HYBRID K-NEAREST NEIGHBOUR AND DISCRIMINANT ANALYSIS FOR PREDICTING MEDICAL DIAGNOSIS IN DECISION SUPPORT SYSTEM

A Thesis submitted for the Degree of Master of Science

In the School of Engineering

By

Rahul Shankar Iyer

DEPARTMENT OF MATHEMATICS

AMRITA SCHOOL OF ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE – 641112(INDIA)

APRIL, 2019

AMRITA SCHOOL OF ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE-641 112

BONAFIDE CERTIFICATE

This is to certify that the thesis entitled "**HYBRID K-NEAREST NEIGHBOUR AND DISCRIMINANT ANALYSIS FOR PREDICTING MEDICAL DIAGNOSIS IN DECISION SUPPORT SYSTEM**" submitted by **Rahul Shankar Iyer** (Register Number-CB.SC.P2MAT17004) for the award of the Degree of Master of Science in the School of Engineering is a bonafide record of the work carried out by him under my guidance and supervision at Amrita School of Engineering, Coimbatore.

Signature of the HOD	**Signature of the Project Coordinator**
Dr. K. Somasundaram	**Dr. R. Radha Iyer**
Signature of the Project Guide	**Signature of the External Examiner**
Dr. O. S. Deepa Gopakumar	_____

AMRITA SCHOOL OF ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE-641 112

DEPARTMENT OF MATHEMATICS

DECLARATION

I, Rahul Shankar Iyer(Register Number-CB.SC.P2MAT17004),hereby declare that this thesis entitled **"Hybrid K-Nearest Neighbour and Discriminant Analysis for Predicting Medical Diagnosis in Decision Support System",** is the record of the original work done by me under the guidance of **Dr. R. Subash Moorthy**, Assistant Professor and **Dr. O. S. Deepa Gopakumar**, Associate Professor, Department of Mathematics, Amrita School of Engineering, Coimbatore. To the best of my knowledge this work has not formed the basis for the award of any Degree/Diploma/Associateship/Fellowship/or a similar award to any candidate in any University.

Place: **Signature of the student**

Date:

COUNTERSIGNED

Dr. R. Subash Moorthy, Thesis Advisor, Assistant Professor

Dr. O. S. Deepa Gopakumar, Thesis Co-Advisor, Associate Professor

Department of Mathematics

ACKNOWLEDGEMENT

I would like to express my deep gratitude to our beloved **Satguru Sri Mata Amritanandamayi Devi** for providing a brilliant academic climate at this college, which has made this entire task achievable. This acknowledgement is intended to be an expression of gratitude to everyone involved directly or indirectly with my project. I would like to thank our Pro-Chancellor **Bramachari Abhayamrita Chaitanya**, Vice Chancellor **Dr. P. Venkat Rangan** and **Dr. Sasangan Ramanathan,** Dean of Engineering of Amrita Vishwa Vidyapeetham, for providing us the necessary infrastructure required for the completion of the project.

I express my thanks to **Dr. K. Somasundaram**, Chairperson of the Department of Mathematics, for his valuable help and support during our study. I express my gratitude to my guide, **Dr. R. Subash Moorthy**, Assistant Professor and **Dr. O. S. Deepa Gopakumar**, Associate Professor, Dept. of Mathematics, for their guidance, support and supervision.

I feel extremely grateful to **Dr. Maroju Prashanth, Dr. R. Radha Iyer,** and **Dr. Sukanta Nayak** for their feedback and encouragement which helped me to complete the project. I would like to thank **Dr. R. Radha Iyer,** for scheduling and organising periodic reviews during the course of my project. I also thank the entire staff of the Department of Mathematics.

I would like to extend my sincere thanks to my family and friends for helping and motivating me during the course of the project. Finally, I would like to thank all those who have helped, guided and encouraged me directly or indirectly during the project work. Last, but not least, I thank **God** for His blessings which made my project a success.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTD.) **PAGE**

LIST OF ABBREVIATIONS

KNN – K Nearest Neighbours

LDA – Linear Discriminant Analysis

QDA – Quadratic Discriminant Analysis

k-NN – k-nearest neighbours

ANN – Artificial Neural Networks

PCA – Principal Component Analysis

ANOVA – Analysis of Variance

ABSTRACT

In recent days, decision support system is widely used in predicting medical diagnosis in order to save time, reduce cost and to overcome errors arising due to various reasons. K-Nearest Neighbour algorithm is simple to understand but works incredibly well in practice. But this non-parametric lazy learning algorithm has many disadvantages. To overcome the disadvantages of KNN, this paper proposes an algorithm which is hybrid of K Nearest Neighbour and discriminant analysis. Two types of discriminant analysis are considered: Linear discriminant analysis and Quadratic discriminant analysis along with KNN. A decision with linear boundary is easy to foresee and apprehend. So, the prediction method using LDA had proved to work well in practice. QDA has more predictability power than LDA. Algorithms on KNN-LDA, KNN-QDA, Condensed KNN-LDA, Condensed KNN-QDA, Fuzzy KNN LDA, Fuzzy KNN-QDA, Constrained Fuzzy LDA, Constrained Fuzzy, Rough Fuzzy LDA and Rough Fuzzy QDA were implemented and validated on medical dataset. The various algorithms based on KNN-LDA and KNN QDA is compared with existing KNN.

CHAPTER 1

INTRODUCTION TO MACHINE LEARNING

1.1 Introduction

Data mining is a process of extracting knowledge from large amounts of data stored either in databases or other information repositories. It is a vital procedure where smart techniques are applied to extract data patterns which can be out-looked from diverse fields. Classification is a data mining technique to predict class of unknown instances. Data mining is the process is to obtain information from the collected data set and convert it into a comprehensible mode. It is a computational procedure of finding various outlines in huge data sets comprising of statistics and machine learning techniques. Data mining involves six common classes of tasks - Anomaly detection, Association rule learning, Clustering, Classification, Regression, Summarization. Classification is a foremost procedure in data mining and extensively used in numerous areas. Classification is used to classify each item in a set of data into one of predefined set of classes or groups. The data analysis task classification where in a model or classifier is created to predict categorical labels. The goal of classification is to accurately predict the target class for each case in the data. Before going into the algorithms, we will give an overview about Machine Learning and K-Nearest Neighbour, Linear Discriminant Analysis, Quadratic Discriminant Analysis.

1.2 Definition of Machine Learning

Machine Learning gives the computer the ability to learn and automatically perform operations without being programmed. This is done without any assistance from human beings. We observe the various patterns of data and make decisions based on the data patterns. The term machine learning was coined by Arthur Samuels in the year 1959. It is often viewed as a subset of artificial intelligence.

1.3 Theory

A core objective of a learner is to generalize from its experience. Generalization in this context is the ability of a learning machine to perform accurately on new data sets after having experienced a learning data set. The training examples come from some generally unknown probability distribution which is considered representative of the space of occurrences and the learner has to build a general model about this space that enables it to produce sufficiently accurate predictions in new cases.

1.4 Types of Machine Learning Algorithms

There are different types of machine learning algorithms, namely:

a) Supervised Machine Learning Algorithms - Takes labelled or classified data from the past and makes prediction for future data values. Supervised learning algorithms include classification and regression. Classification algorithms are used when the class labels are restricted to a limited set of values, and regression algorithms are used when the class labels may have any numerical value within a range.

Examples:

- Bayesian Statistics
- Naïve Bayes Classifier
- Nearest Neighbour Algorithm
- Decision Tree Learning

Applications:

- Ranking
- Recommendation Systems
- Face Verification
- Speaker Verification
- Visual Identity Tracking

b) Unsupervised Machine Learning Algorithms - Takes data which is neither labelled nor classified and draws conclusions based on it. These algorithms are known for identifying some commonalities in the training data set and then acting based on the presence or absence of them in the testing data set.

Examples:

- K-Means Clustering Algorithm
- Hierarchical Clustering
- Neural networks
- Anomaly Detection

Application:

- Density Estimation

c) Semi-supervised Machine Learning Algorithms – These algorithms fall somewhere in between supervised and unsupervised, i.e., uses both classified and unclassified data.

Examples:

- Generative Models
- Low-Density Separation
- Graph-Based Models
- Heuristic Approaches

d) Reinforcement Machine Learning Algorithms - Machines detect its environment and determine the most ideal behaviour. Many reinforcement machine learning algorithms use dynamic programming techniques. Reinforcement learning algorithms are used in autonomous vehicles or in learning to play a game against a human opponent.

Examples:

- End-to-end reinforcement learning
- Inverse reinforcement learning
- Apprenticeship learning

Applications:

- Game Theory
- Control Theory
- Operations Research
- Information Theory
- Simulation-Based Operations
- Multi-Agent Systems
- Swarm Intelligence
- Statistics
- Genetic Algorithms

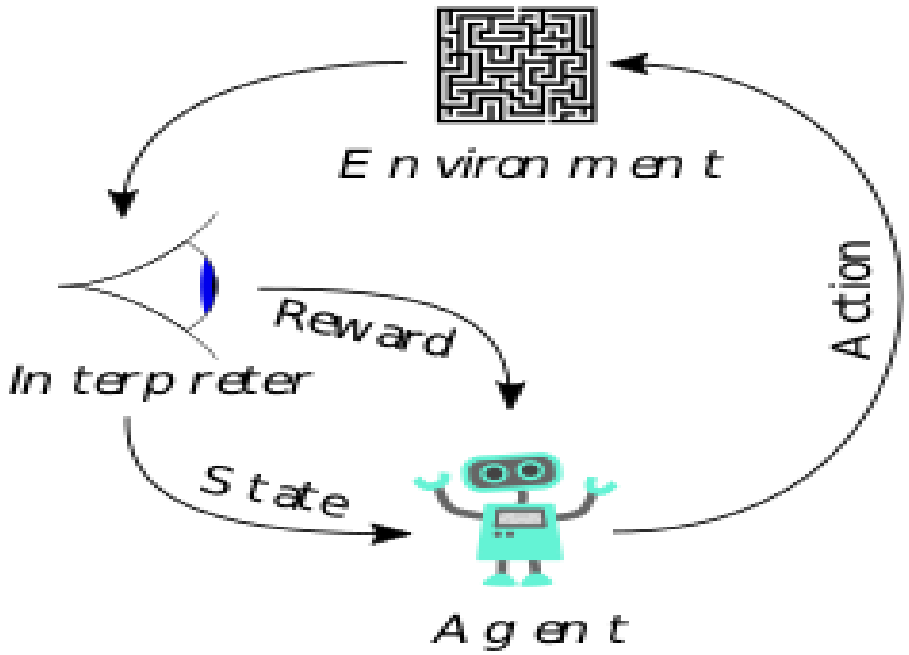


Figure 1.1 Example for Reinforcement Machine Learning Algorithms

1.5 Processes and Techniques of Machine Learning

a) Feature Learning – These algorithms transform data and represent them in a more meaningful manner before performing any classifications or predictions on them. At the same time however, these algorithms also preserve the original data. This technique allows reconstruction of the inputs coming from the unknown data-generating distribution, while not being necessarily faithful to configurations that are implausible under that distribution.

