K-Nearest Leader Follower Classifier

Rakesh Ranjan

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Orissa, India

K-Nearest Leader Follower Classifier

Thesis submitted in

May 2013

to the department of

Computer Science and Engineering

of

National Institute of Technology Rourkela

in partial fulfillment of the requirements

for the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Rakesh Ranjan

[Roll: 109CS0568]

under the guidance of

Mr. Bidyut Kumar Patra

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Orissa, India

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Orissa, India.

May 2013

Certificate

This is to certify that the work in the thesis entitled ***K-Nearest Leader Follower Classifier*** by ***Rakesh Ranjan*** is a record of an original research work carried out under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Dr Bidyut Kumar Patra
Assistant Professor
CSE department of NIT Rourkela

Acknowledgment

The attainment and final outcome of this project required a lot of supervision and assistance from many people and I am extremely privileged to have got this all along the completion of my project.Whatever I have done is only due to such guidance and assistance and I would not forget to express thanks to them. My earnest gratitude goes to my thesis supervisor, **Dr. Bidyut Kumar Patra**, Assistant Professor, Department of CSE, for his support, guidance, inspiration and encouragement throughout the period this work was carried out. His willingness for meeting at all times, his lucrative comments, his concern and support even with practical things have been very helpful. I would also like to thank all instructors and professors, and members of the department of Computer Science and Engineering for their warm help in various ways for the completion of this thesis. A vote of appreciation for my fellow friends for their cooperation.

Rakesh Ranjan

Abstract

In pattern recognition, the k-nearest neighbor classifier (k-NN) is a non-parametric approach for classifying test cases based on closest training set elements in the given dataset. k-NN belongs to the category of instance-based learning, or lazy learning where the function is estimated locally and all computation is delayed until classification. The k-nearest neighbor algorithm is supposed to be easiest of all machine learning algorithms, in which an object is classified by a bulk vote of its neighbors, with the object being allotted to the class most common amongst its k nearest neighbors. If k = 1, then the object is merely assigned to the class of its nearest neighbor. Nearest Neighbor classifier and its variants like k-Nearest Neighbor (KNN) classifier are popular because of their simplicity and good performance. But one of the major limitations of KNN is that it has to search the entire training set in order to classify a given pattern which proves to be very expensive when a big sized training set is given or dimension of training set is high. In this paper, a generalized k-nearest leader follower classification algorithm is presented, which is an improvement over k-nearest neighbor rule. It can find the class of a test dataset in less time with respect to traditional Nearest-neighbor and KNN.

The method is to find a set of leaders with the concept of clustering using a predefined minimum distance between two leaders as tou. After finding leaders set, rest of the remaining training set is classified into follower set, with a group of followers assigned to each leader. Now the k-nearest leaders will be found out of the obtained leader set by finding the distance of a given test set from the leader set. And finally using these k-nearest leaders and their corresponding followers, class of the dataset is found. The algorithm is empirically tested with some standard data sets and a comparison is made with some of the earlier classifiers.

Contents

List of Abbreviations

NN	Nearest Neighbor
KNN	K-Nearest Neighbor
CNN	Condensed Nearest Neighbor
FCNN	Fast Condensed Nearest Neighbor
WKNL	Weighted K-Nearest Leader
KNLF	K-Nearest Leader Follwer

List of Figures

List of Tables

Chapter 1

Introduction

Prompt developments in data gathering and storage technology have empowered organizations to collect enormous quantities of data. However, mining useful information has proven particularly challenging. Often, customary data analysis tools and practices cannot be used because of the enormous size of a data set. At times, the non-traditional behavior of the data means that traditional methods cannot be applied even if the data set is fairly small. In other conditions, the query that need to be answered cannot be addressed using current data mining techniques, and thus, new methods need to be established.

