A Medoid Based Weighting Scheme for Qualitative Improvement of Nearest Neighbor Decision Rule

Avideep Mukherjee

Department of Computer Science
Ramakrishna Mission Vivekananda Educational and Research Institute
Belur Math, Howrah
Pin - 711 202, West Bengal

A thesis submitted to
Ramakrishna Mission Vivekananda Educational and Research Institute
in partial fulfillment of the requirements for the degree of
M.Sc. in Computer Science
2018

to *Didu*

Acknowledgements

It is a genuine pleasure to express my deep sense of thanks and gratitute to my mentor, philosopher and guide Dr. Tanmay Basu. His dedication and keen interest above all his overwhelming attitude to help his students had been solely and mainly responsible for completing my work. His timely and scholarly advice, meticulous scrutiny and scientific approach have helped me to a great extent in accomplishing this task.

I would like to thank my parents, *Maa* and *Baba* who made me what I am by resting their unflinching faith in me at every step of my life.

I owe a deep sense of gratitude to Swami Dhyangamyananda, (Swathy Prabhu Maharaj), Head of the Department of Computer Science. His prompt inspirations, timely suggestions with kindness, enthusiasm and dynamism have enabled me to complete my thesis.

I thank all my friends with extreme gratitude for their constant support and encouragement, specially Anurag, Arnab, Arindam da, Ritam, Sayanta, Kalyani,Sourav, Trisha, Soham, Ashmita, Shibalik, Pratyay, Pralay and Sarbajit da.

I wish to extend my thankfulness to all the faculty members, present and former, of the Department of Computer Science of Ramakrishna Mission Vivekananda Educational and Research Institute for supporting me in numerous ways during the course of this thesis.

I would like to take this opportunity to thank my English Teacher, Mr. Arun Ain whose encouragement has motivated me to pursue a career in academics.

Ramakrishna Mission Vivekananda Educational
and Research Institute, Belur Math, West Bengal

Avideep Mukherjee

June 8, 2018

CERTIFICATE FROM THE SUPERVISOR

This is to certify that the thesis entitled *A Medoid Based Weighting Scheme for Qualitative Improvement of Nearest Neighbor Decision Rule* submitted by *Mr. Avideep Mukherjee*, who has been registered for the award of M.Sc. in Computer Science degree of Ramakrishna Mission Vivekananda Educational and Research Institute, Belur Math, Howrah, West Bengal is absolutely based upon his own work under the supervision of *Dr. Tanmay Basu* of Department of Computer Science, Ramakrishna Mission Vivekananda Educational and Research Institute and that neither his thesis nor any part of the thesis has been submitted for any degree/diploma or any other academic award anywhere before.

Dr. Tanmay Basu
Assistant Professor
Department of Computer Science
Ramakrishna Mission Vivekananda Educational and Research Institute
Belur Math, Howrah, 711 202, West Bengal

Abstract

The k-nearest neighbor decision rule is a simple, robust and widely used classifier. However, there are certain limitations of the nearest neighbor decision rule. First off, for a certain dataset determining the value of a valid k is difficult. Moreover, a slight change in the value of k leads to different decisions on the category of the given test data point. Nearest neighbor method classifies a test point even if the difference between the number of members between two competing classes is one. Moreover, nearest neighbor decision rule puts more stress on the data points that lie on the boundary region of individual classes. These methods rely upon those boundary points to decide the class label of a new data point, but, the boundary points may not be a good representation of a particular class. There are some particular cases where it can be shown that the boundary points fail to determine the actual category of a given test data point.