Examples:

- Principal Components Analysis (PCA)
- Cluster Analysis

Feature Learning Algorithms can be supervised or unsupervised.

Examples of Supervised Feature Learning:

- Artificial Neural Networks
- Multilayer Perceptrons
- Supervised Dictionary Learning

Examples of Unsupervised Feature Learning:

- Independent Component Analysis
- Auto-encoders
- Matrix Factorization
- Various Forms of Clustering

The main motivation behind Feature Learning is that classification often requires data that is mathematically and computationally convenient to process.

b) Sparse Dictionary Learning – In this method, the training data set is represented as a linear combination of basis functions, assumed to be a sparse matrix.

Applications:

- In classification, we use it to find the class label of an unknown data sample which doesn't belong to the training data set.
- For a dictionary where each class has already been built, a new training example is associated with the class that is best sparsely represented by the corresponding dictionary.
- It is also useful in removing the noise from an image.
- Sparse Dictionary Learning is a variation of Feature Learning. It is strongly *NP-Hard* and difficult to solve approximately.

Figure 1.2 Example for Sparse Dictionary Learning

c) Anomaly Detection – It is the process of looking for outliers in the given data. These outliers refer to data sets in the training set that differ significantly from all the other training data sets. These data items can represent anything such as bank fraud, deviation in the performance of a sportsperson, etc. There are three broad categories of Anomaly Detection –

i. In **Supervised Anomaly Detection Techniques**, we require a data set where we have labels normal and abnormal and hence, we need to train the classifier

ii. In **Unsupervised Anomaly Detection Techniques**, we assume that a majority of the instances in the data set are normal.

iii. **Semi-supervised Anomaly Detection Techniques** construct a model representing normal behaviour from a given normal training data set, and then test the likelihood of a test instance to be generated by the model.

d) Decision Trees - This is a predictive model which uses decision trees to go from information on the given data set to conclusions about the class label of the given data set. The information about the data set is denoted by the branches in the decision tree and the conclusions about the data set, which tells us the class label, is depicted by the leaves in the decision tree. There are two types of Decision Trees –

i. Classification Trees

ii. Regression Trees

In **Classification Trees**, the class labels are discretely valued while **Regression Trees** use continuously valued class labels.

Applications:

- Statistics
- Data Analytics
- Machine Learning

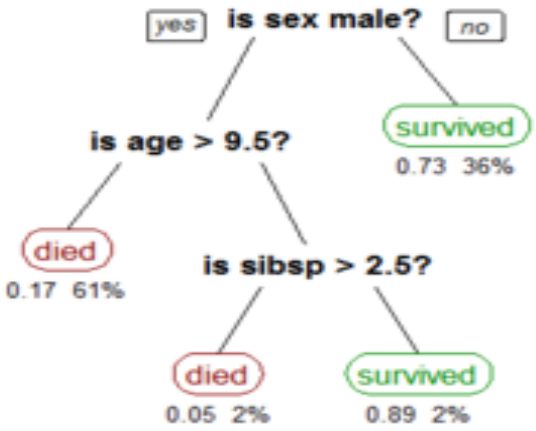


Figure 1.3 Example for Decision Trees

1.6 Model

a) Artificial Neural Networks (ANN) – This model is a framework for many different machine learning algorithms to work together and perform complex tasks. These systems learn to perform tasks by considering examples and are not governed by any specific rules. ANN is a model based on a collection of connected units or nodes called Artificial Neurons. These neurons are loosely modelled in a biological brain. Each connection, or edge, transmits a signal from one artificial neuron to another. The neuron that receives the signals can process it and send it to other neurons.

Applications:

- Computer Vision
- Speech Recognition
- Machine Translation
- Social Network Filtering
- Playing Board
- Video Games
- Medical Diagnosis

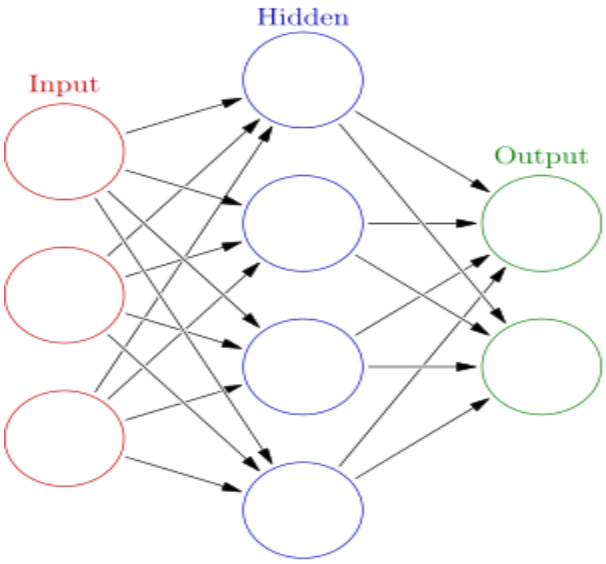


Figure 1.4 Overview of ANN

1.7 Model Assessments

The machine learning models can be validated by using accuracy estimation techniques. There are many different such techniques, namely:

a) Holdout Method – In this method, we split the data into a training data set and a testing data set. Conventionally, the training data set covers 2/3 of the set and the testing data set covers the other 1/3 of the data set. The performance of the training model on the training set is then evaluated.

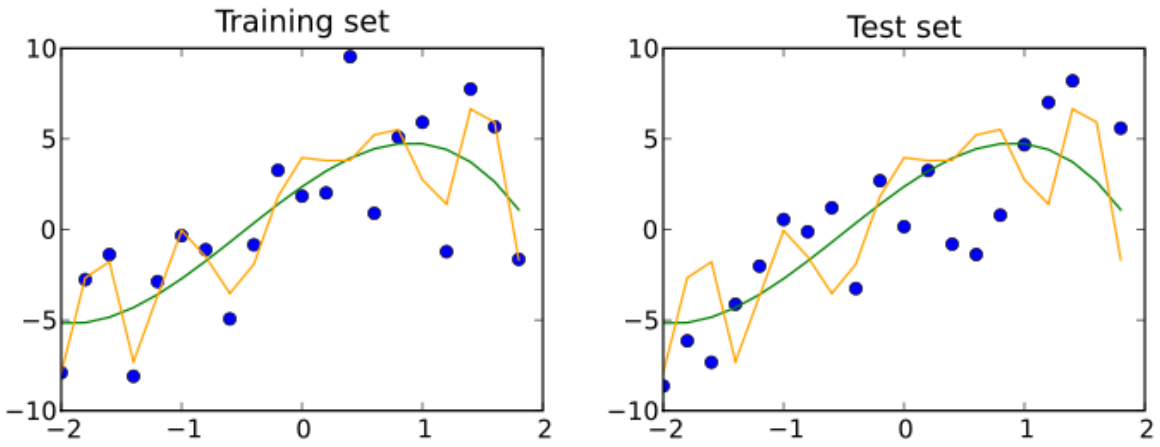


Figure 1.5 Overview of Holdout Method

b) N-Fold-Cross-Validation Method – The data is randomly split into *k* subsets, where *k-1* data subsets are used to train the model and the k^{th} subset is used to test the predictive ability of the training model.

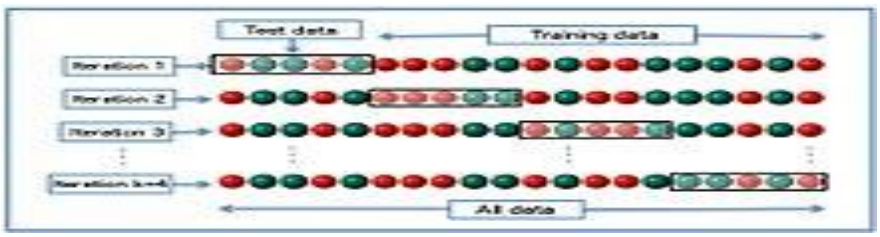


Figure 1.6 Overview of N-Fold-Cross-Validation Method

1.8 Limitations of Machine Learning

In spite of proving useful in many fields, machine learning has often failed to deliver the expected results. Some of the reasons for this include lack of data, lack of access to data, data bias, privacy problems, badly chosen tasks and algorithms, wrong tools and people, lack of resources and evaluation problems.

Data bias - Machine learning approaches in particular can suffer from data bias. This is because the information fed into the machine is provided by human beings. Hence, a machine which is trained on man-made data is likely to have the same kind of biases as human beings.

Ethics – The data bias, as mentioned above, raises many ethical questions in machine learning. When a machine is trained by a human being filled with bias, then it will also learn bias and act in a biased manner.

1.9 Examples

Medical diagnosis

Machine learning can be used in the techniques and tools that can help in the diagnosis of diseases. It is used for the analysis of the clinical parameters and their combination for the prognosis example prediction of disease progression for the extraction of medical knowledge for the outcome research, for therapy planning and patient monitoring. These are the successful implementations of the machine learning methods. It can help in the integration of computer-based systems in the healthcare sector.