1.1 Data mining

Data Mining is the process of finding a specific pattern in a given data set. This method includes use of automated data analysis techniques to expose the re-

lationship between the data items. This data analysis technique needs the involvement of many different algorithms. The overall objective of the data mining process is to excerpt information from an available data set and transform it into a human-understandable form for future use. The knowledge discovery process consists of numerous steps, viz. data selection, data cleaning, integration, transformation, data mining, pattern estimation and knowledge depiction. The initial four steps are different methods of data preprocessing, where data is set for data mining. The data mining step is a necessary step where data analysis technique is applied to extract patterns. The extracted pattern is then evaluated in process evaluation step and is then represented before the user in knowledge representation step. Basic data mining tasks are classification, regression, prediction, time-series-analysis, clustering, association, summarization, and sequence formation.

Data mining is a technique which deals with the extraction of unknown analytical information from large datasets. It uses different algorithms for the process of categorization of large amount of data and finding relevant information. Data mining tools can predict future trends and behaviors, it allows busi-

nesses to make practical, knowledge-driven decisions. With the amount of data increasing each year, more data is gathering and data mining is becoming a vital tool to transform this data into information. Long course of research and product development has evolved data mining. This evolution started when business data were first stored on big computers, continued with various improvements in data access and extraction, and more lately, generated technologies that allow users to obtain any data in real time. Data mining takes this evolutionary step beyond simple data access and navigation to prospective and practical information supply.

1.2 Data Classification

Classification is the task of allocating an unknown object to one of the several pre-defined categories. It is a pervasive problem that incorporates various applications. Various examples like email spam detection based upon the e-mail message header and its content, determining cancer by categorizing cells as malignant or benign based upon the results of MRI scans, and classifying galaxies based upon position and shapes explain the concept of data classification. Classification is the task of learning a target function f that maps each attribute set x to one of the prede-

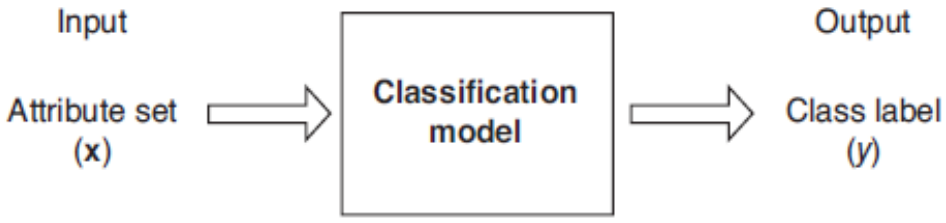

Figure 1.1: Classification as the task of mapping an input attribute set x into its class label y

fined class labels y. The target function is also known informally as classification model.

1.2.1 General Approach to Solving a Classification Problem

A classification technique (or a classifier) is a methodical approach of building classification models from given input data set. Some of the examples include decision tree classifiers, rule-based classifiers, support vector machines, neural networks, and Naive-Bayes classifiers. Each technique employs a specific algorithm to find a model that best fits the association between the attributes and class label of the input dataset. The model produced by any learning algorithm should fit the input data well and also predict the class labels of test records correctly. Therefore, the main aim of the learning algorithm is to build a model having good generalization capability; i.e., models that precisely determine the class labels of previously unidentified records.