In this thesis, a method is proposed in spirit of the nearest neighbor decision rule to overcome some of their limitations using a medoid based weighting scheme. The proposed weighting scheme considers the samples in the training set that do not lie on the boundary region of individual classes of the data set. The data points that not only lie close to the test data point but also lies close to the medoid of its corresponding class are given more weightage, unlike the standard weighting scheme of nearest neighbor algorithms that put more weightage to the points that are just close to the test data point. The proposed classifier is evaluated and compared with other standard nearest neighbor classification techniques and some state of the art classifiers using some well known benchmark data sets over UCI and different text collections. The experimental results have shown that the proposed method outperforms the state of the art classifiers in terms of fmeasure.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The k-nearest neighbor (kNN) decision rule is a simple, robust and widely used classifier [5,7]. The kNN decision rule finds the k nearest neighbors from the training set by using a distance function and assigns a new test data point to a particular class by taking a majority vote among the k nearest neighbors [7]. The neighborhood parameter k of the kNN decision rule has high impact on its performance and hence choice of k is crucial [5, 8]. The cross validation technique is generally used to estimate an optimal value of k, but choosing a valid k is still a difficult job [3]. Moreover, there is no bound on the majority voting between the competing classes i.e., following the kNN decision rule one may put a test point into a class which has a win by one vote to the next competing category. This decision may not be intuitively satisfactory, if the test point belongs to the intersection region between the competing classes, where one may not always necessarily be interested in classifying every point.

A slight change in the value of k also leads to different assignment of class labels to the test data point. For example, consider a two-class classification problem. Let the class labels are A and B. Let there are 8 samples in the training set. Let d_t be a test data point. According to kNN algorithm the training data points are arranged according to increasing order of distances from d_t. Let the class labels of the arranged training data points are A, A, B, B, A, B, B, A. Now it can be seen that for $k = 5$, d_t gets classified to class label A whereas for $k = 7$, class label B is assigned to d_t. It is clear from the above demonstration that simple majority voting principle may not be an appropriate way to classify the test data points. Basically, when there is more or less same representation from the competing classes among the nearest neighbors,it is preferable to keep the test data point unclassified rather than making a wrong judgment [4]. Hence, the value of k is very crucial in determining the class of the test data sample. Moreover, we cannot determine whether k should be 5 or 7. Therefore identifying a suitable value of k for a certain dataset is not clearly defined in the kNN classification technique.

The other widely used variant of kNN decision rule is distance weighted kNN decision rule [9]. The method assigns different weights to different k nearest neighbors based on their distances with the test data point, where the closer neighbors get higher weights. Let $\vec{x}_1, \vec{x}_2, ..., \vec{x}_k$ are the k nearest neighbors of a test data point, say, $\vec{x}_t$. Let the corresponding distances of these neighbors from $\vec{x}_t$ is denoted by $\rho(\vec{x}_j, \vec{x}_t); \forall j = 1, 2, ...k$ where ρ is the distance function. The weight w_j associated with the j^{th} nearest neighbor $\vec{x}_j$ is defined as

$$w_j = \begin{cases} \frac{\rho(\vec{x}_k,\vec{x}_t)-\rho(\vec{x}_j,\vec{x}_t)}{\rho(\vec{x}_k,\vec{x}_t)-\rho(\vec{x}_1,\vec{x}_t)} & \text{if } \rho(\vec{x}_k, \vec{x}_t) \neq \rho(\vec{x}_1, \vec{x}_t) \\ 1 & \text{otherwise} \end{cases}$$

The test data point $\vec{x}_t$ is assigned to the class for which the sum of the weights of the representative data points of the class among these k nearest neighbors is maximum [9]. The major limitation of this method is that it also suffers from the influence of neighborhood parameter k. Different values of k leads to different assignments of class labels to the test data point.

It may be noted that kNN and weighted kNN decision rule put more stress on the data points that lie on the boundary region of individual classes. These methods rely upon those boundary points to decide the class label of the test data point. However, boundary points may not be a good representation of a particular class. The weighted kNN decision rule assigns more weights to the nearer points that are close to the boundary region of a class rather than the farther points for a test data point. A method is thus desirable to overcome these limitations of the kNN decision rules.

The proposed method finds the medoid of each class in the data set and subsequently finds the distance of a test data point to all the points that lie in between the medoid of individual classes and the test point. These data points constitute the neighborhood of the test data point. The weight of a point in that neighborhood is computed by considering the distance of that point from the medoid and also from the test data point. Thereafter the first β (say) neighbors are considered and the weights of data points belonging to the individual classes are aggregated. The test data point is assigned to a particular class that has the maximum aggregated weight and the difference between the aggregated weights of the competing classes is greater than a given threshold (say γ). Thus the proposed method can overcome the influence of the boundary points. The method continues until the condition is not satisfied or the method has considered all the the members of the neighborhood, but can not take a decision. The performance of the proposed method has been compared with the standard nearest neighbor techniques and other state of the art classifiers using some standard data sets from UCI Machine Learning Repository. The algorithm is also tested in the domain of text classification using standard text corpora from Text Retrieval Conferences and Newspapers [4]. The empirical results has shown that the proposed technique outperforms the state of the arts for majority of the datasets.

