CANCER INFORMATICS

A Major Project Report

Submitted in partial fulfillment for the award of the degree

of

BACHELOR OF TECHONOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

Rajitha Bhavani Kantheti	**(14SS1A0538)**
Rumandla Mounika	**(14SS1A0539)**
Sree Harshini Veeramalla	**(14SS1A0551)**
Dharmana Jyothi	**(14SS1A0510)**

Under the guidance of

Dr. G. Narsimha

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING
Jawaharlal Nehru Technological University Hyderabad
College of Engineering Sultanpur
Sultanpur (V), Pulkal (M), Sangareddy-502273 Telangana

CERTIFICATE

Date:

This is to certify that the project work entitled **"CANCER INFORMATICS"** is a bonafide work carried out by **RAJITHA BHAVANI KANTHETI (14SS1A0538), RUMANLA MOUNIKA (14SS1A0539), SREE HARSHINI VEERAMALLA(14SS1A0551), DHARMANA JYOTHI (14SS1A0510)** in partial fulfillment of the requirements for the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** by the Jawaharlal Nehru Technological University, Hyderabad during the academic year 2017-2018.

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree.

G.Narsimha **G.Narsimha**

Internal Guide **HOD-CSE dept**

External Guide

DECLARATION

We do declare that the project work entitled **"CANCER INFORMATICS"** submitted by us in the Department of Computer Science And Engineering, JNTUH College of Engineering, Sultanpur in partial fulfillment of degree for the award of Bachelor of Technology in Computer Science and Engineering is a bonafide work, carried out by me under the supervision of Dr.G.Narsimha, HOD, Department of CSE, JNTUHCES.

(Rajitha Bhavani Kantheti-14SS1A0538)

(Rumandla Mounika-14SS1A0539)

(Sree Harshini Veeramalla-14SS1A0551)

(Dharmana Jyothi-14SS1A0510)

ACKNOWLEDGEMENT

We wish to take this opportunity to express our deep gratitude to all those who helped, encouraged, motivated and have extended their cooperation in various ways during our project work. It is our pleasure to acknowledge the help of all those individuals who were responsible for foreseeing the successful completion of our project.

We express our sincere gratitude to **Dr. B. BALU NAIK,** Principal of JNTUHCES for his encouragement and providing facilities to accomplish our project successfully.

We thank our Vice Principal **Dr. V. VENKATESHWAR REDDY** for extending his help and cooperation.

We are thankful to **Dr. G. NARSIMHA,** Head of The Department of Computer Science And Engineering, JNTUHCES and a senior faculty member as well as a supervisor, for his encouragement and guidance for preparing and presenting seminar.

We are indebted to our supervisor **Mr. JOSHI SHRIPAD,** Assistant professor in the department of CSE, JNTUHCES for his valuable advice and help throughout the development of this project by providing us with required information without whose guidance, cooperation and encouragement, this project couldn't have been materialized.

It is our pleasure to thank our co-guide **Ms. NEERAJA**, Department of CSE, JNTUHCES for her help and encouragement during the project work.

Last but not least, we express our gratitude with great admiration and respect to all Department Staff and Lab Assistants for their moral support and encouragement throughout the course.

(Rajitha Bhavani Kantheti)

(Rumandla Mounika)

(Sree Harshini Veeramalla)

(Dharmana Jyothi)

ABSTRACT

Over the past decades, a continuous evolution related to cancer research has been performed. Scientists applied different methods, such as screening in early stage, in order to find types of cancer before they cause symptoms. Moreover, they have developed new strategies for the early prediction of cancer treatment outcome. With the advent of new technologies in the field of medicine, large amounts of cancer data have been collected and are available to the medical research community. However, the accurate prediction of a disease outcome is one of the most interesting and challenging tasks for physicians. As a result, ML methods have become a popular tool for medical researchers. These techniques can discover and identify patterns and relationships between them, from complex datasets, while they are able to effectively predict future outcomes of a cancer type.

Given the significance of personalized medicine and the growing trend on the application of ML techniques, we here present a review of studies that make use of these methods regarding the cancer detection and diagnosis. In these studies detective features are considered which may be independent of a certain treatment or are integrated in order to guide therapy for cancer patients, respectively.

It is clear that the application of ML methods could improve the accuracy of cancer susceptibility, recurrence and survival prediction. The accuracy of cancer prediction outcome has significantly improved by 15%–20% the last years, with the application of ML techniques.

Several studies have been reported in the literature and are based on different strategies that could enable the early cancer diagnosis and prognosis. Specifically, these studies describe approaches related to the profiling of circulating miRNAs that have been proven a promising class for cancer detection and identification. However, these methods suffer from low sensitivity regarding their use in screening at early stages and their difficulty to discriminate benign from malignant tumors. These studies list the potential as well as the limitations of microarrays for the prediction of cancer outcome. Even though gene signatures could significantly improve our ability for prognosis in cancer patients, poor progress has been made for their application in the clinics. However, before gene expression profiling can be used in clinical practice, studies with larger data samples and more adequate validation are needed.

In the present work only studies that employed ML techniques for modeling cancer diagnosis and prognosis are presented. The importance of classifying cancer patients into high or low risk groups has led many research teams, from the biomedical and the bioinformatics field, to study the application of machine learning (ML) methods. Therefore, these techniques have been utilized as an aim to model the progression and treatment of cancerous conditions. In addition, the ability of ML tools to detect key features from complex datasets reveals their importance.

Even though it is evident that the use of ML methods can improve our understanding of cancer progression, an appropriate level of validation is needed in order for these methods to be considered in the everyday clinical practice.

Till now there were many projects on cancer detection using ML where pathologists are involved in reviewing the pathology slides. There by, features are extracted manually by pathologists. This may lead into two problems, misdiagnosis and late diagnosis. But In this work, we are replacing the role of pathologists and we present the detection of cancer using SVM, resulting in effective and accurate decision making.

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

In developing and developed countries, along with heart disease, cancer is the second leading cause of death. Rebecca L. et al. estimated that there would be 600,920 deaths and 1,688,780 new cases of cancer in the United States by 2017. Compared to the prevention of heart disease, the underlying causes and possible mechanisms for cancer are not well understood. As a result, cancer will become the most common cause of death over time, predicted by several experts. Despite numerous times of exertion, upgrades in cancer treatment have not completely satisfied desires. As a regular case from many that could be mentioned, a form of cancer with a very poor prognosis, a multi-country initiative on colorectal cancer, a 0.5% improvement per year resulted in a change in relative survival of five years rate of 59% to 65%, over twelve years. Uncontrolled growth of malignant cells causes cancer that contains mutations in a person's normal DNA. Malignant cells undergo normal cells for some time. A remarkable source of difficulty in influencing malignancy as a disease is its confusing assortment. Even when they occur on the same site, there are not two manifestations of cancer alike. There may be multiple mutations of normal DNA within a single cancer tumor.

Current research and treatment techniques address what may be referred to as "dominant" mutations within a tumor, as these are the most recognized. Independently of the possibility that the treatment is to kill all dominant mutant cells, other transformations will eventually evolve. This may be the reason why, in most cases, although cancer therapy seems to work, the tumor decreases or may even become undetectable after a time. When the tumor is revealed and shows a new growth explosion. In addition, the recurrent tumor is regularly impermeable to the treatment already applied. Since cancer manifestations vary considerably from one person to another, cancer is an ideal candidate for a "personalized prescription," in which the treatment is tailored for each patient.

A pathologist’s report after reviewing a patient’s biological tissue samples is often the gold standard in the diagnosis of many diseases. For cancer in particular, a pathologist’s diagnosis has a profound impact on a patient's therapy. The reviewing of pathology slides is a very complex task, requiring years of training to gain the expertise and experience to do well.

Even with this extensive training, there can be substantial variability in the diagnoses given by different pathologists for the same patient, which can lead to misdiagnoses. For example, agreement in diagnosis for some forms of breast cancer can be as low as 48 per cent, and similarly low for prostate cancer. The lack of agreement is not surprising given the massive amount of information that must be reviewed in order to make an accurate diagnosis. Pathologists are responsible for reviewing all the biological tissues visible on a slide. However, there can be many slides per patient, each of which is 10+ gigapixels when digitized at 40X magnification. Imagine having to go through a thousand 10 megapixel (MP) photos, and having to be responsible for every pixel. Needless to say, this is a lot of data to cover, and often time is limited.

To address these issues of limited time and diagnostic variability, we created an automated detection algorithm that can naturally complement pathologists' workflow.

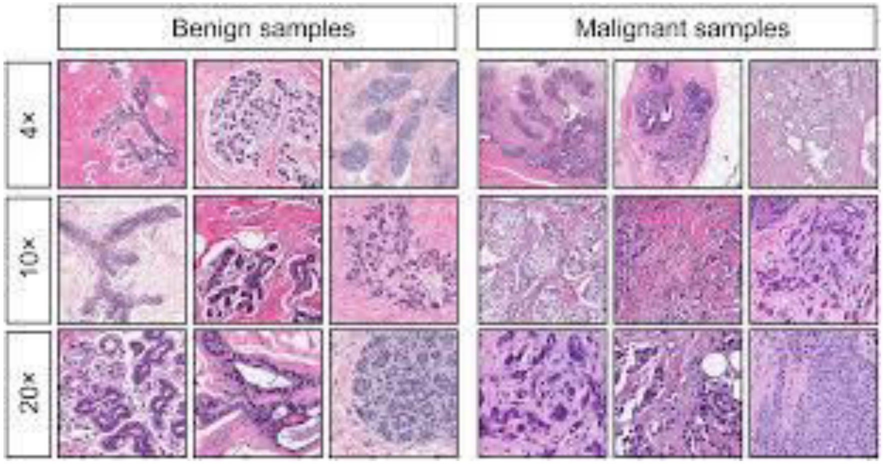


Figure 1.1

1.1. Literature Survey

Image pre-processing is one of the preliminary steps which are highly required to ensure the high accuracy of the subsequent steps. The raw histopathology images normally consist of many artifacts such as intensity inhomogenities, extra cranial tissues, etc. which reduces the overall accuracy.

Several researches are reported in the literature to minimize the effects of artifacts in the histopathology images. Enhancement is the process of manipulating an image so that the result is more suitable than the original image. It is used to extract the useful information. Filtering techniques are applied to remove the noise in the image. Noise removal is most important in the medical image analysis. The most frequently affected noises in the medical images are Gaussian, pepper, speckle and Poisson noises. To remove the noises from the input image and restore the quality of the original image is called as image restoration. This is most efficiently done by using CLAHE algorithm. In this function, the pixel value is restored. An analysis on filtering techniques with CLAHE for noise reduction is performed by Nicu et al (2000).

Chunyan et al (2004) have implemented the colour ray casting method to differentiate the region of interest from the background. But this technique is image dependent and is only applicable for gray level images. Expectation Maximization Segmentation (EMS) software package is used by Hayit et al (2006) for image pre-processing. The main advantage of this technique is that it is a fully automatic technique. Diffusion filtering combined with simple non-adaptive intensity thresholding is used by Yong et al (2006) to enhance the region of interest. The main drawback of this technique is the non-adaptive nature of the threshold value. Fuzzy connectedness based intensity non uniformity correction has been implemented by Yongxin et al (2006). A sequential approach with fuzzy connectedness, atlas registration and bias field correction is used in this approach. The conclusions revealed that the proposed technique can be used only if the intensity variations between the images are of a limited range.

Marianne et al (2006) have minimized the effects of inter-slice intensity variation with the weighted least square estimation method. The selection of weights for the least square method is the major disadvantage of this approach. Bo et al (2008) have proposed the noise removal technique using wavelets and curvelets. Hybrid approaches involving Variance Stabilizing Transform (VST) are also used in this work. But this technique is applicable for images with Poisson noise. Tracking algorithm based de-noising technique is performed by Jaya et al (2009). Since the seed point for tracking is random in nature, the efficiency of this technique is low. A contrast agent accumulation model based contrast enhancement is implemented by Marcel et al (2009). This improves only the contrast of the image and the unwanted tissues are not eliminated. Rajeev et al (2009) have used the wiener filtering methodologies for noise removal in abnormal MR brain images. Apart from noise removal, several 12 other pre-processing steps are also reported in the literature. This includes image format conversion, image type conversion etc. Rajeev et al (2009a) also have used the combination of three modalities of patholgy images for further processing. All the above mentioned techniques remove only specific artifacts which is not sufficient for high classification accuracy and segmentation efficiency.

R. Burbidge et. al., have shown that the support vector machine (SVM) classification algorithm, proves its potential for structure–activity relationship analysis. In a benchmark test, they compared SVM with various machine learning techniques currently used in this field. Among three tested artificial neural networks, they found that SVM is significantly better than all of these. Giorgio Valentini [32] have proposed classification methods, based on linear SVM , and output coding (OC), ensembles of learning machines to separate normal from malignant tissues, to classify different types of lymphoma and to analyze the role of sets of coordinately expressed genes in carcinogenic processes of lymphoid tissues. By using gene expression data from ''Lymphochip'', he has shown that SVM can correctly separate the tumoural tissues, and OC ensembles can be successfully used to classify different types of lymphoma. Shutao Li et. al., have applied SVMs by taking DWFT as input for classifying texture, using translation-invariant texture features. They used a fusion scheme based on simple voting among multiple SVMs, each with a different setting of the kernel parameter, to alleviate the problem of selecting a proper value for the kernel parameter in SVM training and performed the experiments on a subset of natural textures from the Brodatz album. They claim that, as compared to the traditional Bayes classier and LVQ, SVMs, in general, produced more accurate classification results.

A training method to increase the efficiency of SVM has been presented by Yiqiang Zhan [34] for fast classification without system degradation. Experimental results on real prostate ultrasound images show good performance of their training method in discriminating the prostate tissues from other tissues and they claim that their proposed training method is able to generate more efficient SVMs with better classification abilities.