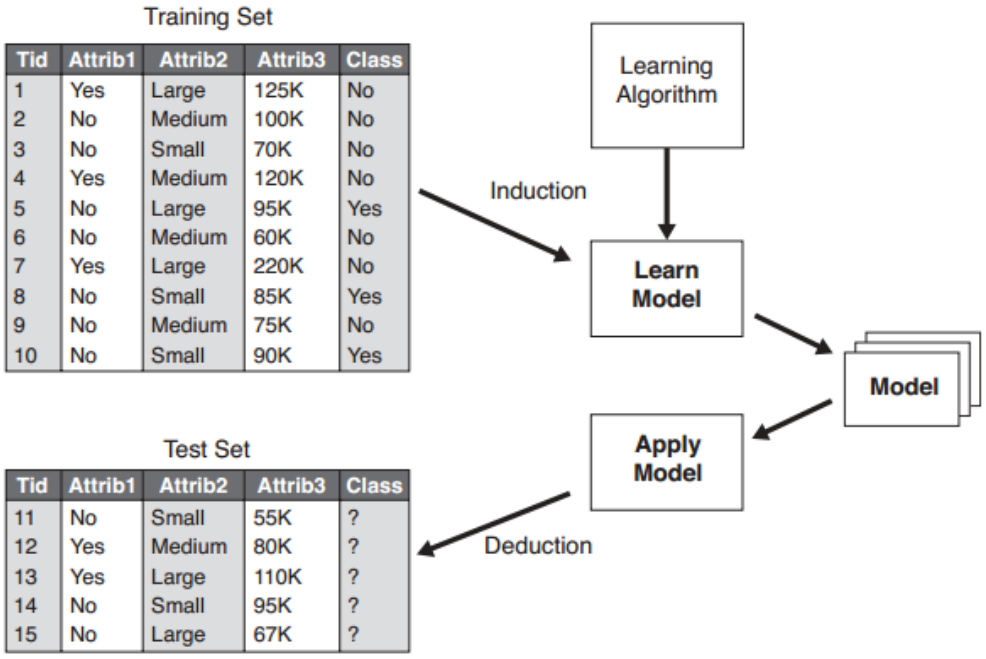

Figure 1.2: General approach for building a classification model

1.3 Data Clustering

Partition of data into different groups of alike objects is called clustering. Sometimes a few particulars are lost by representing the data with less number of clusters but it attains simplification. Data is modelled using clusters. Data modeling puts clustering in a rhetorical perspective rooted in mathematics, statistics, and numerical analysis. According to machine learning perspective, clusters correspond to unknown patterns, the searching for clusters belongs to the class of unsupervised learning, and resulting system represents an important data concept. From a practical viewpoint clustering plays an important

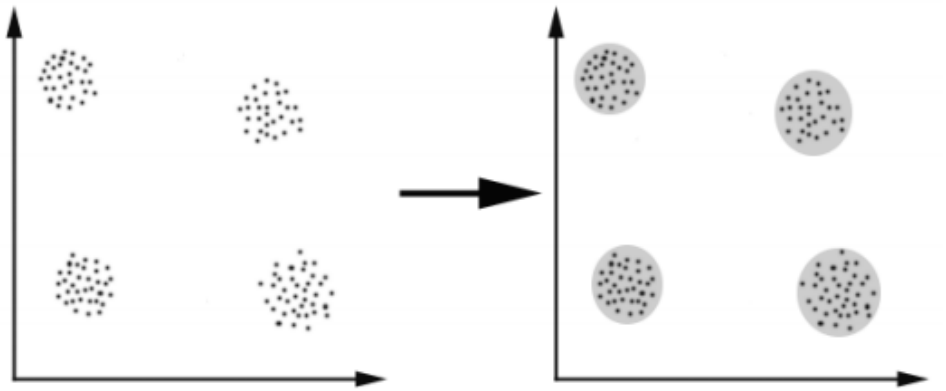

Figure 1.3: Example showing 4 cluster of data

role in data mining applications such as information retrieval, scientific data exploration, text mining, web analysis, spatial database applications, CRM, marketing, medical diagnostics, computational biology, and a number of other fields. Clustering can be done using different criteria to distribute data efficiently. The similarity criterion is distance in which two or more objects will belong to the same cluster if they are at a proximity distance according to a given distance (euclidean distance).This is known as distance-based clustering. Another one is conceptual clustering in which two or more objects belong to the same cluster if there exists a concept common to all the objects. In other words, objects are grouped if they fit some descriptive concepts, not according to some distance similarity measures.

Chapter 2

Literature Survey

In this chapter different methods of classification and clustering that are in use at present have been discussed briefly.

2.1 Nearest Neighbor Classifier

Nearest Neighbor Classifier is a method of supervised learning in which a test set with unknown class label is given and the goal is find its class. To make a prediction for a test set, its distance is computed from all the training set example, and the class of the nearest training set is assigned to the test set.