Classification

A classification is a process of placing each individual under study in many classes. Classification helps to analyze the measurements of an object to identify the category to which that object belongs. To establish an efficient relation, analysts use data. For example, before a bank decides to distribute loans, it assesses the customers on their ability to pay loans. By considering the factors like customer's earnings, savings, and financial history, we can do it. This information is taken from the past data on the loan.

Prediction

Machine learning can also be used in the prediction systems. Considering the loan example, to compute the probability of a fault, the system will need to classify the available data in groups. It is defined by a set of rules prescribed by the analysts. Once the classification is done, we can calculate the probability of the fault. These computations can compute across all the sectors for varied purposes. Making predictions is one of the best machine learning applications.

Extraction

Extraction of information is one of the best applications of machine learning. It is the process of extracting structured information from the unstructured data. For example, the web pages, articles, blogs, business reports, and emails. The relational database maintains the output produced by the information extraction. The process of extraction takes a set of documents as input and outputs the structured data.

Regression

We can also implement machine learning in the regression as well. In regression, we can use the principle of machine learning to optimize the parameters. It can also be used to decrease the approximation error and calculate the closest possible outcome. We can also use the machine learning for the function optimization. We can also choose to alter the inputs in order to get the closest possible outcome.

LIST OF KEYWORDS

- K-Nearest Neighbour
- Discriminant Analysis
- Medical Diagnosis
- Fuzzy K-Nearest Neighbour
- Decision Support System

CHAPTER 2

K-NN AND DISCRIMINANT ANALYSIS

2.1 Definition of KNN Algorithm

k*-nearest neighbours algorithm** (k*-NN**) is a non-parametric method used for classification and regression. In both cases, the input consists of the *k* closest training examples in the feature space. The output depends on whether *k*-NN is used for classification or regression:

- In **k-NN classification**, the output is a class membership. An object is classified by a majority vote of its neighbours, with the object being assigned to the class most common among its *k* nearest neighbours (*k* is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of that single nearest neighbour.
- In **k-NN regression**, the output is the property value for the object. This value is the average of the values of its *k* nearest neighbours.

For the purpose of this project, we will be dealing only with the classification part. An overview of the KNN algorithm is given on the next page.