Yuchun Tang et. al., [35] have developed an innovative learning model called granular support vector machines for data classification problems by building just two information granules in the top-down way. The experiment results on three medical binary classification problems show that granular support vector machines proposed in their work provides an interesting new mechanism to address complex classification problems, which are common in medical or biological information processing applications. Bo-Suk Yang et. al., [36] have presented a novel scheme to detect faulty products at semi-product stage in an automatic mass product line of reciprocating compressors for small-type refrigerators used in family electrical appliances. They presented the classification accuracy using the

ANNs, SVM, LVQ, SOFM and SOFM with LVQ (SOFM-LVQ) and found SOFM-LVQ gives high accuracy and are the best techniques for classifying healthy and faulty conditions of small reciprocating compressors.

The result shows SOFM with LVQ can improve the classification performance of SOFM but cannot eliminate the classification error, indicated in the concluding remarks. Rung-Ching Chen [37] has proposed a web page classification method for extraction of feature vectors from both the LSA and WPFS methods by using a SVM based on a weighted voting schema. The LSA classifies semantically related web pages, offering users more complete information. The experimental results show that the anova kernel function yields the best result of these four kernel functions.

1.2. Effects of Late diagnosis

Almost half of people who get cancer are diagnosed late, which makes treatment less likely to succeed and reduces their chances of survival. Cancer survival is an important issue but delayed diagnosis can also have a negative effect on quality of life, with the use of more toxic treatments when cancer is diagnosed at an advanced stage and an increase in psychological distress. Once a cancer has spread, it is often harder to treat successfully, meaning that a person's chances of surviving are much lower.

A total of 2299 new cancer patients were referred to the Northern Israel Oncology Center in 1974 and in 1980. The stage of disease, delay in diagnosis, the responsibility for the delay, and the survival of those referred in 1974 were investigated. At the time of diagnosis, 39% of the patients had localized disease, 34% had locally advanced disease, and 23% had metastatic disease. In 52% of the patients there was no delay in diagnosis. No correlation was found in the group as a whole between the stage of disease and delay in diagnosis. Only in the breast cancer group without delay in diagnosis, however, were there significantly more patients at an early stage than at an advanced stage of disease. At each stage of disease, responsibility for the delay was shared about equally between the patients and the physicians, except in advanced breast cancer, where the patients were more often responsible for the delay. The survival rate was higher in patients in whom the disease was diagnosed earlier. It was also higher at each clinical stage (Stages I and II) in patients who had no delay than in those with delay in diagnosis. The survival rate was higher in patients who were themselves responsible for the delay in diagnosis than in patients whose physicians were responsible for the delay. In 1980, less diagnosis were delayed in fewer patients than in 1974 (42% versus 65%). Responsibility for the delay in 1980 lay equally with the patients and with the physicians, but when compared to 1974, the physicians' responsibility and administrative delay were less. Campaigns for early diagnosis are advocated.

1.3. What are Pathology Slides?

Histopathology refers to the microscopic examination of tissue in order to study the manifestations of disease. Specifically, in clinical medicine, histopathology refers to the examination of a biopsy or surgical specimen by a pathologist, after the specimen has been processed and histological sections have been placed onto glass slides. In contrast, cytopathologyexamines (1) free cells or (2) tissue micro-fragments.

Histopathological examination of tissues starts with surgery, biopsy, or autopsy. The tissue is removed from the body or plant, and then...often following expert dissection in the fresh state...placed in a fixative which stabilizes the tissues to prevent decay. The most common fixative is formalin (10% neutral buffered formaldehyde in water).

The tissue is then prepared for viewing under a microscope using either chemical fixation or frozen section.

If a large sample is provided e.g. from a surgical procedure then a pathologist looks at the tissue sample and selects the part most likely to yield a useful and accurate diagnosis - this part is removed for examination in a process commonly known as grossing or cut up. Larger samples are cut to correctly situate their anatomical structures in the cassette. Certain specimens (especially biopsies) can undergo agar pre-embedding to assure correct tissue orientation in cassette & then in the block & then on the diagnostic microscopy slide. This is then placed into a plastic cassette for most of the rest of the process.

In addition to formalin, other chemical fixatives have been used. But, with the advent of immunohistochemistry (IHC) staining and diagnostic molecular pathology testing on these specimen samples, formalin has become the standard chemical fixative in human diagnostic histopathology. Fixation times for very small specimens are shorter, and standards exist in human diagnostic histopathology.

Processing

Water is removed from the sample in successive stages by the use of increasing concentrations of alcohol. Xylene is used in the last dehydration phase instead of alcohol - this is because the wax used in the next stage is soluble in xylene where it is not in alcohol allowing wax to permeate (infiltrate) the specimen. This process is generally automated and done overnight. The wax infiltrated specimen is then transferred to an individual specimen embedding (usually metal) container. Finally, molten wax is introduced around the specimen in the container and cooled to solidification so as to embed it in the wax block. This process is needed to provide a properly oriented sample sturdy enough for obtaining a thin microtome section(s) for the slide.

Once the wax embedded block is finished, sections will be cut from it and usually placed to float on a waterbath surface which spreads the section out. This is usually done by hand and is a skilled job (histotechnologist) with the lab personnel making choices about which parts of the specimen microtome wax ribbon to place on slides. A number of slides will usually be prepared from different levels throughout the block. After this the thin section mounted slide is stained and a protective cover slip is mounted on it. For common stains, an automatic process is normally used; but rarely used stains are often done by hand.

The second method of histology processing is called frozen section processing. This is a highly technical scientific method performed by a trained histoscientist (a Medical Laboratory scientist specialist in Histopathology. 5–6 years degree training) In this method, the tissue is frozen and sliced thinly using a microtome mounted in a below-freezing refrigeration device called the cryostat. The thin frozen sections are mounted on a glass slide, fixed immediately & briefly in liquid fixative, and stained using the similar staining techniques as traditional wax embedded sections. The advantages of this method is rapid processing time, less equipment requirement, and less need for ventilation in the laboratory. The disadvantage is the poor quality of the final slide. It is used in intra-operative pathology for determinations that might help in choosing the next step in surgery during that surgical session (for example, to preliminarily determine clearness of the resection margin of a tumor during surgery).

Staining

This can be done to slides processed by the chemical fixation or frozen section slides. To see the tissue under a microscope, the sections are stained with one or more pigments. The aim of staining is to reveal cellular components; counterstains are used to provide contrast.

The most commonly used stain in histopathology is a combination of hematoxylin and eosin (often abbreviated H&E). Hematoxylin is used to stain nuclei blue, while eosin stains cytoplasm and the extracellular connective tissue matrix pink. There are hundreds of various other techniques which have been used to selectively stain cells. Other compounds used to color tissue sections include safranin, Oil Red O, congo red, silver salts and artificial dyes. Histochemistry refers to the science of using chemical reactions between laboratory chemicals and components within tissue. A commonly performed histochemical technique is the Perls' Prussian blue reaction, used to demonstrate iron deposits in diseases like Hemochromatosis.

Recently, antibodies have been used to stain particular proteins, lipids and carbohydrates. Called immunohistochemistry, this technique has greatly increased the ability to specifically identify categories of cells under a microscope. Other advanced techniques include in situ hybridization to identify specific DNA or RNA molecules. These antibody staining methods often require the use of frozen section histology. These procedures above are also carried out in the laboratory under scrutiny and precision by a trained specialist Medical laboratory scientist (Histoscientist) Digital cameras are increasingly used to capture histopathological images.

The histological slides are examined under a microscope by a pathologist, a medically qualified specialist who has completed a recognised training program. This medical diagnosisis formulated as a pathology report describing the histological findings and the opinion of the pathologist. In the case of cancer, this represents the tissue diagnosis required for most treatment protocols. In the removal of cancer, the pathologist will indicate whether the surgical margin is cleared, or is involved (residual cancer is left behind). This is done using either the bread loafing or CCPDMA method of processing.

1.4. Machine Learning Techniques

Machine learning is a part of artificial intelligence, relates the issue of gaining from information tests to the general idea of induction. Each learning procedure comprises of two stages: first, estimation of obscure conditions in a framework from a given dataset and second, utilization of assessed conditions to anticipate new yields of the framework. Machine learning has also been shown to be an intriguing territory in biomedical research with many applications, where satisfactory speculation is obtained through an ndimensional space for a given arrangement of organic examples, using systems and algorithms. There are two primary basic sorts of machine learning strategies known as supervised learning and unsupervised learning. In supervised learning, a named set of preparing information is utilized to gauge or guide the information to the coveted yield.

On the other hand, under the unsupervised learning strategies, no named illustrations are given, and there is no thought of the yield amid the learning procedure. Subsequently, it is up to the learning plan/model to find designs or finds the gatherings of the info information. This procedure can be considered as a classification problem in supervised learning. The classification task refers to a learning process that classifies the data into a set of finite classes. Clustering and regression are the two other common machine learning tasks. A learning function maps the data into a real-valued variable in the regression problems. Clustering is a typical unsupervised assignment in which one tries to find the classifications or groups keeping in mind the end goal to portray the information things. In light of this procedure, each new example can be appointed to one of the identified clusters concerning the comparative qualities that they share.

Another kind of machine learning strategies that have generally been connected is semi-supervised learning, which is a blend of supervised and unsupervised learning. To build an accurate learning model, it combines marked and unmarked data. This sort of learning is utilized when there are more unlabeled datasets than marked. Data samples are the basic components while applying a machine learning strategy. Each sample is depicted with a few elements, and each component comprises of various sorts of values. Besides, knowing ahead of time the specific kind of information being utilized permits the correct determination of tools and techniques that can be utilized for their investigation. A few information related issues allude to the nature of the information and the preprocessing ventures to make them include the presence of aberrant, missing or duplicated sound data and biased, unrepresentative data. While enhancing the data quality, commonly the nature of the subsequent examination is likewise made strides.

Likewise, to make the raw data more appropriate for further examination, preprocessing steps ought to be connected that emphasis on the modification of the data.

There are a number of different techniques and strategies relevant to data pre-processing that focus on modifying data for better fixation in a specific machine learning method. Among these approaches the absolute most critical methodologies incorporate reducing dimensionality, selecting features, and extracting feature. There are many advantages in reducing dimensionality when data sets have a large number of features. When the dimension is lower, machine learning works better. In addition, due to the involvement of fewer features, reducing dimensionality can eliminate irrelevant features, reduce noise and produce more robust learning models. The dimensionality diminishment by choosing new features which are a subset of the old ones is known as feature selection.

At the point when the data are preprocessed, and we have defined the sort of learning assignment, a rundown of machine learning techniques including Artificial Neural Networks (ANNs), Bayesian Networks (BNs), Support Vector Machines (SVMs) and Decision Trees (DTs) is available. Based on the intensity of this survey, we will refer only these machine learning techniques that have been widely used in the literature for a case study of cancer prediction and prognosis. We distinguish the patterns in regards to the sorts of machine learning strategies that are utilized, the sorts of information that are incorporated and in addition the assessment techniques utilized for surveying the general execution of the techniques utilized for growth expectation or illness results.

1.5. Problem Statement

For cancer, pathologist's diagnosis has a profound impact on a patient's therapy. Reviewing of pathology slides is a very complex task, it requires years of training to gain the expertise and experience to do well. Even with this extensive training, there can be substantial variability in the diagnoses given by different pathologists for the same patient, which can lead to misdiagnoses.

To solve late diagnosis and misdiagnosis problems we are creating an automated detection algorithm that can naturally complement pathologists' workflow. A variety of techniques, including Artificial Neural Networks (ANNs), Bayesian Networks (BNs), Support

Vector Machines (SVMs) and Decision Trees (DTs) can be applied in cancer research for the development of predictive models, but we are using SVM for an effective and accurate decision making.

1.6. Existing Solutions

At present, the diagnosis is done manually by the pathologist. Detecting cancer is a multi-stage process. Often, the patient will go to a doctor because of some symptom or other. Sometimes cancer is discovered by chance or from screening. The final cancer diagnosis is based on a pathologist's opinion. Diagnosis ivolves the following steps:

- The organ or tissue biopsied
- Specific part of the organ the sample came from
- The biopsy procedure
- Specific findings in the tissue
- Other important results
- Whether other tests are needed

After processing these steps the pathologist suggests the medication and sends the reports to the doctors.

1.7. Software Requirements

1.7.1Matlab Introduction

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

What is MATLAB?

MATLAB is a high-performance language for technical computing. It integrates computation, visualization and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Math and computation
- Algorithm development
- Modeling, simulation and prototyping
- Data analysis, exploration and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

Why Matlab

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTAN. The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB uses software developed by the LAPACK and ARPACK projects, which together represent the state-of-the-art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering and science. In industry, MATLAB is the tool of choice for high-productivity research,development and analysis.

MATLAB features a family of application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation and many others.

The MATLAB system consists of five main parts:

Development Environment

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical interfaces. It includes the MATLAB desktop and Command Window, a command history and browsers for viewing help, the workspace, files and the search path.

The MATLAB Mathematical Function Library

This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions and fast Fourier transforms.

The MATLAB Language

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output and object-oriented programming features. It allows both “programming in the small” to rapidly create quick and dirty throw-away programs and “programming in the large” to create complete large and complex application programs.

Handle Graphics

This is the MATLAB graphics system. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation and presentation graphics. It also includes low-level commands that allow you to fully customize the appearance of graphics as well as to build complete graphical user interface on your MATLAB applications.

The MATLAB Application Programming Interface(API)

This is a library that allows you to write C and FORTAN programs that interact with MATLAB. It include facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine and for reading and writing MAT-files.

1.7.2. Development Environment

Introduction to starting and quitting MATLAB and the tools and functions that helps you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation which is available online as well as print.

Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop. On a UNIX platform, to start MATLAB, type MATLAB at the operating system prompt. After starting MATLAB, the MATLAB desktop opens – see MATLAB Desktop. You can change the directory in which MATLAB starts, define startup options including running a script upon startup and reduce startup time in some situations.