Algorithm:-

- Let Training set,$T = \{x_i\}_{i=1}^{n}$
- Let x be a pattern with unknown class label
- Find a pattern x in T such that $d(x, x'') < d(x, x_i)$ where $x \in T - x_i$

- Class label of x = Class label of x''

2.2 k-Nearest Neighbor Classifier

This algorithm computes the distance between test set and all the training sets, and a list of k nearest training set examples is maintained. The class to which maximum number of k-nearest training set belongs to, is assigned to test set.

Algorithm:-

- Let Training set,$T = \{x_i\}_{i=1}^{n}$
- Let x be a pattern with unknown class label
- $KNN = \emptyset$
- For each t=T
  if $|KNN| < K$
  $KNN = KNN \cup T$
  else
  find a pattern $x' \in KNN$ such that
  $dist(x, x') > dist(x, t)$
- $KNN = KNN - x'$
- $KNN = KNN \cup \{t\}$

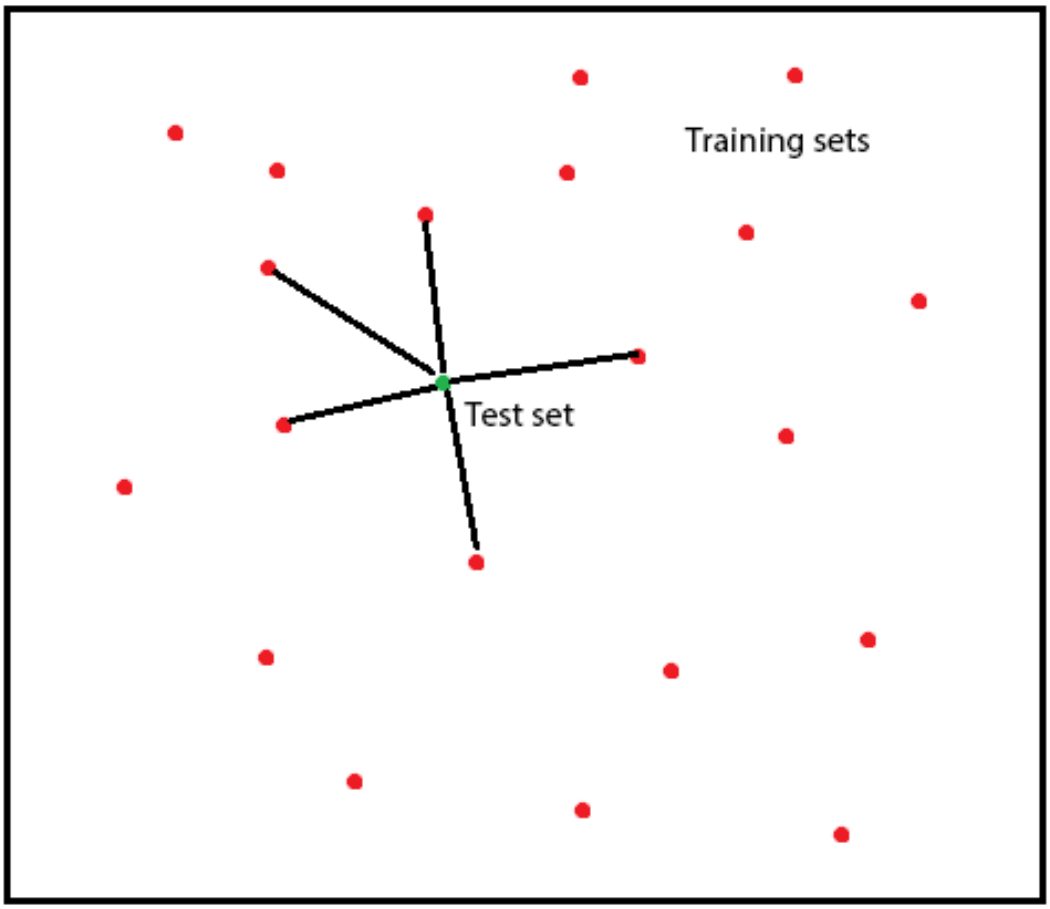

Figure 2.1: Diagrammatic representation of KNN

2.3 Condensed Nearest Neighbor

The concept of Condensed Nearest Neighbor rule was introduced by Hart in order to determine a training-set consistent subset of the original sample set. The algorithm uses two sets, called S and T. Initially, set S is initialized with $\emptyset$. Then, one pass through T is performed. During the scan, whenever a point of T is misclassified using the content of S, it is transferred from T to S. The algorithm terminates when no point is transferred during a complete pass of T.

Algorithm

- *Input*: Training set T
- *Output*: Condensed set S
- Start with empty condensed set S=$\emptyset$.
- For each x $\in$ T,
  Classify x using NN considering S as training set.
- If x is misclassified, then $S = S \cup x$
- Repeat above step until no change is found.

2.4 Modified Condensed Nearest Neighbor

The MCNN rule computes a training-set-consistent subset in an incremental manner. The consistent sub-

set S is initialized with the set of centroids of training set T. During each main iteration of the algorithm, first, set G with elements misclassified by using S as training set are selected. Then a representative set R is found containing one pattern of each class from G. Set S is augmented with the set R. The process is repeated until there are no misclassified points of T by using S.

Algorithm

- *Input*: Training set T
- *Output*: Condensed set S
- Start with condensed set S=centroids(T).
- G=$\emptyset$
- For each x $\in$ T
  Classify x using NN considering S as training set.
- If x is misclassified, then $G = G \cup x$