The thesis is organized in the following manner. Chapter 2 provides a literature survey of various nearest neighbor techniques that are proposed as an improvement over kNN decision rule. The advantages and disadvantages of those methods are also discussed. Chapter 3 demonstrates, in details, the proposed classification technique that is also derived from the nearest neighbor decision rule. The proposed medoid based weighting scheme is also described there. The datasets that are used to evaluate the proposed method and other state of the classifiers are discussed in Chapter 4. The experimental setup and feature-extraction of text data, along with the results of the proposed technique and other classifiers on the UCI and TREC datasets are also explained there. Chapter 5 deals with various statistical tests and representations of the results obtained on the datsets. These representations help to demonstrate the robustness and consistency of the proposed technique in comparison to other classifiers on the considered datasets. Finally, the general comments and future scope of this work has been discussed in Chapter 6.

Chapter 2

Literature Review

The nearest neighbor (NN) rule identifies the category of unknown data point on the basis of its nearest neighbor whose class is already known. Cover and Hart [7] introduced the kNN Technique in which the concerned test data point (d_t) is assigned to the class which has the maximum number of representative training data points among the k nearest neighbors of d_t. It's simplicity is its main advantage. However, it carries some disadvantages too, like memory requirements and computational complexity. To cope up with these limitations, many methods have been developed namely weighted kNN,Adaptive Nearest Neighbor, Model based kNN Condensed NN, Reduced NN, Generalized NN etc [5].

The condensed nearest neighbor technique (CNN) proposed by Gowda and Krishna [12] eliminates similar or redundant datasets which do not add extra information and thus are unnecessary. Although it reduces the memory requirements and recognition rate while improving query time, it still poses the problem of computational cost. The Reduced Nearest Neighbor (RNN) [11] does an extra job over CNN by removing the patterns which are independent of the training dataset results. The advantages and disadvantages of RNN are same as that of CNN.

Another technique, namely the Model Based kNN (MkNN) [13] creates a 'similarity matrix' using the similarity measures from the given training set. After that for each category, largest local neighbor is found which surrounds large number of neighbors and a data point is located with largest global neighborhood. These steps are repeated until all data tuples are grouped. Once data is formed using model, kNN is executed to specify category of unknown sample. MkNN has advantages like it yields better classification accuracy and it also selects the value of k automatically. Although it does not consider marginal data points outside a class region, MkNN can be used in dynamic web mining for large repositories.

Subash C. Bagui and Sikha Bagui [1] betters the kNN by assigning ranks to the training data for each category. Rank based kNN is quite effective in case of data with huge variations between features. Also, RkNN depends on the distribution of the data. In Modified kNN [18], an alteration of wkNN, validity of all data samples in the training data set is computed and accordingly weights are assigned and then validity and weights both together take part in deciding which class should the test data point be assigned to. This method partially overcomes the low accuracy of the weighted kNN decision rule, but still suffers from the cost of computation.

Yong Zeng, Yupu Zeng and Liang Zhou [26] define the novel idea of pseudo-neighbor to classify sample data point. Pseudo neighbor is not the actual nearest neighbor but a new nearest neighbor selected determined by the weighted sum of distances of kNN of unclassified data points in each class. Then Euclidean distance is evaluated and pseudo neighbor with greater weight is found and classified for unknown sample. In the technique purposed by Zhou Yong [26], clustering is used to calculate nearest neighbor. The steps include, first of all removing the samples which are lying near to the border, from training set. Cluster each training set by k value clustering and all cluster centers form new training set. Assign weight to each cluster according to number of training samples each cluster have [17].

