=3, histogram=F)
Aka.new.obs.class.T.multiclasses(Adat, classes,
 method="rank" ,  click=T )

fortable=character()
keepobs=numeric()
for (i in 1:20){
commonobs=common.obs.Liao.Aka.ours(Adat0,i)
keepobs=c(keepobs,commonobs)
akares=Aka.new.obs.class.T.multiclasses(Adat, classes,  method="rank" ,
 click=F, obsgiven=commonobs )
ours=new.obs.class.T.3classes(Adat, classes,  method="rank" ,  click=F,
obsgiven=commonobs,threshold=1e-3)
fortable=rbind(fortable, c(i,format(akares$pvalue,digits=4),akares$result,
 format(ours$pvalue,digits=4),round(ours$distance,3),ours$result))
#fortable=rbind(fortable, c(i,format(akares$pvalue,digits=4),akares$result))
}
write.csv(fortable,file="5class.csv")

k0=length(classes)
keeptable=fortable[ ,  -( k0+2+(1:k0))]
 #write.csv(keeptable, file="5class.keep.csv", row.names=F)
```

```
 # keeptable=read.csv("5class.keep.csv")
 # keeptable[, 8:12]=round(keeptable[, 8:12], 3)
colnames(keeptable)=c("obs", paste("PV", 1:k0, sep=""),
 "LA", paste("d", 1:k0, sep=""), "New")
library(xtable)
xtable(as.matrix(keeptable[,-1]))



setwd("C:\\Users\\Santosh\\Documents\\Test images\\test
 images\\riverboat5classes\\newtry")
# output the data to be used by other classification methods
alldata=numeric()

for (i in 1:k0){
alldata=rbind(alldata,cbind(c(classes[[i]]),
rep(i,length(c(classes[[i]])))))
write.csv(classes[[i]], file=paste("eg.five.classes.dat",
i, ".csv", sep="") )
}
colnames(alldata)=c("x","y")
write.table(keepobs,file="testobs.txt",row.names=F)

library(MASS)
ldares=lda(as.factor(y)~x,data=data.frame(alldata))
pred.lda=predict(ldares,newdata=data.frame(x=keepobs))$class



qdares=qda(as.factor(y)~x,data=data.frame(alldata))
```

```
pred.qda=predict(qdares,newdata=data.frame(x=keepobs))$class

library(rpart)
tres=rpart(as.factor(y)~x,data=data.frame(alldata))
pred.tree=predict(tres,newdata=data.frame(x=keepobs),type="class")

#library(mgcv)
#gamres=gam(as.factor(y)~x,data=data.frame(alldata),family=)



alldata=data.frame(alldata)
alldata$y=as.factor(alldata$y)
attach(alldata)
library(polspline)
marsres=polyclass(y, x)
pred.polyclass=cpolyclass(cov=matrix(keepobs), fit=marsres)



allresult=data.frame(keeptable, lda=pred.lda, qda=pred.qda, tree=pred.tree,
 polyclass=pred.polyclass,obsvalue=round(keepobs, 2))
write.csv(allresult, file="allresult.csv", row.names=F)
library(xtable)
xtable(as.matrix(allresult[,-1]))

plot.classes(classes)



#library(mlogit)
```

```
#malldata= mlogit.data(alldata, shape = "wide", choice = "y")
#mlogitres=mlogit(y~x,data=malldata)

fortable=character()
for (i in 1:10){
commonobs=common.obs.Liao.Aka.ours(Adat,i)
akares=Aka.new.obs.class.T.multiclasses(Adat, classes,  method="rank" ,
 click=F, obsgiven=commonobs )
ours=new.obs.class.T.3classes(Adat, classes,  method="rank" ,  click=F,
obsgiven=commonobs,threshold=1e-3)
fortable=rbind(fortable, c(i,akares$result,akares$pvalue, ours$result,
ours$pvalue,format(ours$distance,digits=4)))
}
fortable
```