- Find a representative set R, containing pattern of each class, from G.
- $S = S \cup R$
- $G = \emptyset$
- Repeat above step until no change is found.

2.5 Leader's algorithm

Leaders method finds a partition of dataset like most other clustering methods.Leaders method works as follows: Initially the leader set L is empty and is built incrementally. For given threshold value Γ , foreach pattern x in the dataset if there is no leader l $\in$ L whose distance from x is less than Γ , we transfer x from training set T to leader set L. In this way the leader set is generated.

Algorithm

- Initialize L with first element of training set T.
- for each $t_i \in T$
- for each $l_j \in L$
  compute d_{ij}
  find $dmin = min < dij >$
- if $dmin > \Gamma$
  $L = L \cup \{t\}$
  else
  continue

2.6 Weighted K-nearest Leader Classifier

This algorithm was proposed by V. Suresh Babu and P. Viswanath which is an extension of leaders method.

The algorithm works as follows:
Let L_i be the set of leaders obtained after eliminating the noisy leaders for class ω_i, for i = 1 to c. Let L be the set of all leaders i.e. L = $L1 \cup ... \cup Lc$. For a given query pattern q, the k nearest leaders from L is obtained. For each class of leaders among these k leaders, their respective cumulative weight is found. Let this for class w_i be W_i, for i = 1 to c. The classifier chooses the class according to $argmax_{w_i} \{W_1 * P(w_1), W_2 * P(w_2), ..., W_c * P(w_c)\}$. If $P(w_i)$ is not explicitly given then it is taken to be n_i/n where n_i is the number of training patterns from class w_i and n is the total number of training patterns.

Algorithm

- Find the k nearest leaders of q from L
- Among the k nearest leaders find the cumulative weight of leaders that belongs to each class. Let this be W_i for w_i, for i= 1 to c.
- Class label assigned for q= $argmax_{w_i} \{W_1 * P(w_1), W_2 * P(w_2), ..., W_c * P(w_c)\}$.

Chapter 3

Motivation and Objective

3.1 Motivation

Nearest neighbor classier (NNC) and its variants like k-nearest neighbor classier (k-NNC) are popular because of:-

- simplicity
- good performance

Limitations:-

- Store the entire training set.
- Search the entire training set in order to classify a given pattern.
- Presence of noisy patterns degrade the performance of classifier.

Various remedies for the above mentioned problem:-

- Reduce the training set size by some editing techniques.