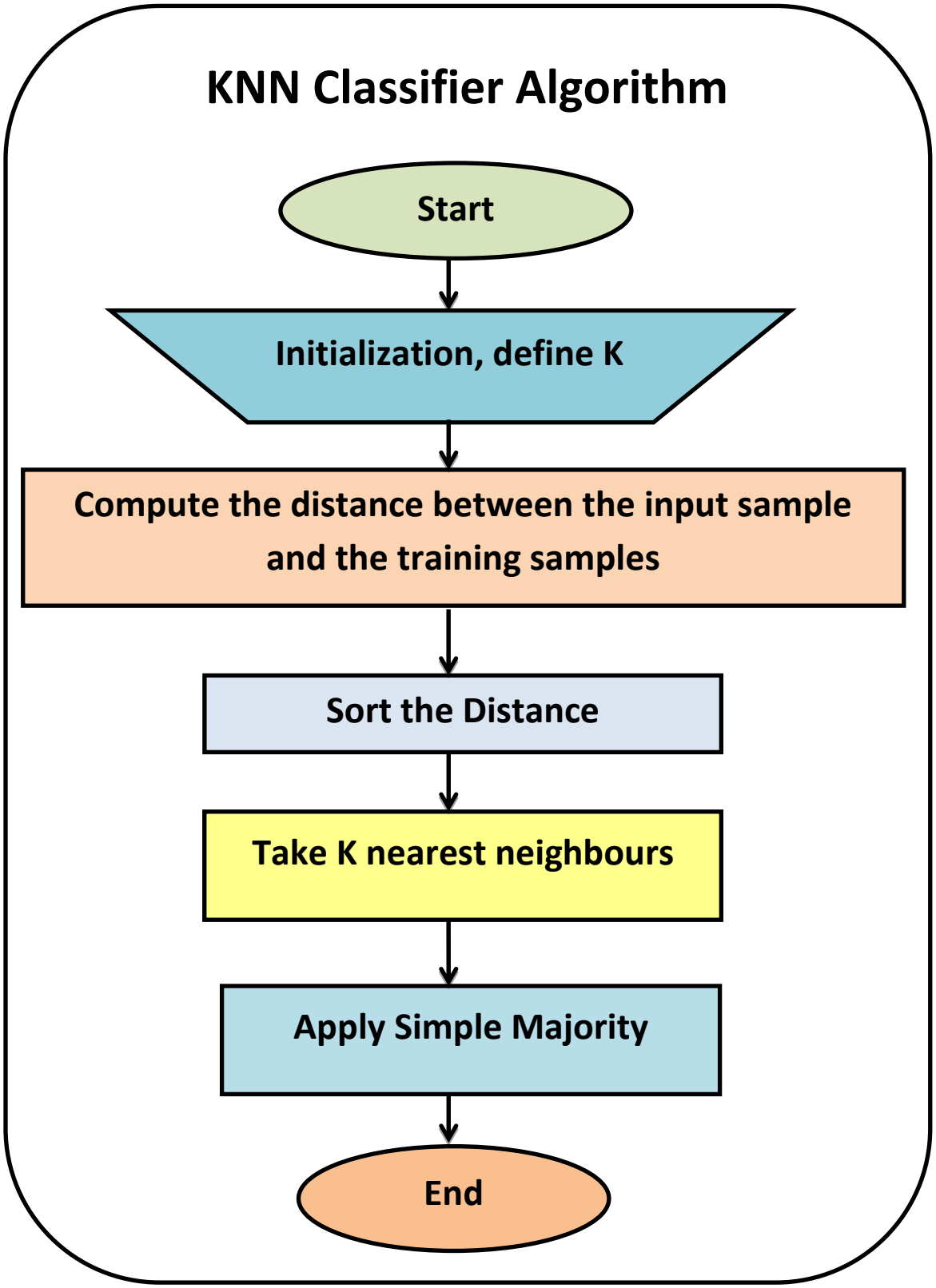


Figure 2.1 Overview of KNN Classifier

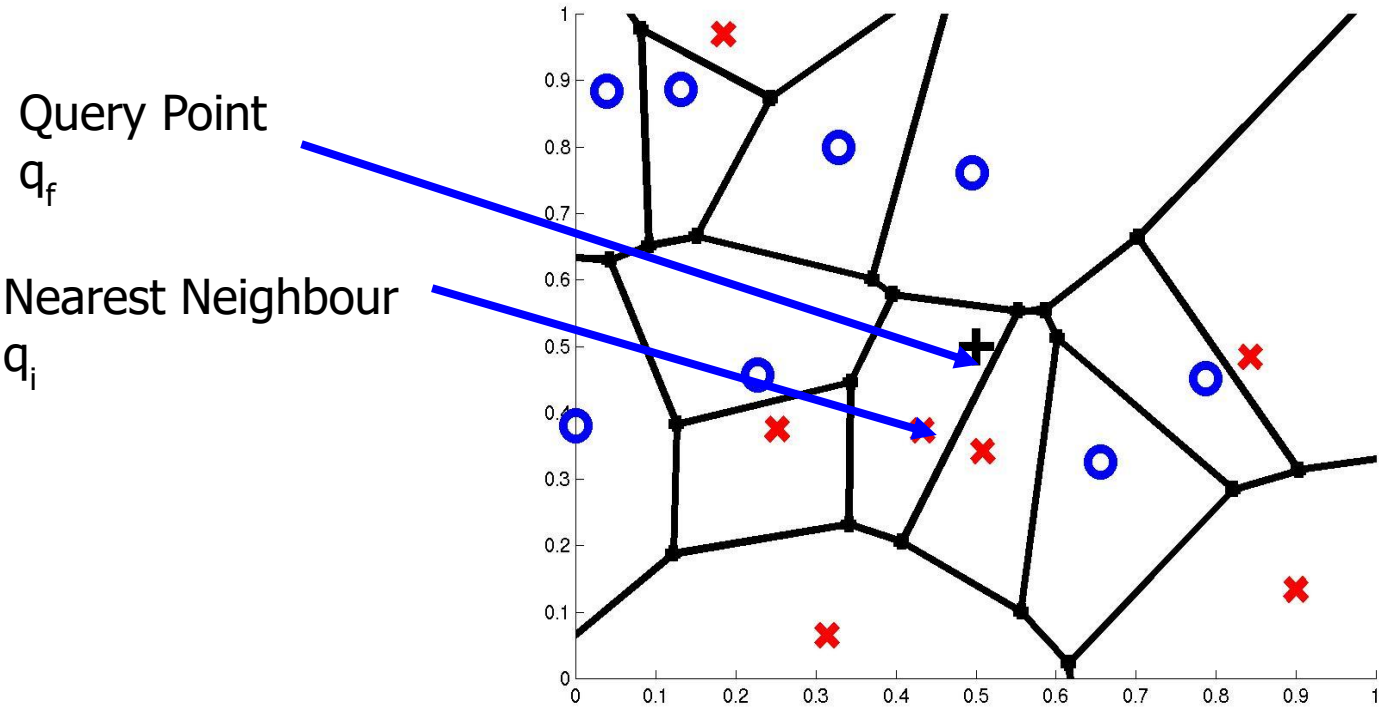


Figure 2.2 1-Nearest Neighbours

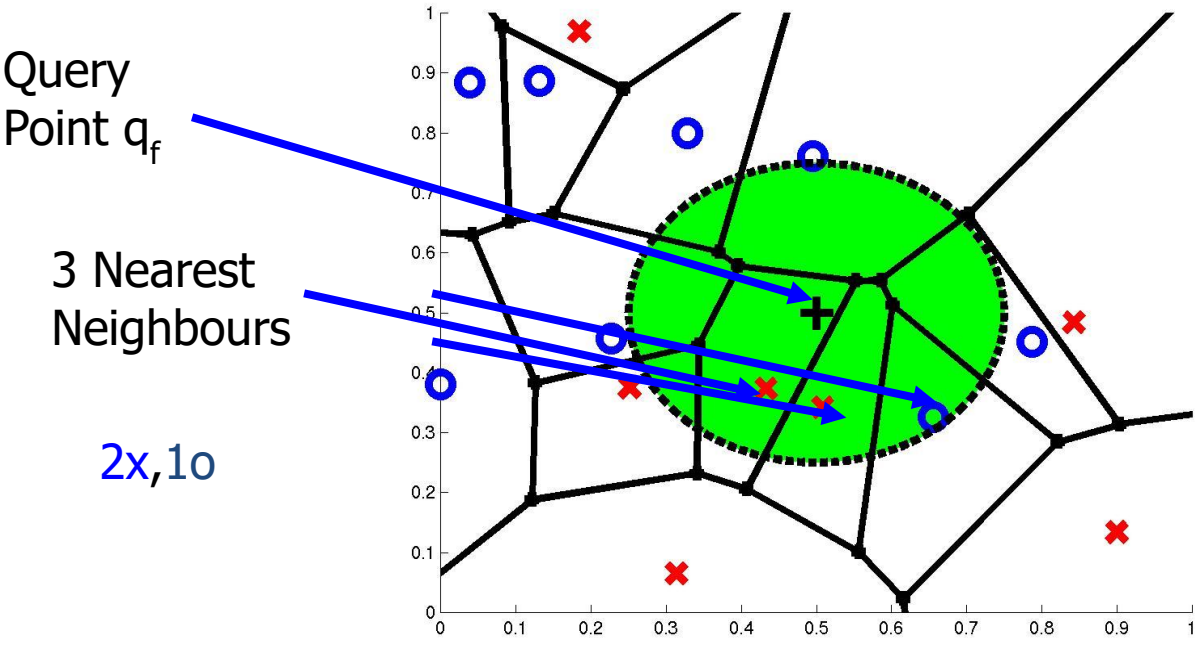


Figure 2.3 3-Nearest Neighbours

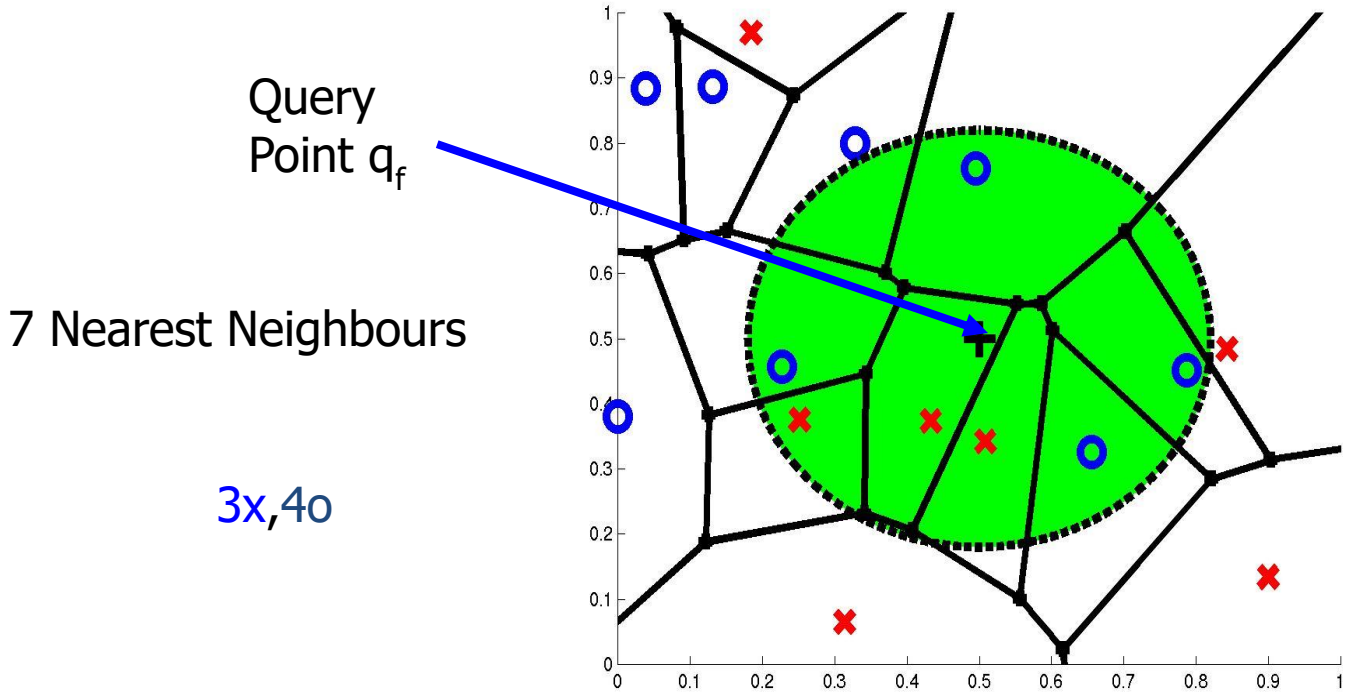


Figure 2.4 7-Nearest Neighbours

2.2 Advantages of KNN Algorithm

- Easy to implement
- Building model is inexpensive
- Extremely flexible classification scheme
- Well suited for multi-modal classes (classes of multiple forms) and records with multiple class labels
- Nearest Neighbour classification expects class conditional probability to be locally constant

2.3 Disadvantages of KNN Algorithm

- It takes time to classify unknown data points
- The distance computation of the k-nearest neighbours is quite intensive, especially when the size of the training set grows

- Accuracy can be severely degraded by the presence of noisy or irrelevant features
- Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes

2.4 Ways to Overcome the Disadvantages of KNN

- Choose k as an odd number to avoid any ties in class labels among the nearest neighbours
- If the value of k is too small, then the classification for the testing data sample will be too sensitive to points in the data which cause noise
- If the value of k is too large, then the neighbourhood for the testing data sample will consist of points that belong to other classes
- Start with k=1 and use a test set to validate the error rate of the classifier
- Repeat with k=k+2
- Choose the value of k for which the error rate is minimum
- Discard features that are irrelevant to the process

2.5 Definition of LDA Algorithm

Linear Discriminant Analysis (LDA) is a method that helps us find a linear combination of features that splits two classes of objects from each other. The linear combination obtained as a result of LDA can be used as a linear classifier or for dimensionality reduction before classification. LDA is a generalization of Fischer's Linear Discriminant, a classification method, developed by and named after R.A. Fischer in the year 1936. LDA is used in both dimensionality reduction and as a classifier. It can be used for both multiclass and binary classification. For the purpose of our project however, we will be looking at how LDA is used only for binary classification.

2.6 Working of LDA for Two classes

It is simple, mathematically robust and often produces models whose accuracy is as good as more complex methods. LDA is based upon the concept of searching for a linear combination of variables (predictors) that best separates the two classes (targets). Given a score function, the problem is to estimate the linear coefficients that maximize the score.

1. Take m sample vectors with n parameters say $X_1, X_2 \ldots\ldots, X_n$ that are assigned any one of two different classes, $c_1 and\ c_2$. Call these as training data sets.
2. Take another sample vector$k$, with the same n parameters. We shall call this a testing data set.
3. For each class, store the belonging training data sets in a matrix, where each row of the matrix is a training data set. Call these as class matrices, $C_1 and\ C_2$.
4. Calculate the average data set for the training data set in each matrix, say, $avgC_1 and\ avgC_2$, and for the distance vectors in both matrices, say avg.
5. For each of the class matrices, create new matrices, say C_{01}andC_{02}, which are obtained by calculating the distance vector between each row of the class matrices and the average vector, avg.
6. For each class we can calculate the covariance matrix using the formula $(G * G^T)/$ $(number\ of\ rows\ in\ G)$ for any given matrix G. In this case, we shall apply the formula to the matrices C_{01} and C_{02} and call the resulting covariance matrices $CovC_1$ and $CovC_2$.
7. Calculate the pooled covariance matrix, Cov, which is given by the formula ∑(no of rows in each distance matrix * covariance matrix)/∑ (no of rows in each distance matrix).
8. Apply this formula to distance matrices $d1$ & $d2$ and covariance matrices $CovC_1$and $CovC_2$.
9. Calculate the linear model coefficient vector, which is in this case, $\beta = (avgC_1$ - $avgC_2)/\ (Cov\ )$.

10. Let $X = (K - ((avgC_1 + avgC_2)/2))\beta^T$ and $Y = -\log((number\ of\ rows\ in\ C_1/(number\ of\ rows\ in\ C_2))$.
11. If $X > Y$, then assign the class C_1 to the testing data set. Otherwise, assign the class C_2.

2.7 Assumptions of LDA Algorithm

- The classes are normally distributed
- The co-variances of both the classes are equal
- Multivariate Normality - Independent variables are normal for each level of the grouping variable.
- Multi Co-Linearity - Predictive power can decrease with an increased correlation between predictor variables.
- Independence - Participants are assumed to be randomly sampled, and a participant's score on one variable is assumed to be independent of scores on that variable for all other participants.

2.8 Fischer's Linear Discriminant

It is actually similar to LDA, but the discriminant is slightly different and does not make some of the same assumptions that the LDA does.

2.9 Closely Related Fields

2.9.1 Analysis of Variance (ANOVA)

ANOVA is a procedure used to analyze the difference between group means in a sample. It was developed by Ronald Fisher, a statistician. The type of data that is analyzed is experimental in nature. This method can be used to test any number of group means. However, there is a difference between ANOVA and LDA. In ANOVA, the independent variables are of categorical

data type and the dependent variables are continuous data type. On the other hand, LDA uses continuous independent variables and categorical dependent variables.

2.9.2 Regression Analysis

Regression Analysis is a statistical model that is used to express a dependent variable in terms of one or more independent variables. This will help us to understand the relationship between the variables and how the behaviour of the dependent variable changes based on the changes in the independent variable(s). The relationship between the dependent variable and the independent variable(s) is usually written as a regression equation. For example, $z=a+by+cx$, where z is a dependent variable and x and y are independent variables, a is the regression constant and b and c are known as the regression coefficients. This makes it very easy for us to predict and forecast the values of dependent variables based on a given set of independent variables. Regression can either be linear or non-linear in nature.