Quitting MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, suchas saving the workspace, you can create and run a finish m script.

MATLAB Desktop

When you start MATLAB ,the MATLABdesktop appears, containing tools (graphical user interfaces) for managing files, variables and applications associated with MATLAB. The first name MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries. You can change the way your desktop looks by opening, closing, moving and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts. You can specify certain characteristics forthe desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

Current Directory Browser

Workspace Browser

Array Editor

Editor/Debugger

Command Window

Command History

Launch Pad

Help Browser

1.8. Competitors

- **Google AI to detect breast cancer**

Google has successfully applied deep learning artificial intelligence algorithms to the diagnosis of breast cancer.In a study carried out by researchers taking part in Google's Brain Residency Program – a 12-month educational course in machine and deep learning – an algorithm was trained to detect breast cancer tumours in a dataset of digitised pathology slides provided by Dutch medical institute the Radboud University Medical Center.

After 'training' the algorithm, researchers were able to achieve a 92% sensitivity in picking out tumour cells from the slides – significantly higher than the 73% achieved by trained pathologists with no time constraint.

In addition, the team recreated the accuracy in different datasets taken from other hospitals and scanning machinery.The team did report an average of eight false positive per slide compared to none from trained pathologists. However, this rate was lowered through further customisation of the algorithm.

The algorithm results display as a heat map (shown below) that overlay a particular colour in regards to the likelihood of a specific part of an image housing cancerous cells.

- **Microsoft is trying to implement cancer detection and prediction using ML**

At Microsoft's research labs around the world, computer scientists, programmers, engineers and other experts are trying to crack some of the computer industry's toughest problems, from system design and security to quantum computing and data visualization.

A subset of those scientists, engineers and programmers have a different goal: They're trying to use computer science to solve one of the most complex and deadly challenges humans face: Cancer.

And, for the most part, they are doing so with algorithms and computers instead of test tubes and beakers.

"We are trying to change the way research is done on a daily basis in biology," said Jasmin Fisher, a biologist by training who works in the programming principles and tools group in Microsoft's Cambridge, U.K., lab.

One team of researchers is using machine learning and natural language processing to help the world's leading oncologists figure out the most effective, individualized cancer treatment for their patients, by providing an intuitive way to sort through all the research data available. Another is pairing machine learning with computer vision to give radiologists a more detailed understanding of how their patients' tumors are progressing.

Yet another group of researchers has created powerful algorithms that help scientists understand how cancers develop and what treatments will work best to fight them.

And another team is working on moonshot efforts that could one day allow scientists to program cells to fight diseases, including cancer.

CHAPTER 2
ALGORITHMS

2.1. Supervised Learning

Supervised learning is a type of machine learning algorithm that uses a known dataset (called the training dataset) to make predictions. The training dataset includes input data and response values. From it, the supervised learning algorithm seeks to build a model that can make predictions of the response values for a new dataset. A test dataset is often used to validate the model. Using larger training datasets often yield models with higher predictive power that can generalize well for new datasets.

Supervised learning includes two categories of algorithms:

- Classification
- Regression

2.2. Classification

Classification is for categorical response values, where the data can be separated into specific “classes”. Common classification algorithms include:

- Support vector machines (SVM)
- Neural networks
- Naïve Bayes classifier
- Decision trees
- Discriminant analysis
- Nearest neighbors (kNN)

2.3. SVM Classifier

For a dataset consisting of features set and labels set, an SVM classifier builds a model to predict classes for new examples. It assigns new example/data points to one of the classes. If there are only 2 classes then it can be called as a Binary SVM Classifier.

There are 2 kinds of SVM classifiers:

- Linear SVM Classifier
- Non-Linear SVM Classifier

SVM Linear Classifier:

In the linear classifier model, we assumed that training examples plotted in space. These data points are expected to be separated by an apparent gap. It predicts a straight hyperplane dividing 2 classes. The primary focus while drawing the hyperplane is on maximizing the distance from hyperplane to the nearest data point of either class. The drawn hyperplane called as a maximum-margin hyperplane.

SVM Non-Linear Classifier:

In the real world, our dataset is generally dispersed up to some extent. To solve this problem separation of data into different classes on the basis of a straight linear hyperplane can't be considered a good choice. For this Vapnik suggested creating Non-Linear Classifiers by applying the kernel trick to maximum-margin hyperplanes. In Non-Linear SVM Classification, data points plotted in a higher dimensional space.

2.4. Algorithm Comparision

Lets start with logistic regression. Many of us are confused about shape of decision boundary given by a logistic regression. This confusion mainly arises because of looking at the famous S-shaped curve too many times in context of logistic regression.

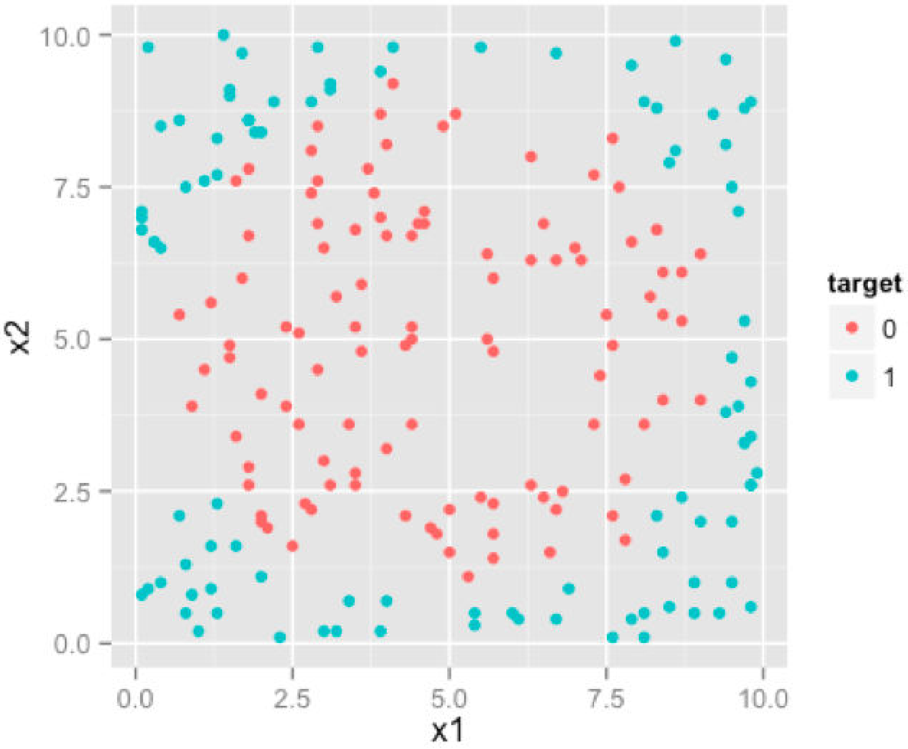


Figure 2.1

This blue curve that you see is not a decision boundary. Its simply in a way is transformed response from binary response which we model using logistic regression. Decision boundary of logistic regression is always a line [or a plane , or a hyper-plane for higher dimension]. Best way to convince you will be , by showing the famous logistic regression equation that you are all too familiar with.

$$\log(\frac{p}{1+p}) = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 \ldots.$$

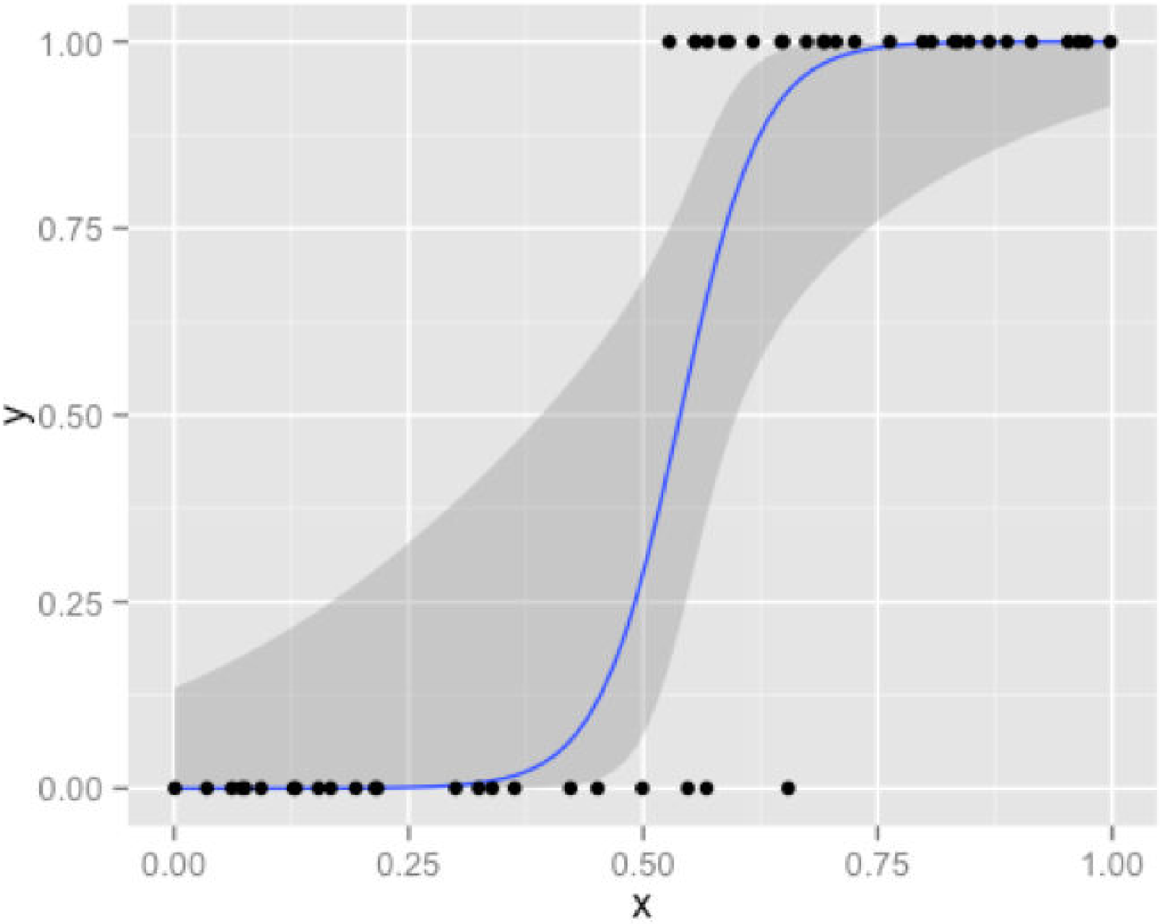


Figure 2.2

Let’s assume for simplification, F is nothing but a linear combination of all the predictors .

$$F = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 \ldots.$$

The above equation can also be written as :

$$p = \frac{1}{1 + e^{-F}}$$

Now to predict in logistic regression you decide a particular score cutoff for the probabilities, above which your prediction will be 1 or 0 otherwise. Lets say that cutoff is c. so your decision process will be like this :

Y=1 if p>c , otherwise 0. Which eventually gives the decision boundary F > constant.

F>constant, here is nothing but a linear decision boundary . Result of logistic regression for our sample data will be like this. You can see that, it doesn't do a very good job. Because whatever you do, decision boundary produced by logistic regression will always be linear , which can not emulate a circular decision boundary which is required. So, logistic regression will work for classification problems where classes are approximately linearly separable. [Although you can make classes linear separable in some cases through variable transformation, but we'll leave that discussion for some other day].

Now lets see how decision trees handle these problems. We know that decision trees are made of hierarchical one variable rules . Such an example for our data is given below.

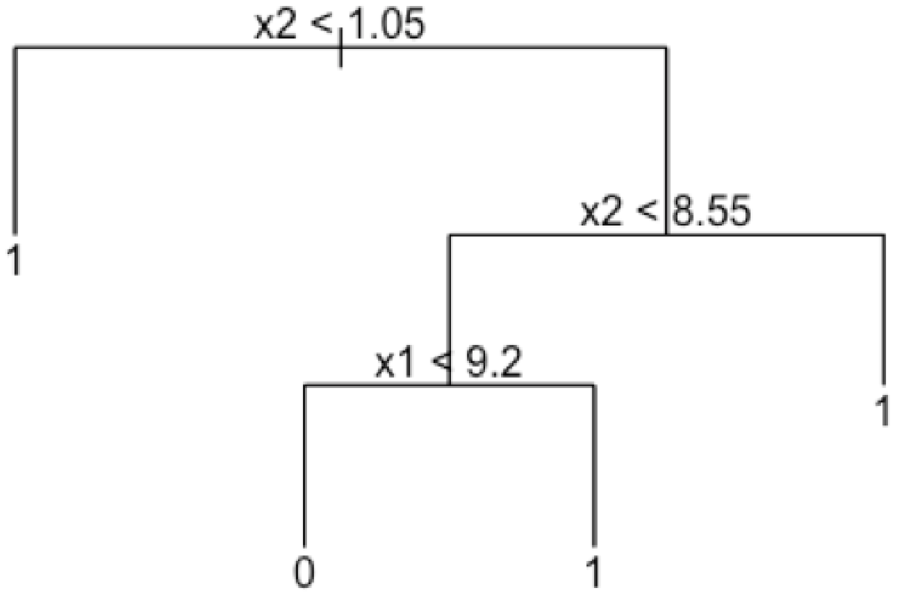


Figure 2.3

If you think carefully, these decision rules x2 |</>| const OR x1 |</>| const do nothing but partition the feature space with lines parallel to each feature axis like the diagram given below.