- Use only a few selected prototypes from the training set.
- Reduce the effect of noisy patterns.

3.2 Objective

To present a classification algorithm with less classification time and without degrading the performance as compared to K-nearest neighbor.

Chapter 4

Proposed Work

A faster algorithm which needs less classification time as compared to Nearest Neighbor Classifier and KNN has been proposed.

4.1 Method

- First a set of leaders is found with the concept of leader clustering using a pre-defined minimum distance between two leaders known as Γ
- After finding leaders set, rest of the remaining training set is classified into follower set, with a group of followers assigned to each leader.
- Now the k-nearest leaders is found out of the obtained leader set by finding the distance of a given test set from the leader set.
- And finally using these k-nearest leaders and their corresponding followers, class of the dataset is found.

4.2 Algorithm

4.2.1 Finding Leader Set

- Initialize L with first element of training set T.
- For each $t_i \in T$
- For each $l_j \in L$
  compute d_{ij}
  find $dmin = min < dij >$
- if $dmin > \Gamma$
  $L = L \cup \{t\}$
  else
  continue

4.2.2 Finding Leader Follower Set

- For each $t_i \in T$
- For each $l_j \in L$
  compute d_{ij}
  find $dmin = min < dij >$
- if $dmin <= \Gamma$
  t=follower(l)
  else
  continue

4.2.3 Find k-nearest leader set with respect to test set x

- KNL=∅
- For each t=T
- if $|KNL| <= K$
  $KNL = KNL \cup \{t\}$
  else
  find a pattern $x' \in KNL$ such that
  $dist(x, x') > dist\,(x, t)$

4.2.4 Find condensed set with respect to test set x

- Let C be the condensed set which is used to find class label of test set
- Let x be the test set with unknown class label
- Let K=number of k-nearest leader
- Let f(i)= number of followers in ith K-nearest leader
- for i=1 to K
  for j=1 to f(i)
  if $dist(x, i.leader) > dist(x, j.follower)$
  C=C ∪ j.follower
- Find the class to which maximum elements in this condensed set belong.
- Assign that class to test set x.

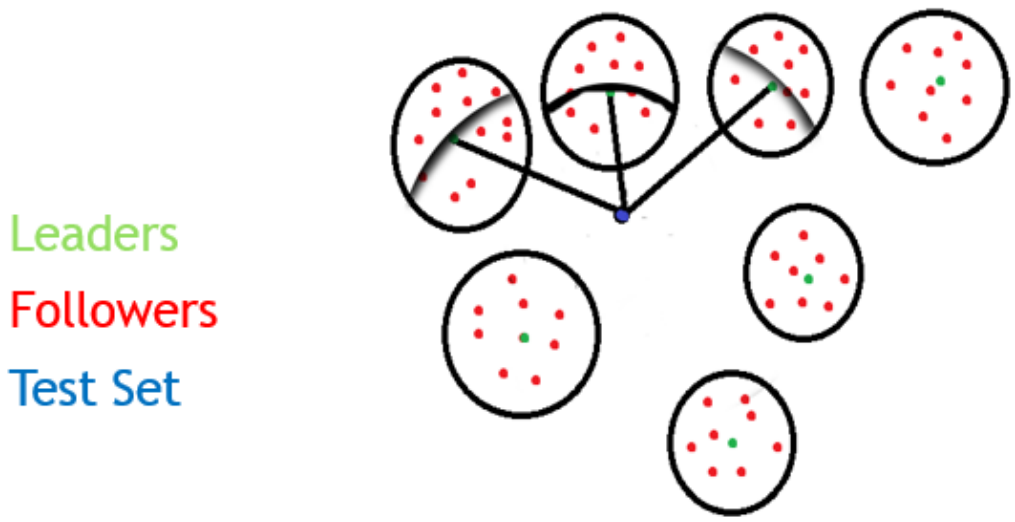

Figure 4.1: Diagrammatic Representation of KNLF

Chapter 5

Result and Analysis

In this chapter an analysis of classification time and accuracy is made on three classifiers viz. K-Nearest Neighbor classifier, Weighted K-Nearest Leader classifier, and proposed algorithm K-Nearest Leader Follwer classifier, using 5 different datasets