2.9.3 Principal Component Analysis (PCA)

This procedure uses an orthogonal transformation method to convert a group correlated variables into a group of linearly uncorrelated variables. These variables are known as principal components. The first principal component has the largest variance and each principal component thereafter, has the largest variance given that it is orthogonal to all preceding principal components. This method is mostly used as a tool in exploratory data analysis and for making predictive models. The difference between PCA and LDA is that LDA attempts to differentiate between classes while PCA doesn't.

2.9.4 Factor Analysis

This is a statistical method used to reduce the number of variables by searching for interdependence among the variables. It describes the variability among the observed, correlated

variables in terms of a potentially lower number of unobserved variables known as factors. This is where it is different from LDA. LDA is not an interdependence technique. It is a technique used to make distinctions between the independent variables and the dependent variables. It has several applications in fields such as biology, personality theories, marketing, product management, operations research and finance.

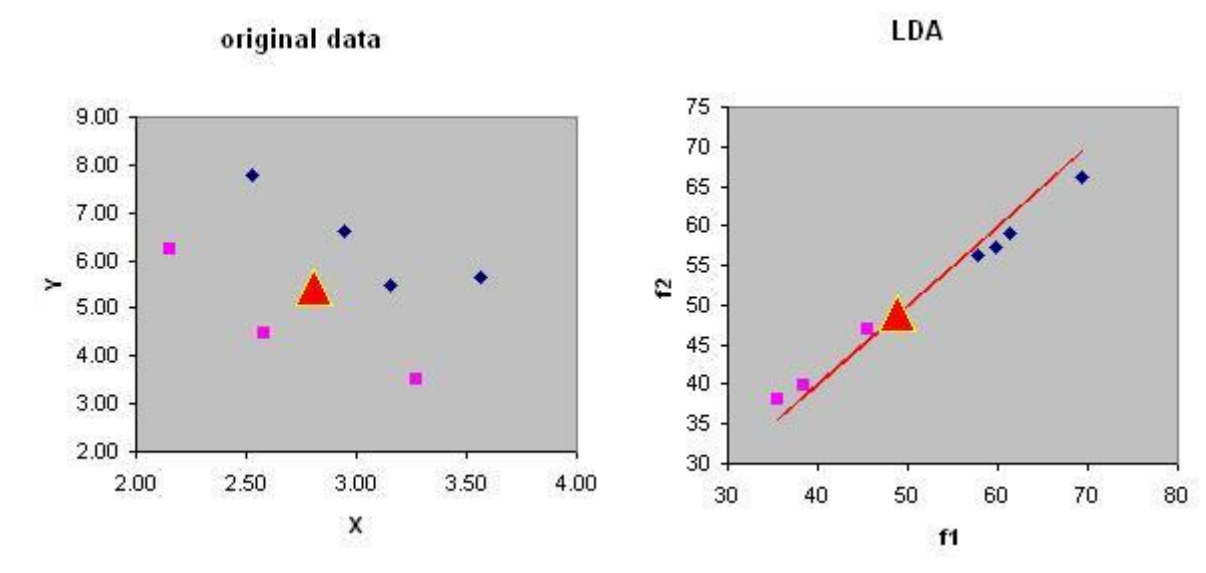


Figure 2.5 Overview of LDA

2.10 Quadratic Discriminant Analysis

Quadratic Discriminant Analysis (QDA) is a more general version of LDA. Like LDA, this method also separates two or more classes from one another. But instead of a line, it is a quadric surface that acts as a boundary separating the different classes. This method assumes that the measurements from each class are normally distributed just like in LDA but does not assume that the co-variance of all the classes is equal. When the assumption of normality is true, then we predict the likely class for the data set using the likelihood estimation test, which will be part of our working procedure given below. QDA is most commonly used in machine learning and is considered as one of the statistical classification technique.

2.11 Working of QDA

As you will notice, the QDA will almost resemble the LDA except for the last three steps.

1. Take m sample vectors with n parameters say$X_1, X_2 \ldots\ldots, X_n$ that are assigned any one of two different classes say $c_1, c_2 \ldots . c_n$. Call these as training data sets.
2. Take another sample vector, say k, with the same n parameters. We shall call this a testing data set.
3. For each class, store the belonging training data sets in a matrix, where each row of the matrix is a training data set. Call these as class matrices, say $C_1, C_2 \ldots . C_n$
4. Calculate the average data set for the training data set in each matrix, say $avgC_1, avgC_2 \ldots . avgC_n$ and for the distance vectors in both matrices, say avg.
5. For each of the class matrices, create new matrices, say $C_{01}, C_{02} \ldots . C_{0n}$ which are obtained by calculating the distance vector between each row of the class matrices and the average vector, avg.
6. For each class we can calculate the covariance matrix using the formula $(G * G^T)/(number\ of\ rows\ in\ G)$ for any given matrix G. In this case, we shall apply the formula to the matrices $C_{01}, C_{02} \ldots . C_{0n}$ and call the resulting covariance matrices$CovC_1, CovC_2 \ldots . CovC_n$.
7. Calculate the pooled covariance matrix, say Cov which is given by the formula $\sum_{i=1}^{n}(number\ of\ rows\ in\ each\ distance\ matrix * Covariance\ Matrix)/\sum_{i=1}^{n}(number\ of\ rows\ in\ each\ distance\ matrix)$
8. Apply this formula to distance matrices $C_{01}, C_{02} \ldots . C_{0n}$ and covariance matrices $CovC_1, CovC_2 \ldots . CovC_n$.

9. For each class, calculate

$$ZC_1 =$$

$$(-0.5 * (\left(1 - \left(\frac{avg\,C_1}{Cov\,C_1}\right)\right) * (-avgC_1)^T - \log(\det(CovC_1))) +$$

$$\log((number\ of\ rows\ in\ the\ distance\ matrix\ of\ \left(\frac{C_1}{k}\right))$$

10. Apply this same formula to classes$c_1, c_2 \ldots . c_n$.

11. Compare the values calculated in step 9.

12. Assign the testing data set with the class having the greatest value calculated in step 9.

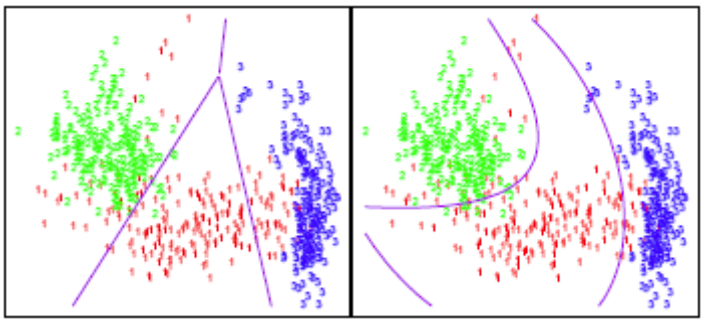


Figure 2.6 Overview of QDA

2.12 Usage of LDA and QDA

Generally LDA tends to be a better bet than QDA if there are relatively few training observations and so has substantially lower variance. In contrast, QDA is recommended if the training set is very large, so that the variance of the classifier is not a major concern, or if the assumption of a common covariance matrix is clearly untenable. QDA has more predictability power than LDA but it needs to estimate the covariance matrix for each classes.

CHAPTER 3

HYBRID K-NN IN MEDICAL DIAGNOSIS

KNN is a nonparametric method for classifying unknown data i.e. it does not make any assumptions on underlying data distribution. Halil Yigit had worked on ABC- based distance–weighted kNN algorithm [7]. Xuejun Ma, et.al [13] studied on a variant of K nearest neighbour quantile regression. Some of the related works based on KNN and Discriminant Analysis can be found in [1, 8, 9, 10, 11, 12].

In this Paper, 15 Algorithms are carried out. Out of the 15 Algorithms, three are existing Algorithms, which are compared with 12 proposed Algorithms of K-Nearest Neighbour (KNN) with Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA). The Algorithms used are:

1. KNN Algorithm (Existing technique)

2. KNN LDA Algorithm

3. KNN QDA Algorithm

4. Fuzzy KNN Algorithm (Existing technique)

5. Fuzzy KNN LDA

6. Fuzzy KNN QDA

7. Condensed KNN Algorithm (Existing technique)

8. Condensed KNN LDA Algorithm

9. Condensed KNN QDA Algorithm

10. Constrained KNN Algorithm

11. Constrained KNN LDA Algorithm

12. Constrained KNN QDA Algorithm

13. Rough Fuzzy KNN Algorithm

14. Rough Fuzzy KNN LDA Algorithm

15. Rough Fuzzy KNN QDA Algorithm

3.1 KNN Algorithm

1. Take m sample vectors with n parameters, say, $X_1, X_2 \ldots\ldots, X_n$ which can be assigned to any one of n different classes, say, $c_1, c_2 \ldots . c_n$. These are considered as training data sets.
2. For each of training and testing data the distance is calculated by taking the sum of the squares of the distance between each parameter of the training and testing data, i.e., given two vectors $x_1, x_2, \ldots . . x_n$ and $y_1, y_2, \ldots . . y_n$ the distance is given by $(x_1 - y_1)^2, (x_2 - y_2)^2, \ldots . . (x_n - y_n)^2$.
3. Take the k training data sets with the least values of distance from the training data set.
4. From the k training data sets obtained above, count how many of these belong to each class.
5. Assign the testing data set with the class which is assigned to the maximum number of the k training data sets. If two classes are jointly assigned to the maximum number of the k training data sets, then find the closest of the k Neighbours belonging to one of these two classes and assign that class to the testing data set.

3.2 KNN-LDA Algorithm

1. Take m sample vectors with n parameters say $X_1, X_2 \ldots\ldots, X_n$ that are assigned any one of two different classes say $c_1 and\ c_2$. Call these as training data sets.
2. Take another sample vector, say k, with the same n parameters. We shall call this a testing data set.
3. For each of training and testing data the distance is calculated by taking the sum of the squares of the distance between each parameter of the training and testing data, i.e., given

two vectors $x_1, x_2, \ldots . x_n$ and $y_1, y_2, \ldots . y_n$ the distance is given by $(x_1 - y_1)^2, (x_2 - y_2)^2, \ldots . (x_n - y_n)^2$.