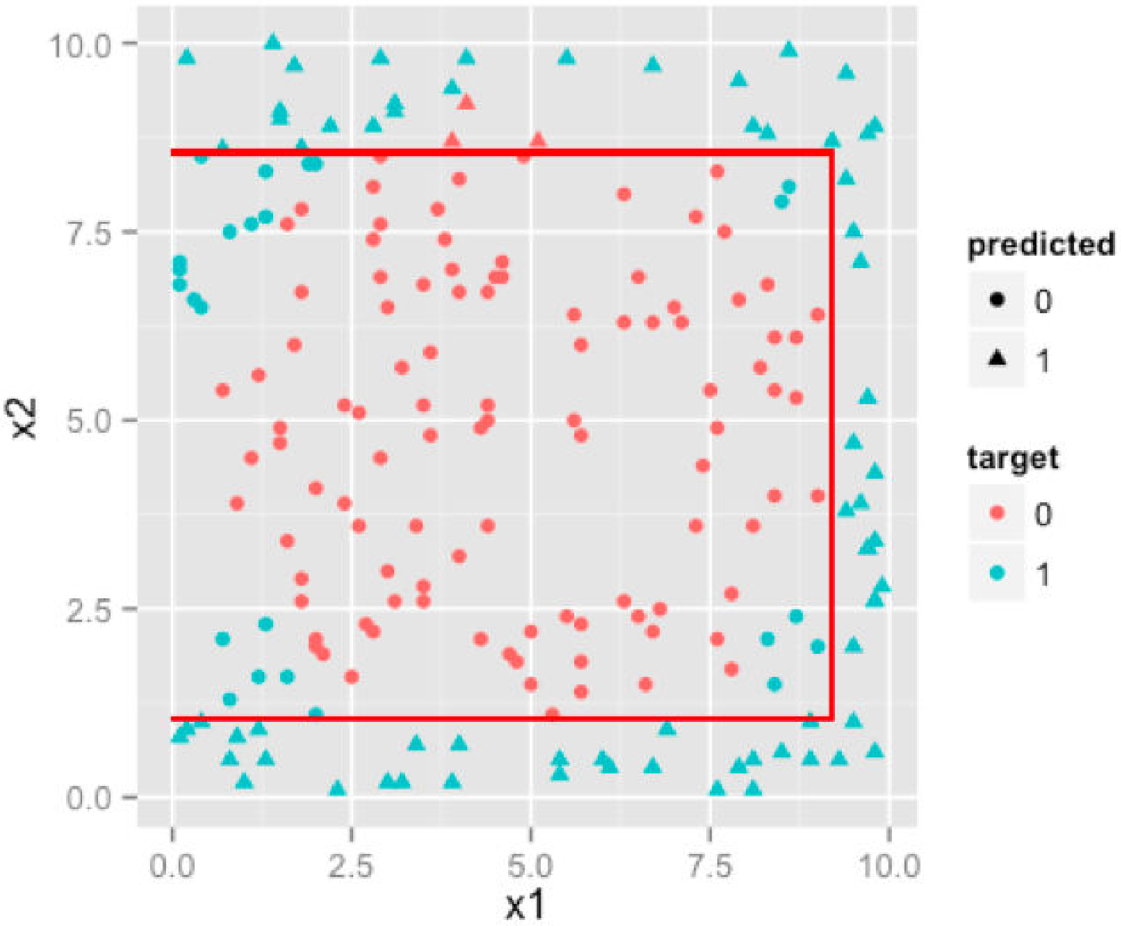


Figure 2.4

We can make our tree more complex by increasing its size , which will result in more and more partitions trying to emulate the circular boundary.

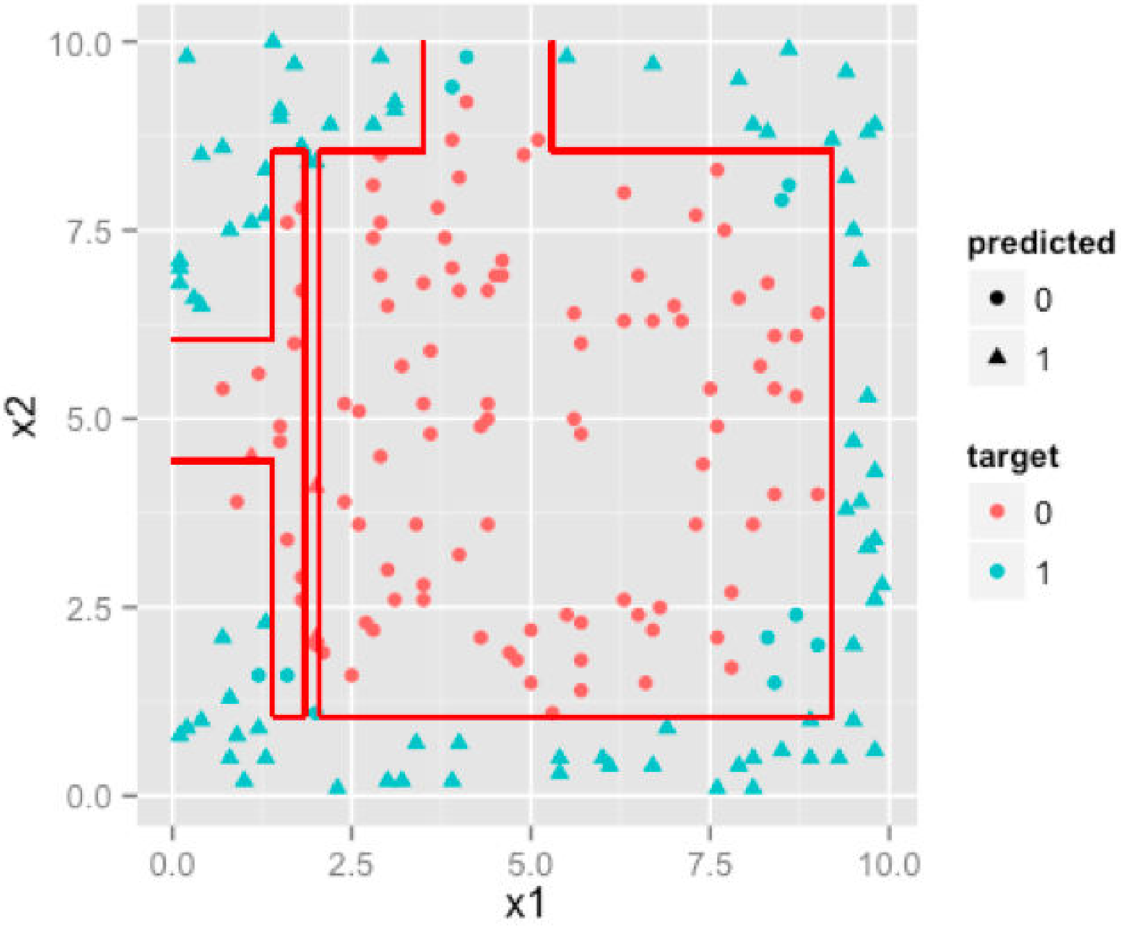


Figure 2.5

Not a circle but it tried, that much credit is due. If you keep on increasing size of the tree , you'd notice that decision boundary will try to emulate circle as much as it can with parallel lines.So, if boundary is non-linear and can be approximated by cutting feature space into rectangles [or cuboids or hyper-cuboid for higher dimensions] then D-Trees are a better choice than logistic regression.

Next we'll look at result of SVM. SVM works by projecting your feature space into kernel space and making the classes linearly separable. An easier explanation to that process would be that SVM adds an extra dimension to your feature space in a way that makes classes linearly separable. This planar decision boundary when projected back to original feature space emulates non linear decision boundary . Here this picture might explain better than me.

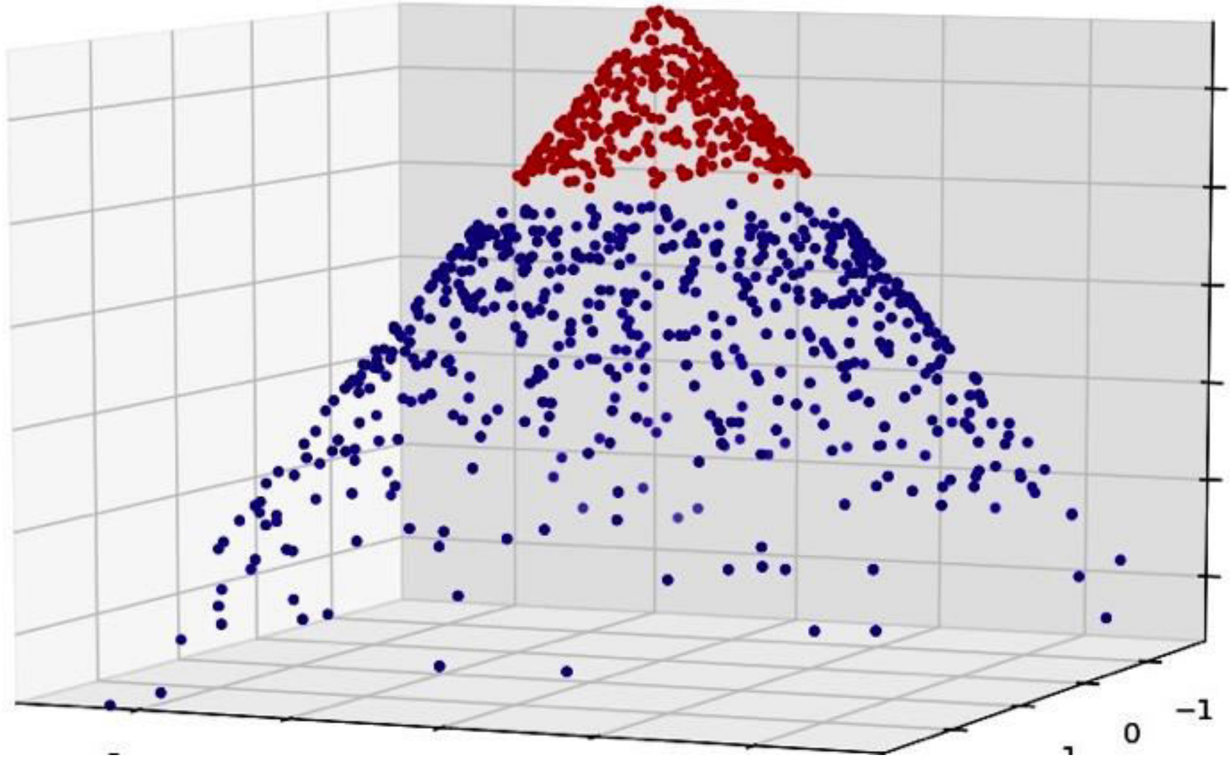


Figure 2.6

You can see that , once a third dimension in a special manner added to data , we can separate two classes with a plane [a linear separator], which once projected back onto the original 2-D feature space; becomes a circular boundary.

see how well SVM performs on our sample data:

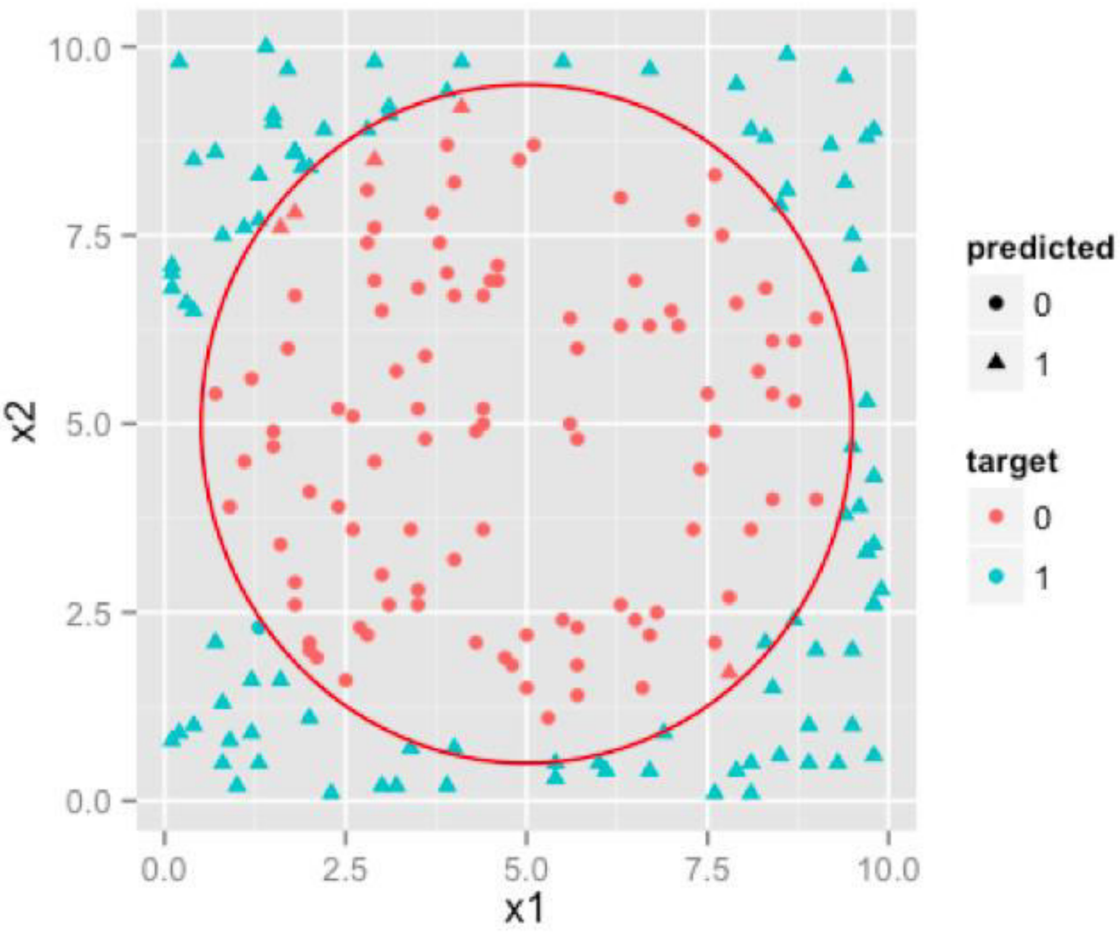


Figure 2.7

Note: The decision boundary will not be such a well rounded circle , but rather a very good approximation [a polygon] to it. We have used simple circle to avoid getting into hassle of drawing a tedious polygon in our software.