5.1 Datasets used

5.1.1 Breast Cancer dataset

Number of Instances: 699
Number of Attributes: 10
`http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original)`

5.1.2 Segment Challenge dataset

Number of Instance:2310
Number of Attributes:19
`http://archive.ics.uci.edu/ml/datasets/Image+`

Segmentation

5.1.3 Iris dataset

Number of Instance:150
Number of Attributes:4
http://archive.ics.uci.edu/ml/datasets/Iris

5.1.4 Wine dataset

Number of Instance:178
Number of Attributes:13
http://archive.ics.uci.edu/ml/datasets/Wine

5.1.5 E.coli dataset

Number of Instance:336
Number of Attributes:8
http://archive.ics.uci.edu/ml/datasets/Ecoli

5.2 Comparison Analysis

5.2.1 Classification Time comparison:-

Dataset	KNN	WKNL	KNLF
Breast Cancer	3.3 s	0.25 s	0.65 s
Segment Challenge	16.93 s	3.2 s	4.27 s
Iris	0.41 s	0.05 s	0.08 s
Wine	0.71 s	0.07 s	0.14 s
E.Coli	2.12 s	0.26 s	0.38 s

Table 5.1: Classification time for different classifiers using different datasets

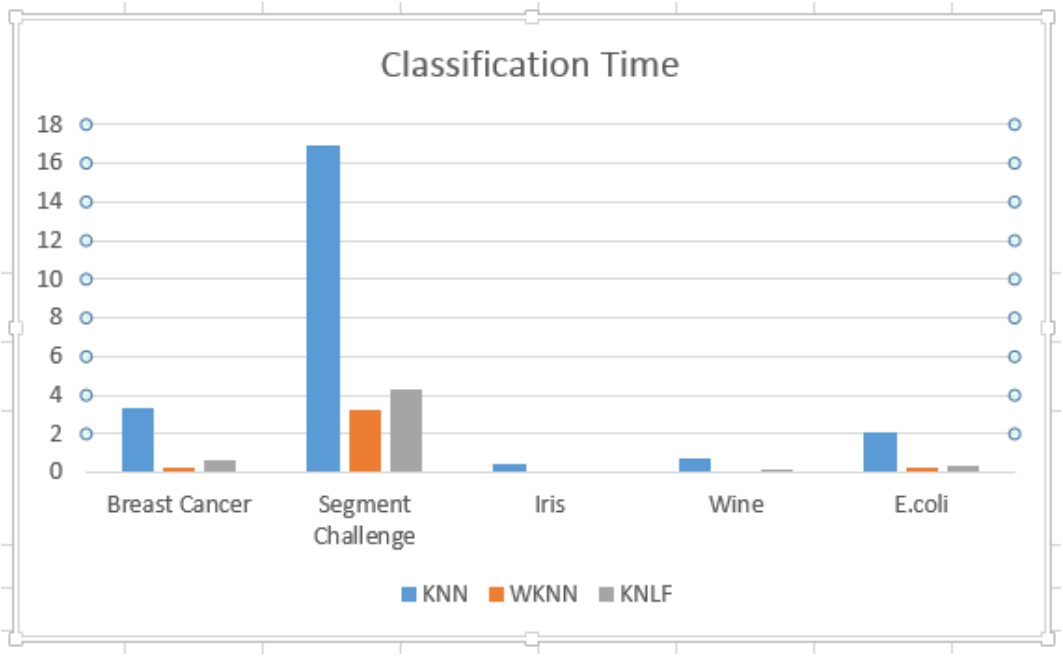

Figure 5.1: Classification time for different classfiers

5.2.2 Classification Accuracy comparison:-

Dataset	KNN	WKNL	KNLF
Breast Cancer	96	82	94
Segment Challenge	91.73	87.6	90.95
Iris	96.96	87.87	96.96
Wine	57.44	31.91	53.19
E.Coli	81.08	58.22	74.68