4. Take the k training data sets with the least values of distance from the training data set.
5. From the k training data sets obtained above, count how many of these belong to each class.
6. For each class, store the belonging training data sets in a matrix, where each row of the matrix is a training data set. Call these as class matrices, say, $C_1 and\ C_2$.
7. Calculate the average data set for the training data set in each matrix, say, $avgC_1 and\ avgC_2$, and for the distance vectors in both matrices, say avg.
8. For each of the class matrices, create new matrices, say C_{01} and C_{02}, which are obtained by calculating the distance vector between each row of the class matrices and the average vector, avg.
9. For each class we can calculate the covariance matrix using the formula $(G * G^T)/(number\ of\ rows\ in\ G)$ for any given matrix G. In this case, we shall apply the formula to the matrices C_{01} and C_{02} and call the resulting covariance matrices $CovC_1$ and $CovC_2$.
10. Calculate the pooled covariance matrix Cov, which is given by the formula $\sum (no\ of\ rows\ in\ each\ distance\ matrix * covariance\ matrix)/\sum (no\ of\ rows\ in\ each\ distance\ matrix)$. Apply this formula to distance matrices $d1$ & $d2$ and covariance matrices $CovC_1$ and $CovC_2$.
11. Calculate the linear model coefficient vector, which is in this case, $\beta = (avgC_1 - avgC_2)/(Cov)$.
12. Let $X = (K - ((avgC_1 + avgC_2)/2))\beta^T$ and $Y = -\log((number\ of\ rows\ in\ C_1/(number\ of\ rows\ in\ C_2))$.
13. If $X > Y$, then assign the class C_1 to the testing data set. Otherwise, assign the class C_2.

3.3 KNN-QDA Algorithm

1. Take m sample vectors with n parameters say$X_1, X_2 \ldots\ldots, X_n$ that are assigned any one of two different classes say $c_1, c_2 \ldots . c_n$. Call these as training data sets.
2. Take another sample vector, say k, with the same n parameters. We shall call this a testing data set.
3. For each of training and testing data the distance is calculated by taking the sum of the squares of the distance between each parameter of the training and testing data, i.e., given two vector $x_1, x_2, \ldots . . x_n$ and $y_1, y_2, \ldots . . y_n$ the distance is given by $(x_1 - y_1)^2, (x_2 - y_2)^2, \ldots . . (x_n - y_n)^2$.
4. Take the k training data sets with the least values of distance from the training data set.
5. From the k training data sets obtained above, count how many of these belong to each class.
6. For each class, store the belonging training data sets in a matrix, where each row of the matrix is a training data set. Call these as class matrices, say $C_1, C_2 \ldots . C_n$
7. Calculate the average data set for the training data set in each matrix, say $avgC_1, avgC_2 \ldots . avgC_n$ and for the distance vectors in both matrices, say avg.
8. For each of the class matrices, create new matrices, say $C_{01}, C_{02} \ldots . C_{0n}$ which are obtained by calculating the distance vector between each row of the class matrices and the average vector, avg.
9. For each class we can calculate the covariance matrix using the formula $(G * G^T)/(number\ of\ rows\ in\ G)$ for any given matrix G. In this case, we shall apply the formula to the matrices $C_{01}, C_{02} \ldots . C_{0n}$ and call the resulting covariance matrices $CovC_1, CovC_2 \ldots . CovC_n$.

10. Calculate the pooled covariance matrix, say Cov which is given by the formula $\sum_{i=1}^{n}(number\ of\ rows\ in\ each\ distance\ matrix * Covariance\ Matrix)/ \sum_{i=1}^{n}(number\ of\ rows\ in\ each\ distance\ matrix)$

11. Apply this formula to distance matrices $C_{01}, C_{02} \ldots . C_{0n}$ and covariance matrices $CovC_1, CovC_2 \ldots . CovC_n$.

12. For each class, calculate:

$$ZC_1 = (-0.5 * ((1 - \left(\frac{avgC_1}{CovC_1}\right)) * (-avgC_1)^T - \log(\det(CovC_1))) + \log((number\ of\ rows\ in\ the\ distance\ matrix\ of\ \left(\frac{C_1}{k}\right))$$

Apply this same formula to classes$c_1, c_2 \ldots . c_n$.

13. Compare the values calculated in step 12.

14. Assign the testing data set with the class having the greatest value calculated in step 12.

3.4 Fuzzy KNN Algorithm

This algorithm focuses on the membership of each of the training data sets in each class. Our assignment of a class to the testing data set depends upon the results of the calculation involving the membership of all the training data sets. In our algorithm, we assign the membership of a data set as 1 for the class to which it belongs and 0 for all the other classes. We then use KNN criteria for assigning the class.

Steps 1 to 4 are the same as in 3.1

5. For each class, create a Boolean vector which assigns a value 1 to a training data set if it belongs to that class and 0 if it doesn't.

6. For each class, calculate the value

$$\sum(number\ of\ rows\ in\ each\ distance\ matrix * Covariance\ Matrix) \\ /(distance\ of\ each\ training\ data\ set\ from\ the\ testing\ data\ set)$$

These values are considered as $XC_1, XC_2 \ldots . XC_n$.

7. Let $\sum_{i=1}^{n} 1/(distance\ of\ each\ training\ data\ set\ from\ the\ testing\ data\ set)$. For each class, calculate the value $= X/Y$. Let these values be $ZC_1, ZC_2 \ldots . ZC_n$.
8. Assign the class with the highest Z-value to the testing data set.
9. If two of the classes both share the highest Z -value, then find the closest of the k neighbours belonging to one of these two classes and assign that class to the testing data set.

3.5 Fuzzy KNN LDA Algorithm

This algorithm involves a combination of the Fuzzy KNN Algorithm and the KNN LDA Algorithm. This algorithm can be used only if there are two classes.

1. Take m sample vectors with n parameters say $X_1, X_2 \ldots\ldots, X_n$ that are assigned any one of two different classes say $c_1 and\ c_2$. Call these as training data sets.
2. Take another sample vector, say k, with the same n parameters. We shall call this a testing data set.
3. For each of training and testing data the distance is calculated by taking the sum of the squares of the distance between each parameter of the training and testing data i.e., given two vectors $x_1, x_2, \ldots\ldots x_n$ and $y_1, y_2, \ldots\ldots y_n$ the distance is given by $(x_1 - y_1)^2, (x_2 - y_2)^2, \ldots\ldots (x_n - y_n)^2$.
4. Take the k training data sets with the least values of distance from the training data set.

5. From the k training data sets obtained above, count how many of these belong to each class.
6. For each class, store the belonging training data sets in a matrix, where each row of the matrix is a training data set. Call these as class matrices, say $C_1 and\ C_2$.
7. Calculate the average data set for the training data set in each matrix, say $avgC_1 and\ avgC_2$ and for the distance vectors in both matrices, say avg.
8. Multiply the distance vectors of each of the distance matrices of each class by their membership in that class. We shall call these matrices as $d1$ & $d2$.
9. For each of the class matrices, create new matrices, say C_{01} and C_{02}, which are obtained by calculating the distance vector between each row of the class matrices and the average vector, avg.
10. For each class we can calculate the covariance matrix using the formula $(G * G^T)/(number\ of\ rows\ in\ G)$ for any given matrix$G$. In this case, we shall apply the formula to the matrices C_{01} and C_{02} and call the resulting covariance matrices $CovC_1$ and $CovC_2$.
11. Calculate the pooled covariance matrix, say Cov , which is given by the formula $\sum_{i=1}^{n}(number\ of\ rows\ in\ each\ distance\ matrix * Covariance\ Matrix)/\sum_{i=1}^{n}(number\ of\ rows\ in\ each\ distance\ matrix)$. Apply this formula to distance matrices $d1$ & $d2$ and covariance matrices $CovC_1$ and $CovC_2$.
12. Calculate the linear model coefficient vector, which is in this case,

    $\beta = (avgC_1\text{-}avgC_2)\ /\ (Cov\ )$
13. Let $X = (K - ((avgC_1 + avgC_2)/2))\beta^T$ and

    $Y = -\log((number\ of\ rows\ in\ C_1/(number\ of\ rows\ in\ C_2))$.
14. If $X > Y$, then assign the class C_1 to the testing data set. Otherwise, assign the class C_2.

3.6 Fuzzy KNN QDA Algorithm

This algorithm involves a combination of the Fuzzy KNN Algorithm and the KNN QDA Algorithm. This algorithm can be used for two or more classes.

1. Take m sample vectors with n parameters say $X_1, X_2 \ldots\ldots, X_n$ that are assigned any one of two different classes say $c_1, c_2 \ldots . c_n$. Call these as training data sets.
2. Take another sample vector, say k, with the same n parameters. We shall call this a testing data set.
3. For each of training and testing data the distance is calculated by taking the sum of the squares of the distance between each parameter of the training and testing data, i.e., given two vectors $x_1, x_2, \ldots . . x_n$ and $y_1, y_2, \ldots . . y_n$ the distance is given by $(x_1 - y_1)^2, (x_2 - y_2)^2, \ldots . . (x_n - y_n)^2$.
4. Take the k training data sets with the least values of distance from the training data set.
5. From the k training data sets obtained above, count how many of these belong to each class.
6. For each class, store the belonging training data sets in a matrix, where each row of the matrix is a training data set. Call these as class matrices, say $C_1, C_2 \ldots . C_n$
7. Calculate the average data set for the training data set in each matrix, say $avgC_1, avgC_2 \ldots . avgC_n$ and for the distance vectors in both matrices, say avg.
8. Multiply the distance vectors of each of the distance matrices of each class by their membership in that class. We shall call these matrices as $d_1, d_2 \ldots . d_n$
9. For each of the distance matrices, create a new matrices, say $C_{01}, C_{02} \ldots . C_{0n}$, which are obtained by calculating the distance vector between each row of the distance matrices and the average vector, avg.
10. For each class we can calculate the covariance matrix using the formula $(G * G^T)/(number\ of\ rows\ in\ G)$for any given matrix G. In this case, we shall apply the formula to

the matrices $C_{01}, C_{02} \ldots . C_{0n}$ and call the resulting covariance matrices $CovC_1, CovC_2 \ldots . CovC_n$.