CHAPTER 3
STEPS FOR DETECTION

3.1. Enhancement

Image enhancement is the process of adjusting digital images so that the results are more suitable for display or further image analysis. For example, you can remove noise, sharpen, or brighten an image, making it easier to identify key features.

Adaptive histogram equalization (AHE) is a computer image processing technique used to improve contrast in images. It differs from ordinary histogram equalization in the respect that the adaptive method computes several histograms, each corresponding to a distinct section of the image, and uses them to redistribute the lightness values of the image. It is therefore suitable for improving the local contrast and enhancing the definitions of edges in each region of an image.

However, AHE has a tendency to over amplify noise in relatively homogeneous regions of an image. A variant of adaptive histogram equalization called contrast limited adaptive histogram equalization (CLAHE) prevents this by limiting the amplification.

Contrast Limited AHE (CLAHE) is a variant of adaptive histogram equalization in which the contrast amplification is limited, so as to reduce this problem of noise amplification.

In CLAHE, the contrast amplification in the vicinity of a given pixel value is given by the slope of the transformation function. This is proportional to the slope of the neighborhood cumulative distribution function (CDF) and therefore to the value of the histogram at that pixel value. CLAHE limits the amplification by clipping the histogram at a predefined value before computing the CDF. This limits the slope of the CDF and therefore of the transformation function. The value at which the histogram is clipped, the so-called clip limit, depends on the normalization of the histogram and thereby on the size of the neighborhood region. Common values limit the resulting amplification to between 3 and 4.

It is advantageous not to discard the part of the histogram that exceeds the clip limit but to redistribute it equally among all histogram bins.

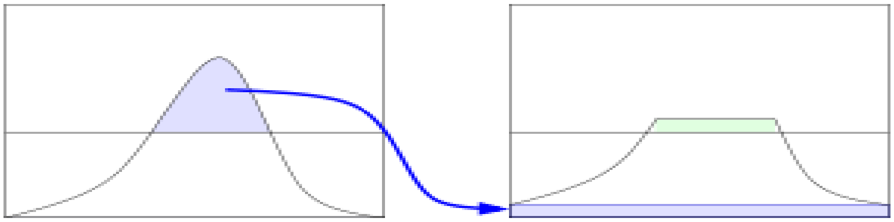


Figure 3.1

The redistribution will push some bins over the clip limit again (region shaded green in the figure), resulting in an effective clip limit that is larger than the prescribed limit and the exact value of which depends on the image. If this is undesirable, the redistribution procedure can be repeated recursively until the excess is negligible.

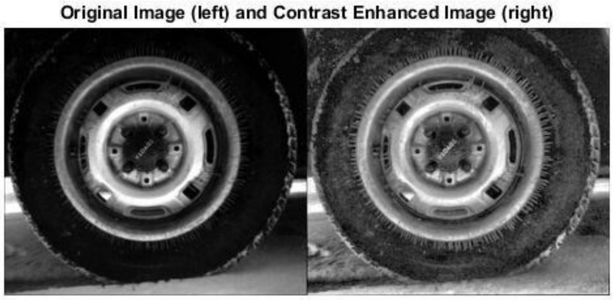


Figure 3.2

3.2. Segmentation

Image segmentation is one of the mostly used methods to classify the pixels of an image correctly in a decision oriented application. It divides an image into a number of discrete regions such that the pixels have high similarity in each region and high contrast between regions. It is a valuable tool in many field including health care, image processing, traffic image, pattern recognition etc. There are different techniques for image segmentation like threshold based, edge based, cluster based, and neural network based1. From the different technique one of the most efficient methods is the clustering method. Again there are different types of clustering: *K*-means clustering, Fuzzy *C*-means clustering, mountain clustering method and subtractive clustering method.

One of most used clustering algorithm is *k*-means clustering. It is simple and computationally faster than the hierarchical clustering. And it can also work for large number of variable. But it produces different cluster result for different number of number of cluster. So it is required to initialize the proper number of number of cluster, *K*2. Again it is required to initialize the *k* number of centroid. Different value of initial centroid would result different cluster.

So selection of proper initial centroid is also important task. Nowadays image segmentation becomes one of important tool in medical area where it is used to extract or region of interest from the background. So medical images are segmented using different technique and process outputs are used for the further analysis in medical. But medical images in their raw form are represented by the arrays of numbers in the computer3, with the number indicating the values of relevant physical quantities that show contrast between different types of body parts. Processing and analysis of medical images are useful in transforming raw images into a quantifiable symbolic form, in extracting meaningful qualitative information to aid diagnosis and in integrating complementary data from multiple imaging modalities. And one of the fundamental problems in medical analysis is the image segmentation which identifies the boundaries of objects such as organs or abnormal region in images. Results from the segmentation make it possible for shape analysis, detecting volume change, and making a precise radiation therapy treatment plant.

K-means algorithm is the one of the simplest clustering algorithm and there are many methods implemented so far with different method to initialize the centre. And many researchers are also trying to produce new methods which are more efficient than the existing methods, and shows better segmented result. Some of the existing recent works are discussed here.

Pallavi Purohit and Ritesh Joshi4 introduced a new efficient approach towards *K*-means clustering algorithm. They proposed a new method for generating the cluster center by reducing the mean square error of the final cluster without large increment in the execution time. It reduced the means square error without sacrificing the execution time. Many comparisons have been done and it can conclude that accuracy is more for dense dataset rather than sparse dataset.

Alan Jose, S. Ravi and M. Sambath5 proposed Brain Tumor Segmentation using *K*-means Clustering and Fuzzy *C*-means Algorithm and its area calculation. In the paper, they divide the process into three parts, pre-processing of the image, advanced *k*-means and fuzzy *c*-means and lastly the feature extraction. First pre-processing is implemented by using the filter where it improves the quality of the image. Then the proposed advance *K*-means algorithm is used, followed by Fuzzy *c*-means to cluster the image. Then the resulted segment image is used for the feature extraction for the region of interest. They used MRI image for the analysis and calculate the size of the extracted tumor region in the image. Madhu Yedla, Srinivasa Rao Pathakota, T. M. Srinivasa6 proposed Enhancing *K*-means clustering algorithm with improved initial center. A new method for finding the initial centroid is introduced and it provides an effective way of assigning the data points to suitable clusters with reduced time complexity. They proved their proposed algorithm has more accuracy with less computational time comparatively original *k*-means clustering algorithm. This algorithm does not require any additional input like threshold value. But this algorithm still initializes the number of cluster *k* and suggested determination of value of *k* as one of the future work.

Contrast Enhancement using Partial Contrast Stretching

Medical images which have been used for the analysis may have their own weakness such as blurred or low contrast. So a contrast enhancement technique such as Partial Spatial Starching (PCS) is used to improve the image quality and contrast of the image8. It is done by stretching and compression process. By applying this technique, the pixel range of lower threshold value and upper threshold value will be mapped to a new pixel range and stretched linearly to a wide range of pixels within new lower stretching value, and the remaining pixels will experience compression

Subtractive Clustering Algorithm

Subtractive clustering is a method to find the optimal data point to define a cluster centroid based on the density of surrounding data points9. This method is the extension of Mountain method, proposed by Chiu10.Mountain method is very simple and effective. It estimates the number and initial location of the cluster centers. It distribute the data space into gridding point and compute the potential for each data point base on its distance to the actual data point. So the grid point with many data point nearby will have high potential value. And so this grid point with highest potential value will be choose as first cluster centre. So after selecting the first cluster centre we will try to find the second cluster centre by calculating the highest potential value in the remaining grid points. As grid points near the first cluster center will reduce its potential value, the next cluster center will be grid with many data point nearby other than first cluster center grid point. So this procedure of acquiring new cluster center and reducing the potential of surrounding grid point repeat until potential of all grid points falls below a threshold value. So this method is one of the simplest and effective methods to find the cluster centers. But with increase in the dimension of data, its computation complexity grows exponentially. So, subtractive clustering algorithm solves the computational method associated with mountain method. It uses data points as the candidates for cluster centre and the computation of this method is proportional to the problem size.

K-Means Clustering Algorithm

Clustering is a method to divide a set of data into a specific number of groups. It's one of the popular method is *k*-means clustering. In *k*-means clustering, it partitions a collection of data into a *k* number group of data11, 12. It classifies a given set of data into *k* number of disjoint cluster. *K*-means algorithm consists of two separate phases.

In the first phase it calculates the *k* centroid and in the second phase it takes each point to the cluster which has nearest centroid from the respective data point. There are different methods to define the distance of the nearest centroid and one of the most used methods is Euclidean distance. Once the grouping is done it recalculate the new centroid of each cluster and based on that centroid, a new Euclidean distance is calculated between each center and each data point and assigns the points in the cluster which have minimum Euclidean distance. Each cluster in the partition is defined by its member objects and by its centroid. The centroid for each cluster is the point to which the sum of distances from all the objects in that cluster is minimized. So *K*-means is an iterative algorithm in which it minimizes the sum of distances from each object to its cluster centroid, over all clusters.

Let us consider an image with resolution of $x \times y$ and the image has to be cluster into k number of cluster. Let $p(x, y)$ be an input pixels to be cluster and ck be the cluster centers. The algorithm for *k*-means13 clustering is following as:

1. Initialize number of cluster *k* and centre.

2. For each pixel of an image, calculate the Euclidean distance *d*, between the center and each pixel of an image

3. Assign all the pixels to the nearest centre based on distance *d*.

4. After all pixels have been assigned, recalculate new position of the centre

5. Repeat the process until it satisfies the tolerance or error value.

6. Reshape the cluster pixels into image.

Although *k*-means has the great advantage of being easy to implement, it has some drawbacks. The quality of the final clustering results is depends on the arbitrary selection of initial centroid. So if the initial centroid is randomly chosen, it will get different result for different initial centers. So the initial center will be carefully chosen so that we get our desire segmentation. And also computational complexity is another term which we need to consider while designing the *K*-means clustering. It relies on the number of data elements, number of clusters and number of iteration.

Median Filter

Median filtering is used as a noise removal in order to obtain a noise free image. After segmentation is done, the segmented image may still present some unwanted regions or noise. So to make the image a good and better quality, the median filter is applied to the segmented image. We can use different neighborhood of $n \times n$. But generally neighborhood of $n = 7$ is used because large neighborhoods produce more severe smoothing.

Proposed Algorithm

The proposed algorithm consists of partial contrast stretching, subtractive clustering, *k*-means clustering and median filter. Mostly the medical images which are used for segmentation have low contrast. So contrast stretching is used to improve the quality of the image. After improving the quality of image, subtractive clustering algorithm is used to generate the centers, based on the potential value of the image. Number of centre is generated based on number of cluster *k*. This centre is used as initial centre in *k*-means algorithm. Using the *k*-means algorithm, the image is segmented into *k* number of cluster. After the segmentation of image, the image can still contain some unwanted region or noise. These noises are removed by using the median filter. The proposed algorithm is followed

1. Load the image to be segmented.

2. Apply partial contrast stretching. Initialize number of cluster, *k*.

3. To calculate the potential for every pixel value of the image.

4. Find maximum potential in step 3 and set that point be first center cluster and its corresponding potential as maximum potential.

5. To update the potential value of other remaining pixels based on the first cluster center..

6. Again find the maximum potential in the step 4 and let that point be second point.

7. Continue the process until it finds the *k* number of cluster.

8. Used *k* centre as initial centre in the *k*-means clustering algorithm.

9. Find the Euclidean distance of each centroid from every pixel of the image 10. Assign the pixel with minimum distance with respect to centroid to its respective cluster of the centroid.

11. Recalculate the new center location

12. Repeat the steps 10–12, until it satisfies the tolerance or error value.

13. Reshape the cluster into image.

14. Median filter is applied to the segmented image to remove any unwanted noise or region.

3.3. Feature Extraction

3.3.1 Introduction

Automate Data Acquisition

MathWorks Consultants help you automate data access from disparate sources including spreadsheets, CSV files, ODBC and JDBC databases, or data feed services such as ISO-NE, CAISO, MISO, or ENTSO-E. As correct and clean data is key to accurate predictive models, we show you how to establish data preprocessing integrity checks, data conversions, and missing data handling.

Develop Predictive Models

We teach you best practices for building your customized predictive models using nonlinear regression, nonparametric, and neural networks techniques. We coach you on identifying most relevant independent variables and calibrating your models with historical predictors such as weather, seasonality, day of week, time of day, and power price.

Scrutinize Models

We show you how your models can be improved and quickly updated to incorporate additional predictors. To allow you to scrutinize your models and gain confidence in their results, MathWorks Consultants demonstrate methods for easily viewing inputs, outputs, and intermediate variables, and stepping through calculations to achieve full understanding of your model's behavior.

Deploy Algorithms to Enterprise Systems and Web

We work with you to integrate and deploy your load forecasting algorithms across your enterprise systems. We help you automatically generate reports to share results, develop data visualizations, and enable use of your models by a wide audience. MathWorks Consultants guide you on deployment strategies from a standalone application or spreadsheet add-in to generating .NET or Java® software components that can be integrated into web and enterprise applications.

3.3.2 Investigate Features in the Scatter Plot

In Classification Learner, try to identify predictors that separate classes well by plotting different pairs of predictors on the scatter plot. The plot can help you investigate features to include or exclude. You can visualize training data and misclassified points on the scatter plot.

Before you train a classifier, the scatter plot shows the data. If you have trained a classifier, the scatter plot shows model prediction results. Switch to plotting only the data by selecting Data in the Plot controls.

- Choose features to plot using the X and Y lists under Predictors.
- Look for predictors that separate classes well.
- Show or hide specific classes using the check boxes under Show.