Table 5.2: Classification accuracy for different classifiers in percentage

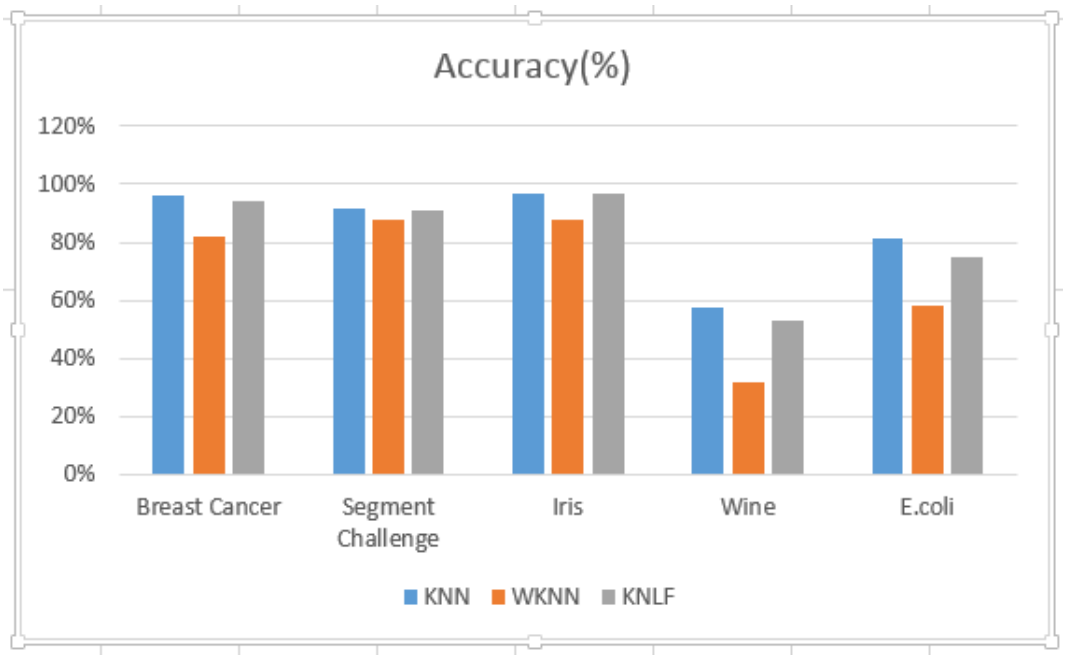

Figure 5.2: Classification accuracy for different classfiers

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this project, we observed that using K-Nearest Leader Follower Classifier,there is a marked difference in classification time as compared KNN classifier. Besides that the performance of KNLF classifier is on par with K-Nearest Neighbor and better than Weighted k-Nearest Leader, which proves that proposed K-Nearest Leader Follower classifier is a suitable one to be used in different data mining applications.

6.2 Future Work

Various condensation methods like Condensed Nearest Neighbor, MCNN, FCNN,etc, can be used before this method for removing outliers, which will further reduce the classification time.

Chapter 7

References

1. T.M. Cover and P.E. Hart, Nearest Neighbor Pattern Classica-tion,IEEE Trans. Information Theory,vol. 13, no. 1, pp. 21-27, 1967
2. Vijaya, P., Murty, M.N., Subramanian, D.K.: Leaders-subleaders: An ecient hierarchical clustering algorithm for large data sets. Pattern Recognition Letters 25, 505513 (2004)
3. V. Suresh Babu and P. Viswanath. Weighted k-nearest leader classier for large data sets. In PReMI, pages 1724, 2007.
4. P.E. Hart, The Condensed Nearest Neighbor Rule,IEEE Trans. Information Theory,vol. 14, no. 3, pp. 515-516, 1968.
5. F Angiulli, Fast Condensed Nearest Neighbor, ACM International Conference Proceedings, Vol 119, pp 25-32