11. Calculate the pooled covariance matrix, say Cov which is given by the formula $\sum_{i=1}^{n}(number\ of\ rows\ in\ each\ distance\ matrix * Covariance\ Matrix)/$ $\sum_{i=1}^{n}(number\ of\ rows\ in\ each\ distance\ matrix)$
12. Apply this formula to distance matrices $d_1, d_2 \ldots . d_n$ and covariance matrices $CovC_1, CovC_2 \ldots . CovC_n$
13. For each class, calculate

    $ZC_1 =$

    $(-0.5 * (\left(1 - \left(\frac{avg\,C_1}{Cov\,C_1}\right)\right) * (-avgC_1)^T - \log(\det(CovC_1))) +$

    $\log((number\ of\ rows\ in\ the\ distance\ matrix\ of\ \left(\frac{C_1}{k}\right))$
14. Apply this same formula to classes $c_1, c_2 \ldots . c_n$.
15. Compare the values calculated in step 13.
16. Assign the testing data set with the class having the greatest value calculated in step 13.

3.7 Condensed KNN Algorithm:

This algorithm involves the use of a combination of K-means algorithm along with KNN algorithm. In this algorithm, we arrange the data into clusters and remove the outliers in each cluster. Then, we perform the KNN algorithm on the remaining data sets.

3.8 Condensed KNN LDA Algorithm:

This algorithm is a combination of the K-means algorithm followed by the KNN LDA algorithm. In this process, we do one iteration of the K-means algorithm. By choosing a certain number of centroids, say C centroids, we split the training data set into C clusters. In each cluster, we check

the classes of each data set and we remove the data of the minority class, which we call outliers. After removing the outliers, we then apply the KNN LDA algorithm to the remaining training data sets.

3.9 Condensed Fuzzy QDA algorithm:

The process is similar to condensed Fuzzy KNN LDA. After removing the outliers, we then apply the KNN QDA algorithm to the remaining training data sets.

3.10 Constrained KNN algorithm

This algorithm is based on the principle of assigning partial membership in a class to a testing data set. The membership of a testing data set can take on any real values in the interval [0,1].

3.11 Constrained Fuzzy KNN LDA Algorithm:

This algorithm is a combination of the K-means algorithm followed by the KNN LDA algorithm. In this process, we do one iteration of the K-means algorithm. By choosing a certain number of centroids, say C centroids, we split the training data set into C clusters. In each cluster, we check the classes of each data set and we remove the data of the minority class, which we call outliers. After removing the outliers, we then apply the KNN LDA algorithm to the remaining training data sets.

3.12 Constrained Fuzzy KNN QDA Algorithm:

It is the same as Fuzzy QDA except that at the end we take the percentage of the X and Y values at the end of the algorithm and represent this as the membership of the testing data set in each class.

3.13 Rough Fuzzy KNN algorithm:

This algorithm is based on calculating ownership of each class towards the testing data set. The class with the highest ownership towards the testing data set is assigned to the testing data set

3.14 Rough Fuzzy KNN LDA Algorithm:

It is the same as the Fuzzy LDA algorithm except that we use weighted distances for calculating the X and Y values at the end of each algorithm instead of the ordinary distances.

3.15 Rough Fuzzy KNN QDA Algorithm:

It is the same as the Fuzzy QDA algorithm except that we use weighted distances for calculating the X and Y values at the end of each algorithm instead of the ordinary distances.

3.16 Implementation

We used an original medical data set to test the accuracy of these algorithms. This data is taken from herbal plants. The compounds have been identified for all the plants. Certain properties for the drugs based on the compounds have also been identified. We identified 23 such properties. This data set contains information for 71 herbal plants. We performed the classification using the above 15 algorithms for the data of 72 herbal plants.

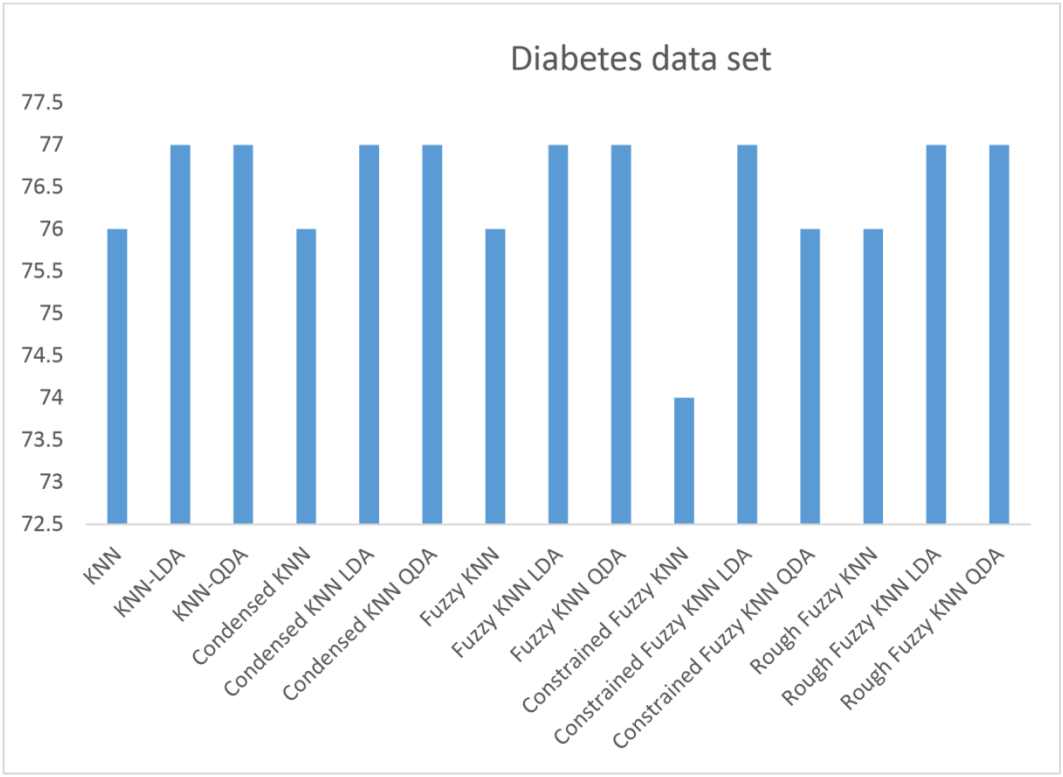
The Drug likeness of chemical compounds considers Lipinski's Rule as main criteria. It is an important rule to determine Drug likeness or decide if a chemical compound with certain molecular descriptors would make it an orally effective Drug in humans. The rule is significant during Drug discovery process. The rule depicts properties critical for a Drug's pharmacokinetics in the human body, including their absorption, distribution, metabolism, excretion and toxicity. The dataset consists of molecular descriptors of chemical compounds and also consists of the status of satisfying the above specified rules. The dataset focuses on chemical compounds available in medicinal plants and their molecular descriptors. The working of this project is based on the study of molecular descriptors for Drug/Non-Drug compounds extracted from medicinal plants. The molecular descriptors of these chemical compounds are identified by Dr. O. S. Deepa Gopakumar and Ani R. They have studied a lot on the classification of the compounds extracted from the medicinal plants [2, 3, 4, 5, 6].

Descriptions of the same can be found in [15, 16]. The machine learning approaches like classification of these compounds to drug compounds and non drug compounds are done by Kormaz, Selcuk, Gokmen Zararsiz, and Dincer Goksusluk [14] had done research on the same. Descriptions of machine learning approaches are also found in [17]. A comparison of different classification algorithms in the prediction of drug likeness of chemical compounds are carried out in this study. The molecular descriptors are used for the prediction of drug likeness.

We also considered a diabetes data set and cancer dataset from UCI repository. The diabetes dataset has 20 attributes and cancer data set has 9 attributes. The nine different classification algorithms are carried out and various performance measures are calculated. In this paper, the result is validated with 10-fold cross-validation technique. The diabetes and cancer dataset data set is randomly divided into 10 partitions and one partition is considered as testing dataset and others are considered as training dataset. The 15 algorithms are implemented on both testing and training dataset and the accuracy is noted. (Please see Table 3.1 on next page)

Table 3.1: Accuracy assessment for 15 algorithms

	Correctly classified instances	Incorrectly classified instances	Correctly classified instances	Incorrectly classified instances
	Diabetes	Diabetes	Cancer	Cancer
KNN	76	24	96	4
KNN-LDA	77	23	97	2
KNN-QDA	77	23	97	2
Condensed KNN	76	24	90	10
Condensed KNN LDA	77	23	92	8
Condensed KNN QDA	77	23	99	1
Fuzzy KNN	76	23	97	3
Fuzzy KNN LDA	77	23	99	1
Fuzzy KNN QDA	77	23	99	1
Constrained Fuzzy KNN	74	24	90	10
Constrained Fuzzy KNN LDA	77	23	92	8
Constrained Fuzzy KNN QDA	76	24	99	1
Rough Fuzzy KNN	76	24	90	10
Rough Fuzzy KNN LDA	77	23	92	8
Rough Fuzzy KNN QDA	77	23	99	1