- Change the stacking order of the plotted classes by selecting a class under Classes and then clicking Move to Front.
- Investigate finer details by zooming in and out and panning across the plot. To enable zooming and panning, hover the mouse over the scatter plot and click one of the buttons that appear near the top-right corner of the plot.
- If you identify predictors that are not useful for separating out classes, then try using **Feature Selection** to remove them and train classifiers including only the most useful predictors.

After you train a classifier, the scatter plot shows model prediction results. You can show or hide correct or incorrect results and visualize the results by class.

3.3.3 Select Features to Include

In Classification Learner, you can specify different features (or predictors) to include in the model. See if you can improve models by removing features with low predictive power. If data collection is expensive or difficult, you might prefer a model that performs satisfactorily without some predictors.

1. On the **Classification Learner** tab, in the **Features** section, click **Feature Selection**.
2. In the Feature Selection tearaway window, clear the check boxes for the predictors you want to exclude.
3. Click **Train** to train a new model using the new predictor options.
4. Observe the new model in the History list. The Current model pane displays how many predictors are excluded.
5. To check which predictors are included in a trained model, click the model in the History list and observe the check boxes in the Feature Selection dialog box.
6. You can try to improve the model by including different features in the model.

3.3.4 Transform Features with PCA in Classification Learner

Use principal component analysis (PCA) to reduce the dimensionality of the predictor space. Reducing the dimensionality can create classification models in Classification Learner that help prevent overfitting. PCA linearly transforms predictors in order to remove redundant dimensions, and generates a new set of variables called principal components.

1. On the Classification Learner tab, in the Features section, select PCA.
2. In the Advanced PCA Options tearaway window, select the Enable PCA check box.

   You can close the PCA tearaway window, or move it. Your choices in the tearaway remain.
3. When you next click Train, the pca function transforms your selected features before training the classifier.
4. By default, PCA keeps only the components that explain 95% of the variance. In the PCA tearaway window, you can change the percentage of variance to explain in the Explained variance box. A higher value risks overfitting, while a lower value risks removing useful dimensions.
5. If you want to manually limit the number of PCA components, in the Component reduction criterion list, select Specify number of components. Edit the number in the Number of numeric components box. The number of components cannot be larger than the number of numeric predictors. PCA is not applied to categorical predictors.

3.3.5 Investigate Features in the Parallel Coordinates Plot

To investigate features to include or exclude, use the parallel coordinates plot. You can visualize high dimensional data on a single plot to see 2D patterns. This plot can help you understand relationships between features and identify useful predictors for separating classes. You can visualize training data and misclassified points on the parallel coordinates plot. When you plot classifier results, misclassified points show dashed lines.

1. On the Classification Learner tab, in the Plots section, select Parallel Coordinates Plot.
2. On the plot, you can drag the bars to reorder the predictors. Changing the order can help you identify predictors that separate classes well.
3. To specify which predictors to plot, use the **Predictors** check boxes. It can be helpful to plot a few predictors at a time. If your data has many predictors, the plot shows the first 10 by default.
4. If the predictors have very different scales, scale the data to make it easier to visualize. Try different options in the **Scaling** list:

- Normalization plots all predictors on the same range from 0 to 1.
- Standardization plots the mean of each predictor at zero and scales the predictors by their standard deviations.

5. If you identify predictors that are not useful for separating out classes, use **Feature Selection** to remove them and train classifiers including only the most useful predictors.

3.4. Classification

3.4.1 Introduction

In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

When data are not labeled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups.

3.4.2 Motivation

Classifying data is a common task in machine learning. Suppose some given data points each belong to one of two classes, and the goal is to decide which class a new Data point will be in. In the case of support vector machines, a data point is viewed as a p-dimensional vector (a list of p numbers), and we want to know whether we can separate such points with a (p-1)-dimensional hyperplane. This is called a linear classifier. There are many hyperplanes that might classify the data. One reasonable choice as the best hyperplane is the one that represents the largest separation, or margin, between the two classes. So we choose the hyperplane so that the distance from it to the nearest data point on each side is maximized. If such a hyperplane exists, it is known as the maximum-margin hyperplane and the linear classifier it defines is known as a maximum margin classifier; or equivalently, the perceptron of optimal stability.

3.4.3 Definition

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. A SVM performs classification by finding the hyperplane that maximizes the margin between the two classes. The vectors (cases) that define the hyperplane are the support vectors.

3.4.4 Maximal-Margin Classifier

The Maximal-Margin Classifier is a hypothetical classifier that best explains how SVM works in practice.

The numeric input variables (x) in your data (the columns) form an n-dimensional space. For example, if you had two input variables, this would form a two-dimensional space.

A hyperplane is a line that splits the input variable space. In SVM, a hyperplane is selected to best separate the points in the input variable space by their class, either class 0 or class 1. In two-dimensions you can visualize this as a line and let’s assume that all of our input points can be completely separated by this line. For example:

$$B0 + (B1 * X1) + (B2 * X2) = 0$$

Where the coefficients (B1 and B2) that determine the slope of the line and the intercept (B0) are found by the learning algorithm, and X1 and X2 are the two input variables.

You can make classifications using this line. By plugging in input values into the line equation, you can calculate whether a new point is above or below the line.

- Above the line, the equation returns a value greater than 0 and the point belongs to the first class (class 0).
- Below the line, the equation returns a value less than 0 and the point belongs to the second class (class 1).
- A value close to the line returns a value close to zero and the point may be difficult to classify.
- If the magnitude of the value is large, the model may have more confidence in the prediction.

The distance between the line and the closest data points is referred to as the margin. The best or optimal line that can separate the two classes is the line that as the largest margin. This is called the Maximal-Margin hyperplane.

The margin is calculated as the perpendicular distance from the line to only the closest points. Only these points are relevant in defining the line and in the construction of the classifier. These points are called the support vectors. They support or define the hyperplane.

The hyperplane is learned from training data using an optimization procedure that maximizes the margin.

3.4.5 Soft Margin Classifier

In practice, real data is messy and cannot be separated perfectly with a hyperplane.

The constraint of maximizing the margin of the line that separates the classes must be relaxed. This is often called the soft margin classifier. This change allows some points in the training data to violate the separating line.

An additional set of coefficients are introduced that give the margin wiggle room in each dimension. These coefficients are sometimes called slack variables. This increases the complexity of the model as there are more parameters for the model to fit to the data to provide this complexity.

A tuning parameter is introduced called simply C that defines the magnitude of the wiggle allowed across all dimensions. The C parameters defines the amount of violation of the margin allowed. A C=0 is no violation and we are back to the inflexible Maximal-Margin Classifier described above. The larger the value of C the more violations of the hyperplane are permitted.

During the learning of the hyperplane from data, all training instances that lie within the distance of the margin will affect the placement of the hyperplane and are referred to as support vectors. And as C affects the number of instances that are allowed to fall within the margin, C influences the number of support vectors used by the model.

- The smaller the value of C, the more sensitive the algorithm is to the training data (higher variance and lower bias).
- The larger the value of C, the less sensitive the algorithm is to the training data (lower variance and higher bias).

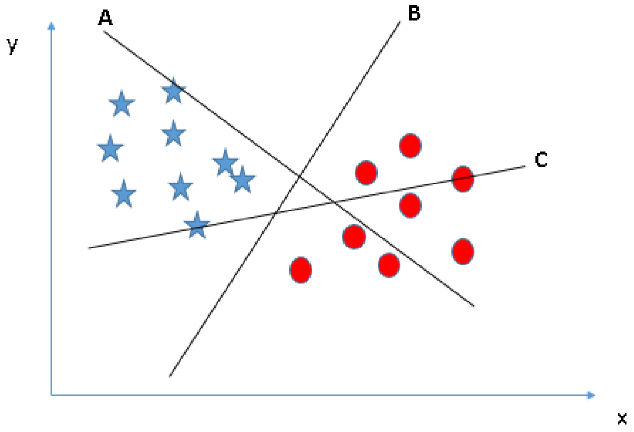


Figure 3.3

CHAPTER 4
DESIGN

4.1. Technology Architecture

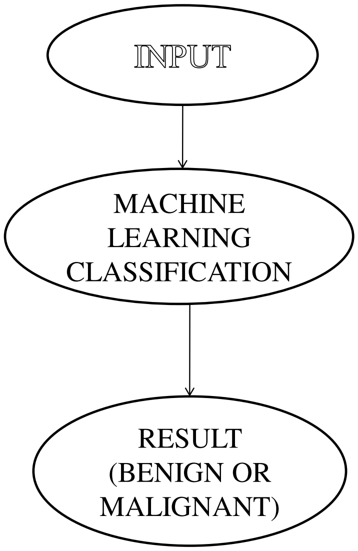


Figure 4.1

4.2. UML Diagrams

Class Diagram:

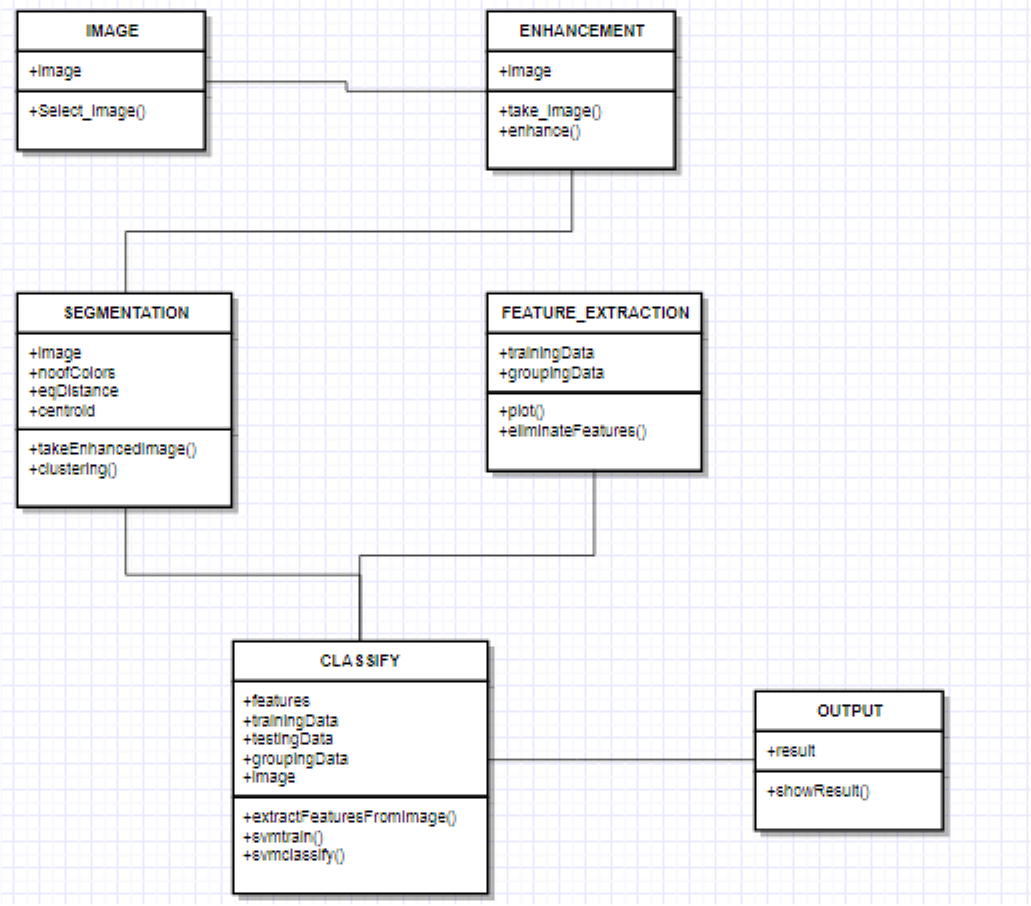


IMAGE
+Image
+Select_Image()
ENHANCEMENT
+Image
+take_Image()
+enhance()
SEGMENTATION
+Image
+noofColors
+eqDistance
+centroid
+takeEnhancedImage()
+clustering()
FEATURE_EXTRACTION
+trainingData
+groupingData
+plot()
+eliminateFeatures()
CLASSIFY
+features
+trainingData
+testingData
+groupingData
+Image
+extractFeaturesFromImage()
+svmtrain()
+svmclassify()
OUTPUT
+result
+showResult()

Figure 4.2

CHAPTER 5
FUNCTIONS USED

5.1. uigetfile

Interactively retrieve a filename.

Syntax:

- uigetfile
- uigetfile('FilterSpec')

Description:

uigetfile displays a dialog box used to retrieve a file. The dialog box lists the files and directories in the current directory.

uigetfile('FilterSpec') displays a dialog box that lists files in the current directory. FilterSpec determines the initial display of files and can be a full filename or include the * wildcard. For example, '*.m'lists all the MATLAB M-files. If FilterSpec is a cell array, the first column is use as the list of extensions, and the second column is used as the list of descriptions.

uigetfile('FilterSpec','DialogTitle') displays a dialog box that has the title DialogTitle.

5.2. imread

Read image from graphics file

Syntax:

- A = imread(filename,fmt)
- [X,map] = imread(filename,fmt)
- [...] = imread(filename)
- [...] = imread(URL,...)

Description:

The imread function also supports several other format-specific syntaxes. A = imread(filename,fmt) reads a greyscale or color image from the file specified by the string filename, where the string fmt specifies the format of the file. If the file is not in the current directory or in a directory in the MATLAB path, specify the full pathname of the location on your system.. If imread cannot find a file named filename, it looks for a file named filename.fmt.

imread returns the image data in the array A. If the file contains a grayscale image, A is a two-dimensional (M-by-N) array. If the file contains a color image, A is a three-dimensional (M-by-N-by-3) array. The class of the returned array depends on the data type used by the file format.. For most file formats, the color image data returned uses the RGB color space [X,map] = imread(filename,fmt) reads the indexed image in filename into X and its associated colormap into map. The colormap values are rescaled to the range [0,1].

[...] = imread(filename) attempts to infer the format of the file from its content.

[...] = imread(URL,...) reads the image from an Internet URL. The URL must include the protocol type (e.g., http://).

5.3. Imresize

Resize image

Syntax:

- B = imresize(A,scale)
- B = imresize(A,[numrows numcols])

Description:

B = imresize(A,scale) returns image B that is scale times the size of A. The input image A can be a grayscale, RGB, or binary image. If A has more than two dimensions, imresize only resizes the first two dimensions. If scale is in the range [0, 1], B is smaller than A. If scale is greater than 1, B is larger than A. By default, imresize uses bicubic interpolation.

B = imresize(A,[numrows numcols]) returns image B that has the number of rows and columns specified by the two-element vector [numrows numcols].

5.4. makecform

Create color transformation structure

The makecform function supports conversions between members of the family of device-independent color spaces defined by the Commission Internationale de l'Éclairage(International Commission on Illumination, or CIE). makecform also supports conversions to and from the sRGB and CMYK color spaces. To perform a color space transformation, pass the color transformation structure created by makecform as an argument to the applycform function.

Syntax:

C = makecform(type)

C = makecform(type,'WhitePoint',WP)

Description:

C = makecform(type) creates a color transformation structure C that defines the color space conversion specified by type.

C = makecform(type,'WhitePoint',WP) specifies the value of the reference white point, WP, for 'xyz2lab' or 'lab2xyz' conversions.

5.5. imadjust

Adjust image intensity values or colormap

Syntax:

- J = imadjust(I)
- J = imadjust(I,[low_in high_in],[low_out high_out])
- RGB2 = imadjust(RGB,___)

Description:

J = imadjust(I) maps the intensity values in grayscale image I to new values in J. By default, imadjust saturates the bottom 1% and the top 1% of all pixel values. This operation increases the contrast of the output image J. This syntax is equivalent to imadjust(I,stretchlim(I)).

J = imadjust(I,[low_in high_in],[low_out high_out]) maps intensity values in I to new values in J such that values between low_in and high_in map to values between low_out and high_out. You can omit the [low_out high_out] argument, in which case, imadjust uses the default [0 1].

RGB2 = imadjust(RGB,___) performs the adjustment on each plane (red, green, and blue) of the RGB intensity image RGB. You can apply the same mapping to the red, green, and blue components of the image or specify unique mappings for each color component.

5.6. applycform

Apply device-independent color space transformation

Syntax

- B = applycform(A,C)

Description:

B = applycform(A,C) converts the color values in A to the color space specified in the color transformation structure C

Convert sRGB to L*a*b* Color Space using Applycform

Read color image that uses the sRGB color space into the workspace.

rgb = imread('peppers.png');

Create a color transformation structure that defines an sRGB to L*a*b* conversion.

C = makecform('srgb2lab');

Perform the transformation with applycform.

lab = applycform(rgb,C);

5.7. reshape

Reshape array

Syntax

- B = reshape(A,m,n)
- B = reshape(A,siz)

Description

B = reshape(A,m,n) returns the m-by-n matrix B whose elements are taken column-wise from A. An error results if A does not have m*n elements.

B = reshape(A,siz) returns an N-D array with the same elements as A, but reshaped to siz, a vector representing the dimensions of the reshaped array. The quantity prod(siz) must be the same as prod(size(A)).

5.8. repmat

Replicate and tile an array

Syntax:

- B = repmat(A,m,n)
- B = repmat(A,[m n])

Description:

B = repmat(A,m,n) creates a large matrix B consisting of an m-by-n tiling of copies of A. The statement repmat(A,n) creates an n-by-n tiling.

B = repmat(A,[m n]) accomplishes the same result as repmat(A,m,n)..

5.9. Imshow

Display image

Syntax:

imshow(I)

imshow(I,[low high])

imshow(RGB)

Description:

imshow(I) displays the grayscale image I in a figure. imshow optimizes figure, axes, and image object properties for image display.

imshow(I,[low high]) displays the grayscale image I, specifying the display range as a two-element vector, [low high]. For more information, see theDisplayRange parameter.

imshow(RGB) displays the truecolor image RGB in a figure.

imshow(BW) displays the binary image BW in a figure. For binary images, imshow displays pixels with the value 0 (zero) as black and 1 as white.

5.10. Imbinarize

Binarize 2-D grayscale image or 3-D volume by thresholding

Syntax:

BW = imbinarize(I)

BW = imbinarize(I,method)

Description:

BW = imbinarize(I) creates a binary image from image I by replacing all values above a globally determined threshold with 1s and setting all other values to 0s. By default, imbinarize uses Otsu's method, which chooses the threshold value to minimize the intraclass variance of the thresholded black and white pixels [1]. imbinarize uses a 256-bin image histogram to compute Otsu's threshold. To use a different histogram, see otsuthresh. BW is the output binary image.

BW = imbinarize(I,method) creates a binary image from image I using the thresholding method specified by method: 'global' or 'adaptive'.

5.11. kmeans(,)

idx = kmeans(X,k) performs k-means clustering to partition the observations of the n-by-p data matrix X into k clusters, and returns an n-by-1 vector (idx) containing cluster indices of each observation. Rows of X correspond to points and columns correspond to variables.

By default, kmeans uses the squared Euclidean distance metric and the k-means++ algorithm for cluster center initialization.

Example :

idx = kmeans(X,k,Name,Value)returns the cluster indices with additional options specified by one or moreName,Value pair arguments.

For example, specify the cosine distance, the number of times to repeat the clustering using new initial values, or to use parallel computing.

5.12. imdistline

Distance tool

Description:

An imdistline object is a type of imline that encapsulates a Distance tool, which consists of an interactive line over an image, paired with a text label that displays the distance between the line endpoints.

You can adjust the size and position of the line by using the mouse. The line also has a context menu that controls aspects of its appearance and behavior.

Syntax:

- h = imdistline(hparent)

Description:

h = imdistline creates a Distance tool on the current axes. The function returns h, a handle to an imdistline object.

h = imdistline(hparent) creates a draggable Distance tool on the object specified by hparent.

imfindcircles

5.13. Svmtrain

Train support vector machine classifier

Syntax:

SVMStruct=svmtrain(Training,Group)

SVMStruct = svmtrain(Training,Group,Name,Value)

Description:

SVMStruct = svmtrain(Training,Group) returns a structure, SVMStruct, containing information about the trained support vector machine (SVM) classifier.

SVMStruct = svmtrain(Training,Group,Name,Value) returns a structure with additional options specified by one or more Name,Value pair arguments.

Name-Value Pair Arguments:

Specify optional comma-separated pairs of Name,Value arguments. Name is the argument name and Value is the corresponding value. Name must appear inside single quotes (' '). You can specify several name and value pair arguments in any order as Name1,Value1,...,NameN,ValueN.

5.14. Svmclassify

Classify using support vector machine (SVM)

Syntax:

- Group=svmclassify(SVMStruct,Sample)
- Group = svmclassify(SVMStruct,Sample,'Showplot',true)

Description:

Group = svmclassify(SVMStruct,Sample) classifies each row of the data in Sample, a matrix of data, using the information in a support vector machine classifier structure SVMStruct, created using the svmtrain function. Like the training data used to create SVMStruct, Sample is a matrix where each row corresponds to an observation or replicate, and each column corresponds to a feature or variable. Therefore, Sample must have the same number of columns as the training data. This is because the number of columns defines the number of features. Group indicates the group to which each row of Sample has been assigned.

Group = svmclassify(SVMStruct,Sample,'Showplot',true) plots the Sample data in the figure created using the Showplot property with the svmtrain function. This plot appears only when the data is two-dimensional.

CHAPTER 6

CODE

6.1 INPUT THE IMAGE TO BE CLASSIFIED:

```
function[]=cancer();
clc
closeall
clearall
[filename, pathname] = uigetfile({'*.*';'*.bmp';'*.jpg';'*.gif'}, 'Pick a Pathology slide’);
he = imread([pathname,filename]);
```

6.2 IMAGE ENHANCEMENT:

```
he = imresize(he,[256,256]);
he = imadjust(he,stretchlim(he));
cform = makecform('srgb2lab');
lab_he = applycform(he,cform);
ab = double(lab_he(:,:,2:3));
nrows = size(ab,1);
ncols = size(ab,2);
ab = reshape(ab,nrows*ncols,2);
```

6.3 SEGMENTATION:

```
nColors = 3;
% repeat the clustering 3 times to avoid local minima
[cluster_idx, cluster_center] = kmeans(ab,nColors,'distance','sqEuclidean', ...
'Replicates',3);
pixel_labels = reshape(cluster_idx,nrows,ncols);
```

```
segmented_images = cell(1,3);
rgb_label = repmat(pixel_labels,[1 1 3]);
for k = 1:nColors
        color = he;
        color(rgb_label ~= k) = 0;
        segmented_images{k} = color;
end
imshow(segmented_images{1});title('Cluster      1');imshow(segmented_images{2});title('Cluster
2');
imshow(segmented_images{3});title('Cluster 3');
```

6.4 FEATURE EXTRACTION:

```
viscircles(centers,radii);
radius=max(radii);
aream=radii*3.14;
aream=aream.*radii;
area1=max(aream);
perim=radii*2*3.14;
peri=max(perim);
Variance = mean2(var(double(blue_nuclei)));
smoothness=Variance/10000;
texture = std2(blue_nuclei);
feat_disease = [radius,texture,peri,area1,smoothness];
```

6.5 CLASSIFY:

```
function [itrfin] = multisvm( T,C,test)
%Inputs: T=Training Matrix, C=Group, test=Testing matrix
%Outputs: itrfin=Resultant class
svmStruct = svmtrain(T,newClass,'kernel_function','rbf');
classes = svmclassify(svmStruct,tst);
itrfin(tempind,:)=val;
```

```
end
```

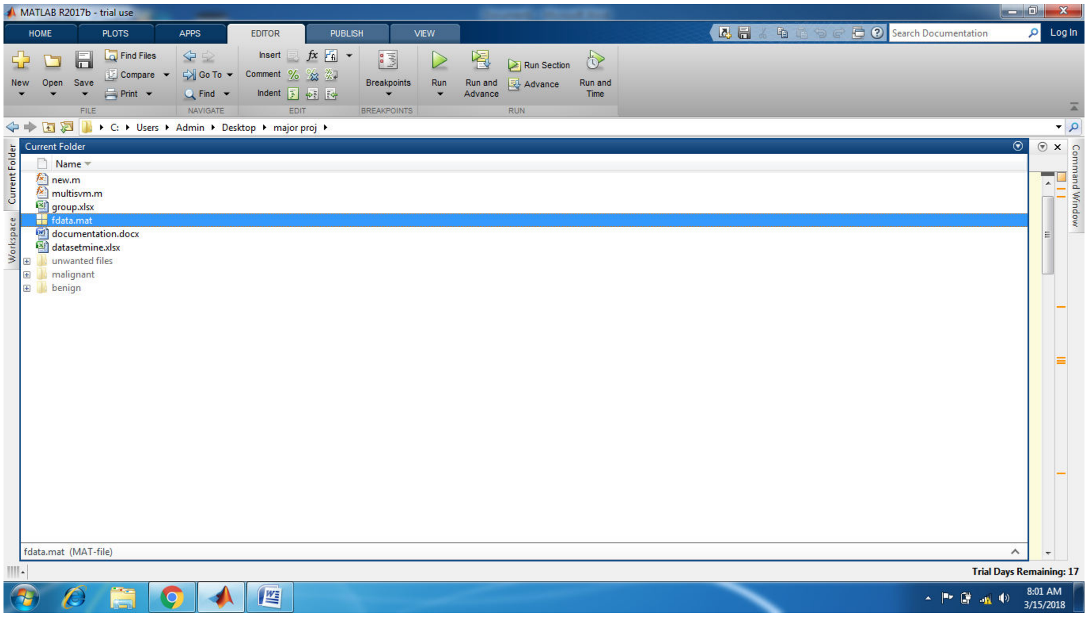
6.6 DISPLAY RESULTS:

```
result = multisvm(data,grp,test);
if result == 0
        helpdlg(' Benign ');
        disp(' Benign ');
elseif result == 1
        helpdlg(' Malignant ');
        disp('Malignant');
```