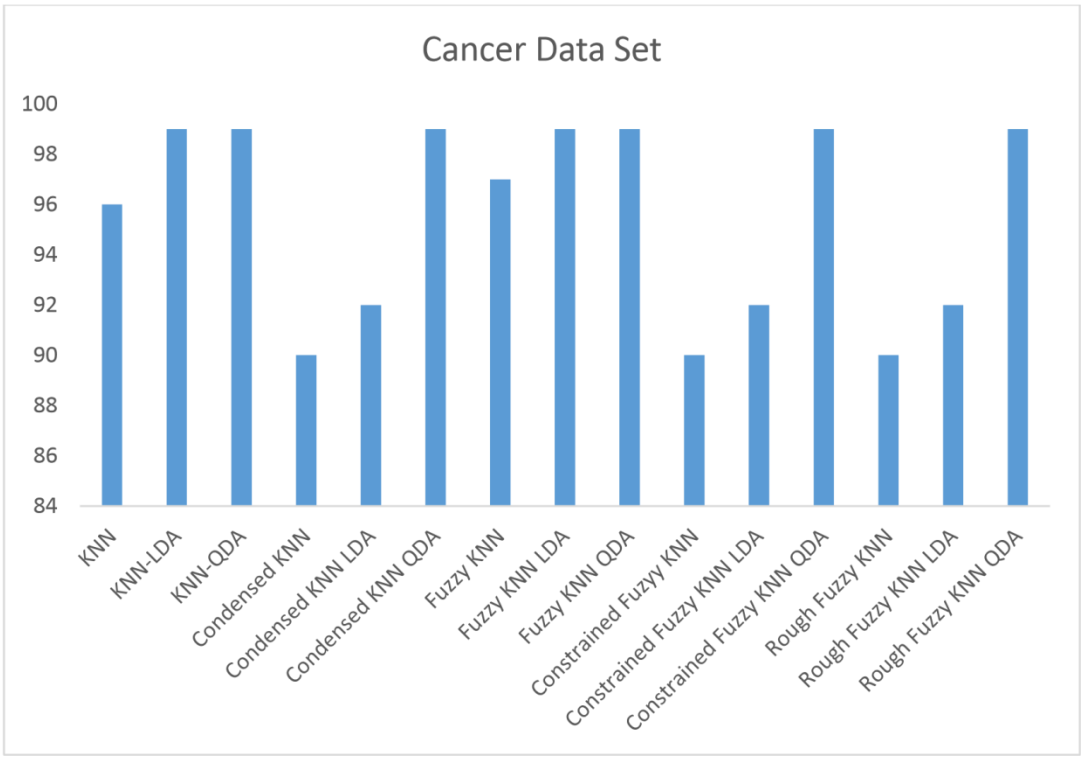


Figure 3.1 Comparison of 15 algorithms on cancer and diabetes datasets

Table 3.2: Accuracy assessment for 15 algorithms using Herbal Plants Data Set

Algorithm	Accuracy %	Correct	Wrong
KNN	75	54	18
LDA	75	54	18
QDA	75	54	18
KNN LDA	95.83	69	3
KNN QDA	86.11	62	10
Fuzzy KNN	75	54	18
Fuzzy LDA	94.4	68	4
Fuzzy QDA	94.4	68	4
Rough Fuzzy KNN	75	54	18
Rough Fuzzy KNN LDA	95.83	69	3
Rough Fuzzy KNN QDA	95.83	69	3
Constrained Fuzzy KNN	75	54	18
Constrained Fuzzy KNN LDA	95.83	69	3
Constrained Fuzzy KNN QDA	86.11	62	10
Condensed Fuzzy KNN	75	54	18
Condensed Fuzzy KNN LDA	94.4	68	4
Condensed Fuzzy KNN QDA	94.4	68	4

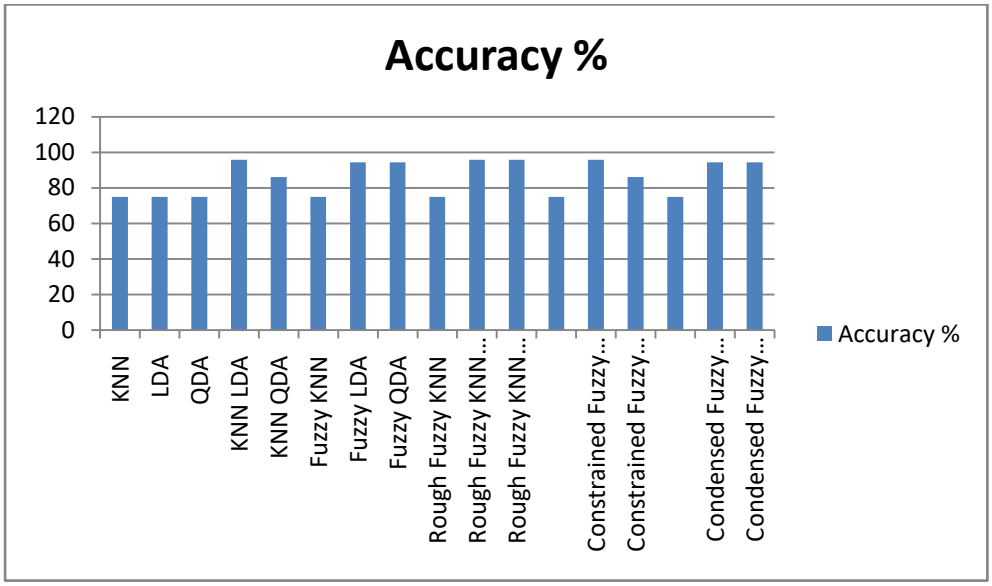


Figure 3.2 Accuracy Percentages for 15 algorithms using Herbal Plants data set

CONCLUSION

For all the three data sets used above, Diabetes, Cancer and Herbal Plants, the KNN LDA, KNN QDA, Condensed KNN LDA, Condensed KNN QDA, Fuzzy KNN LDA, Fuzzy KNN QDA, Constrained Fuzzy KNN LDA, Constrained Fuzzy KNN QDA, Rough Fuzzy KNN LDA and Rough Fuzzy KNN QDA showed a better performance than KNN, Condensed KNN, Constrained KNN and Rough Fuzzy KNN.

From the cancer data set it is found that the accuracy performance of LDA and QDA with respect to Condensed, Fuzzy, Constrained Fuzzy and Rough Fuzzy were high. Also accuracy of QDA on Condensed Fuzzy, Constrained Fuzzy, and Rough Fuzzy over whelmed the accuracy with respect to LDA and hence can be concluded that the use of KNN QDA on Condensed Fuzzy, Constrained Fuzzy and Rough Fuzzy is preferable.

The prediction for diabetes using various algorithms was around 77% and can be improved by using few boosting techniques or by increasing the number of samples. So the computer based decision support systems can be used based on Fuzzy KNN- LDA, Fuzzy KNN- QDA, KNN-LDA and KNN -QDA so as to reduce cost and errors in clinical trials.

In the herbal plants data set, we find the LDA to be more effective than QDA with respect to ordinary KNN as well as with respect to Constrained Fuzzy. With respect to Fuzzy, Rough Fuzzy and Condensed Fuzzy, LDA and QDA are equally effective with higher accuracy in Rough Fuzzy than in Fuzzy and Condensed Fuzzy.

REFERENCES

1. Aiman Moldagulova, Rosnafisha Bte Sulaiman, Using KNN algorithm for classification of textual documents, IEEE Xplore October 2017.
2. Ani R. and O.S. Deepa, Rotation forest ensemble algorithm for the classification of phyto-chemicals from the medicinal plants, Journal of chemical and pharmaceutical science, pp. 14-17, Special issue 4, 2016.
3. Ani R., Jose J. ,Wilson M., Deepa O.S.: Modified rotation forest ensemble classifier for medical diagnosis in decision support systems Advances in Intelligent Systems and Computing,564, pp. 137-146.
4. Ani R., Krishna S., Anju N., Sona A.M., Deepa O.S.: IoT based patient monitoring and diagnostic prediction tool using ensemble classifier, 2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017, 2017 January, pp. 1588-1593.
5. Ani R. and Deepa O.S.: Rotation forest ensemble algorithm for the classification of phyto-chemicals from the medicinal plants, Journal of Chemical and Pharmaceutical Sciences,2016(SpecialIssue4), pp. 6-9
6. Deepa O.S. and Ani R.: Expectation - Maximization algorithm for protein - Ligand complex of HFE gene, Journal of Chemical and Pharmaceutical Sciences, pp. 14-17, 2016(SpecialIssue4),
7. Halil Yigit: ABC – based distance - weighted kNN algorithm, Journal of Experimental and Theoretical Artificial Intelligence, Vol.27, issue 2, 2015.
8. Liwen Huang and Lianta Su: Hierarchical Discriminant analysis and its application, Liwen Huang and Lianta Su, Communication in Statistics – Theory and Methods, Vol 42, issue 11, 2013.

9. P. Kalaivani, K.L. Shumuganathan: An improved K – nearest neighbour algorithm using genetic algorithm for sentiment classification, IEEE Xplore, March 2015.

10. P.T. Pepler, D.W Uys and D.G. Nel; Discriminant analysis under the common principal components model, Communication in statistics - simulation and computations, vol 46, issue 6, Feb 2017.

11. Shweta Taneia, Charu Gupta, Kratika Goval, Dharna Gureia: An Enhanced K-Nearest Neighbour Algorithm Using Information Gain and clustering, IEEE Xplore April 2014.

12. Wei-Yin Loh and Nutal Vanichesetakal: Journal of the American Statistical Association, Vol 83, issue 403, Mar 2012 Tree-Structured Classification via Generalized Discriminant Analysis.

13. Xuejun Ma, Xiaogun He and Xiaokang Shi : A variant of K nearest neighbour quantile regression, Journal of Applied Statistics, Vol 43,issue 3 ,2016.

14. Kormaz, Selcuk, Gokmen Zararsiz, and Dincer Goksusluk. "Drug/nondrug classification using support vector machines with various feature selection strategies." computer methods and programs in biomedicine 117.2(2014): 51-60.

15. Cano, Gaspar, et al. "Automatic selection of molecular descriptors using random forest: Application to Drug discovery." Expert Systems with Applications 72(2017): 151-159.

16. Rodriguez, Juan Jose, Ludmila I. Kuncheva, and Carlos J. Alonso. 'Rotation forest: A new classifier ensemble method." IEEE transactions on pattern analysis and machine intelligence 28.10(2006): 1619-1630.

17. Lavecchia, Antonio. "Machine-learning approaches in Drug discovery: methods and applications." Drug discovery today 20.3(2015): 318-331.