CHAPTER 7

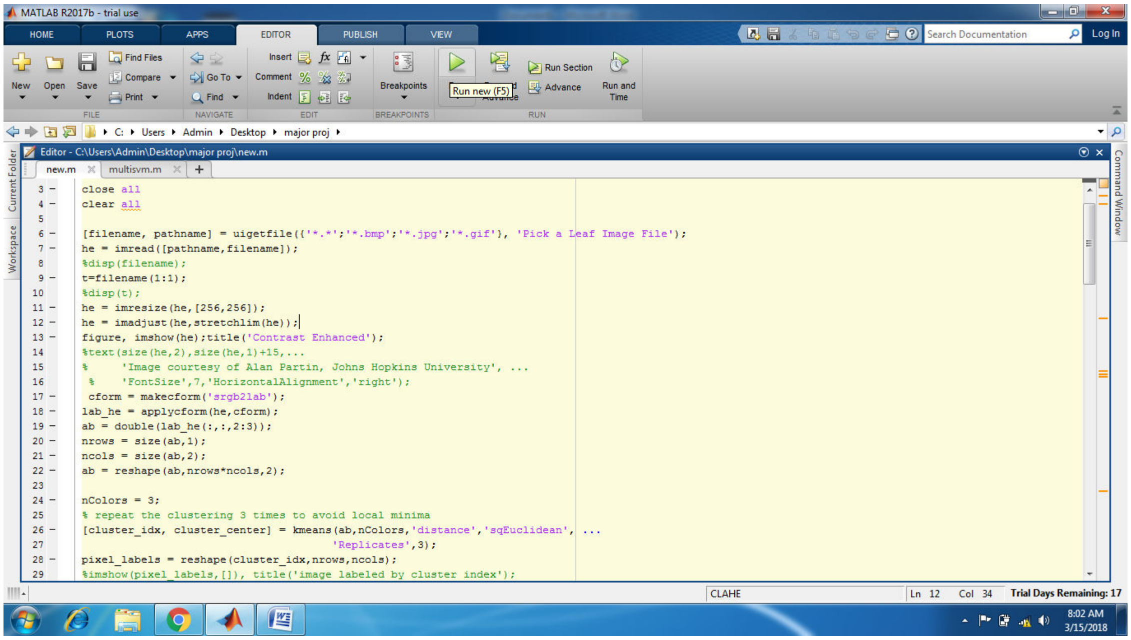
RESULTS AND SCREEN SHOTS

STEP-1 : Load the dataset fdata.mat



Figures 7.1

STEP-2 : Open new.m and click on run button



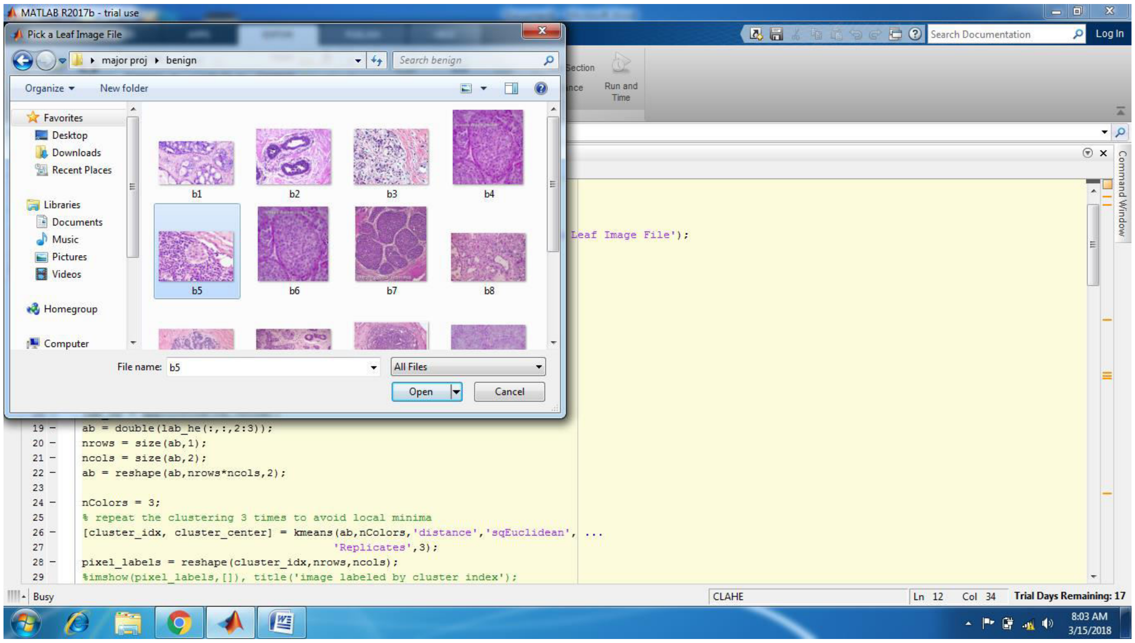
```
close all
clear all

[filename, pathname] = uigetfile({'*.*';'*.bmp';'*.jpg';'*.gif'}, 'Pick a Leaf Image File');
he = imread([pathname,filename]);
%disp(filename);
t=filename(1:1);
%disp(t);
he = imresize(he,[256,256]);
he = imadjust(he,stretchlim(he));
figure, imshow(he);title('Contrast Enhanced');
%text(size(he,2),size(he,1)+15,...
%      'Image courtesy of Alan Partin, Johns Hopkins University', ...
 %     'FontSize',7,'HorizontalAlignment','right');
 cform = makecform('srgb2lab');
lab_he = applycform(he,cform);
ab = double(lab_he(:,:,2:3));
nrows = size(ab,1);
ncols = size(ab,2);
ab = reshape(ab,nrows*ncols,2);

nColors = 3;
% repeat the clustering 3 times to avoid local minima
[cluster_idx, cluster_center] = kmeans(ab,nColors,'distance','sqEuclidean', ...
                                      'Replicates',3);
pixel_labels = reshape(cluster_idx,nrows,ncols);
%imshow(pixel_labels,[]), title('image labeled by cluster index');
```

Figures 7.2

STEP-3 : Select an image from a dataset



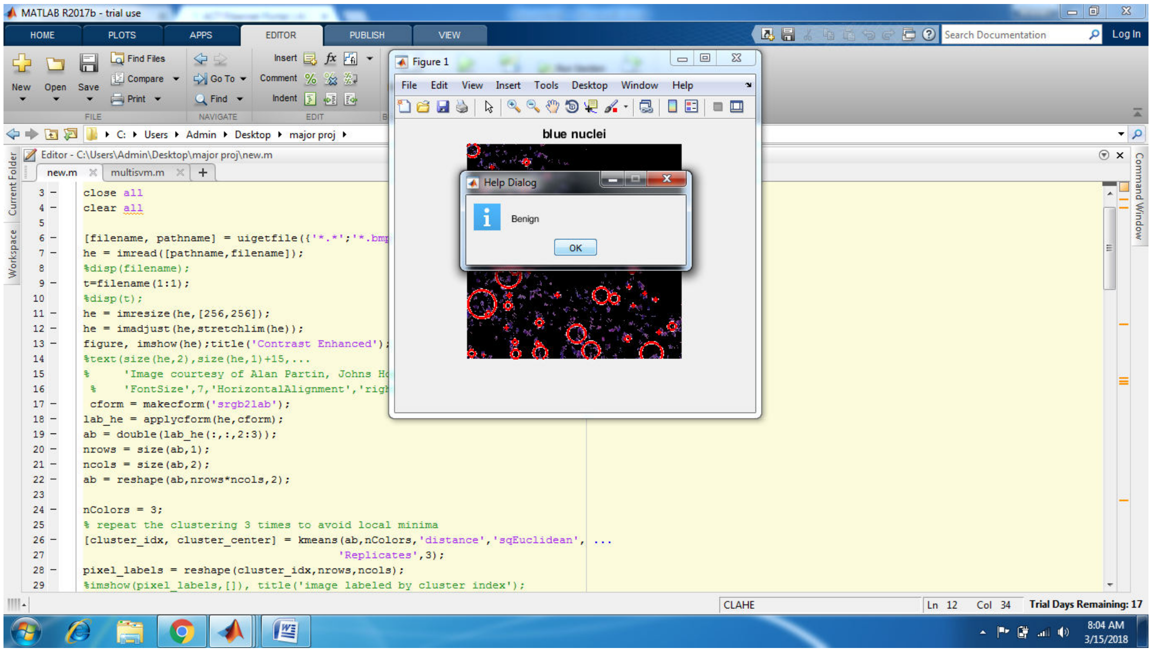
File name: b5 — All Files

```
ab = double(lab_he(:,:,2:3));
nrows = size(ab,1);
ncols = size(ab,2);
ab = reshape(ab,nrows*ncols,2);

nColors = 3;
% repeat the clustering 3 times to avoid local minima
[cluster_idx, cluster_center] = kmeans(ab,nColors,'distance','sqEuclidean', ...
                                      'Replicates',3);
pixel_labels = reshape(cluster_idx,nrows,ncols);
%imshow(pixel_labels,[]), title('image labeled by cluster index');
```

Figure 7.3

STEP-4 : It gives you the result whether the pathology slide is benign or malignant.



Benign

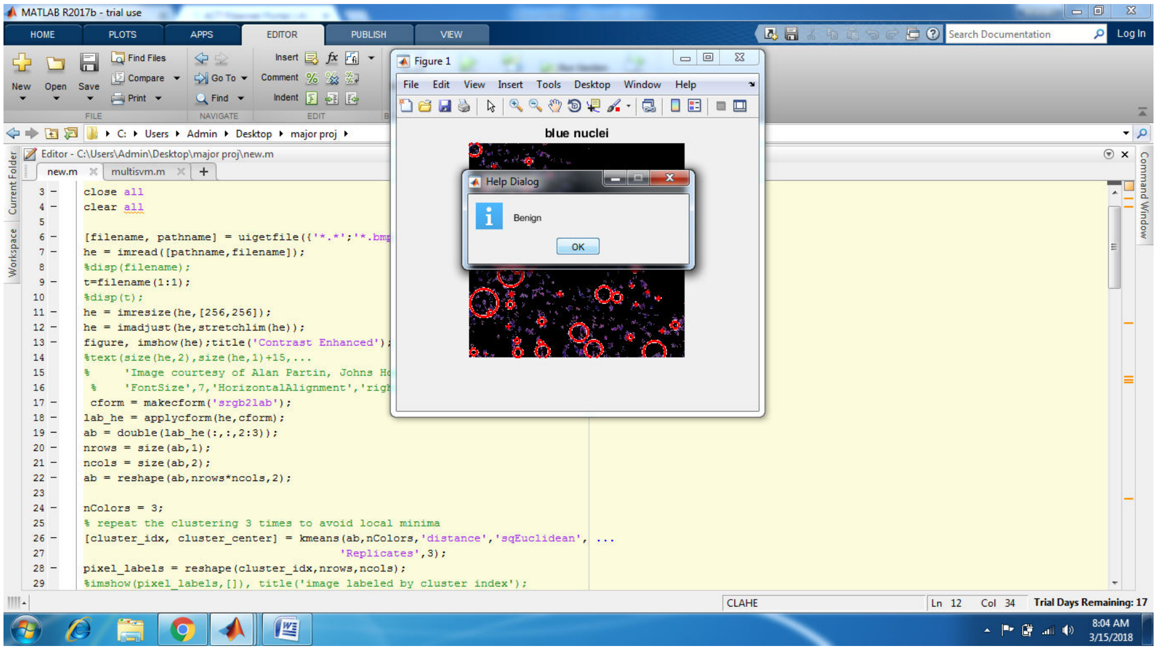
```
close all
clear all

[filename, pathname] = uigetfile({'*.*';'*.bmp
he = imread([pathname,filename]);
%disp(filename);
t=filename(1:1);
%disp(t);
he = imresize(he,[256,256]);
he = imadjust(he,stretchlim(he));
figure, imshow(he);title('Contrast Enhanced');
%text(size(he,2),size(he,1)+15,...
%     'Image courtesy of Alan Partin, Johns Ho
 %    'FontSize',7,'HorizontalAlignment','righ
 cform = makecform('srgb2lab');
lab_he = applycform(he,cform);
ab = double(lab_he(:,:,2:3));
nrows = size(ab,1);
ncols = size(ab,2);
ab = reshape(ab,nrows*ncols,2);

nColors = 3;
% repeat the clustering 3 times to avoid local minima
[cluster_idx, cluster_center] = kmeans(ab,nColors,'distance','sqEuclidean', ...
                                      'Replicates',3);
pixel_labels = reshape(cluster_idx,nrows,ncols);
%imshow(pixel_labels,[]), title('image labeled by cluster index');
```

Figure 7.4

Step 5: Result is displayed



Benign

```
close all
clear all

[filename, pathname] = uigetfile({'*.*';'*.bmp
he = imread([pathname,filename]);
%disp(filename);
t=filename(1:1);
%disp(t);
he = imresize(he,[256,256]);
he = imadjust(he,stretchlim(he));
figure, imshow(he);title('Contrast Enhanced');
%text(size(he,2),size(he,1)+15,...
%     'Image courtesy of Alan Partin, Johns Ho
 %    'FontSize',7,'HorizontalAlignment','righ
 cform = makecform('srgb2lab');
lab_he = applycform(he,cform);
ab = double(lab_he(:,:,2:3));
nrows = size(ab,1);
ncols = size(ab,2);
ab = reshape(ab,nrows*ncols,2);

nColors = 3;
% repeat the clustering 3 times to avoid local minima
[cluster_idx, cluster_center] = kmeans(ab,nColors,'distance','sqEuclidean', ...
                                      'Replicates',3);
pixel_labels = reshape(cluster_idx,nrows,ncols);
%imshow(pixel_labels,[]), title('image labeled by cluster index');
```

Figure 7.5

CHAPTER 8
CONCLUSION

The early diagnosis and prognosis of a cancer type have become a necessity in cancer research, as it can facilitate the subsequent clinical management of patients .The importance of classifying cancer patients into high or low risk groups has led many research teams , from the biomedical and the bioinformatics field, to study the application of machine learning (ML)methods. Cancer Informatics is an automated detection algorithm that can naturally complement pathologists' workflow. Support Vector Machines (SVMs) are used to classify the pathology slides. We are also trying to test the efficiency of different classification algorithms. We shall also try to implement the same by using pathology slides as training data. But, as of now, the project is fit to be implemented in real time and work efficiently with the doctor.

REFRENCES

- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4996960/
- https://www.theguardian.com/society/2014/sep/22/cancer-late-diagnosis-half-patients
- http://www.nrls.npsa.nhs.uk/EasySiteWeb/getresource.axd?AssetID=69895
- https://shiring.github.io/machine_learning/2017/01/15/rfe_ga_post
- https://in.mathworks.com/help/images/examples/detect-and-measure-circular-objects-in-an-image.html
- https://www.omicsonline.org/optimizing-number-of-inputs-to-classify-breast-cancer-using-artificial-neural-network-jcsb.1000247.php?aid=1255
- https://www.edvancer.in/logistic-regression-vs-decision-trees-vs-svm-part1/
- https://www.deccanchronicle.com/technology/in-other-news/040317/googles-ai-is-now-detecting-cancer-with-deep-learning.html
- https://www.sciencedirect.com/science/article/pii/S2001037017300867
- http://journals.sagepub.com/doi/pdf/10.1177/117693510600200030
- https://journals.lww.com/oncology-times/Fulltext/2005/11250/Study__Pathology_Errors_Can_Have_Serious_Effect_on.17.aspx
- http://www.svms.org/anns.html
- https://news.microsoft.com/stories/computingcancer/