Chapter 3

A Medoid Based Nearest Neighbor Decision Rule

3.1 Proposed Method

In this work, a medoid based weighting scheme is proposed to overcome the influence of the boundary points on nearest neighbor decision rule. A *medoid* of a class is the data point of that class that has least distance with the mean vector of the class. Let $\vec{x}_1, \vec{x}_2, ..., \vec{x_n}$ be a set of n points in a space with a distance function ρ. Medoid is defined as

$$\vec{x}_{medoid} = \text{argmin}_{\vec{x} \in \{\vec{x_1}, \vec{x_2}, ..., \vec{x_n}\}} \sum_{i=1}^{n} \rho(\vec{x}, \vec{x}_i)$$

Let us consider a multi class classification problem with K classes, $C_1, C_2, \cdots, C_K$. Let the training sample set contains n samples, $\vec{x}_1, \vec{x}_2, ..., \vec{x}_n$, with m features. Let $\vec{x}_t$ is the concerned test data point.

At the beginning of the algorithm, the corresponding medoids of the individual classes are computed. The effective neighborhood, S, of a particular test data point $\vec{x}_t$ consists of the data points from the training sample set whose distance from the test data point is less than or equal to the distance between the medoid of its corresponding class and the test data point. S is then rearranged in increasing order of distances between the test data point and individual members of S. The method starts with the first L data points of S and stores them in S_L. The initial values of L is to be predefined and it is denoted as β in Algorithm 1. Thereafter, the weights of each data point $x_j \in S_L$ is computed based on their distance from test data point and the medoid of its corresponding class. The weighting scheme is explained in details in Section 3.2. The weight of a class, C_i is computed by adding the weights of the data points that belong to that particular class and is stored in L_i. The weight of a class, C_i is therefore given as

$$L_i = \sum_{d_j \in C_i} w_j \quad \forall i = 1, 2, \cdots, K$$

The maximum and the second maximum class weights are obtained from the set $\{L_1, L_2, \cdots, L_K\}$. Let they be called L_{max_1} and L_{max_2} respectively. These weights are then divided by the total number of representative points of the respective classes i.e. $|C_{max_1}|$ and $|C_{max_2}|$ respectively to get normalized scores. Now, according to weighted kNN algorithm, the test data point is assigned to the class that has the maximum weight among the $L_i's$. The proposed decision rule assigns the test data point to the best class, when the weights of the best class and its competing class is differed by a predefined threshold, γ, i.e. if $\frac{L_{max_1}}{|C_{max_1}|} - \frac{L_{max_2}}{|C_{max_2}|} \geq \gamma$. If this criterion is not satisfied then the value of L is incremented by 1 and the weight of the next data point in S is computed. The method is repeated until the aforesaid condition is satisfied or the method has traversed all the members of S. If the method still can not satisfy the given criterion after exhausting all the training data points in S, then the test data point, d_t is kept unclassified.

Note that $\beta = 1$ implies one neighbor and thus the minimum value of β is 2. The method only selects those training data points in S which lie between the test data point and its corresponding medoid. Therefore the value of β can be at most $|S|$. Hence, the range of β is between 2 and $|S|$. The difference between the weights of competing classes is at least 0 and thus $\gamma \geq 0$. As the class weights are normalized between 0 and 1, the maximum value of γ can not be greater than 1. Hence, the value of γ lies between 0 and 1, both inclusive.

3.2 Proposed Weighting Scheme

The philosophy behind the weighting scheme is the concept of *similarity*. The notion of *similarity* is obtained from *cosine similarity*, a measure broadly used in text classification. The cosine distance function is computed as 1 − cosine similarity. Therefore for any distance function ρ the corresponding similarity measure can be

Algorithm 1: Proposed Nearest Neighbor Decision Rule Using Medoid Based Weighting Scheme

```
   Input: $X = \{\vec{x}_1, \vec{x}_2, ..., \vec{x}_n\}$ be the training set and $n = |X|$. $\vec{x}_t$ is a test data
          point.
   $\beta$ is the initialization parameter on neighborhood.
   $\gamma$ is a threshold on class weights.
   $C = \{C_1, C_2, ..., C_m\}$ be the set of $m$ classes.
   $\rho$ be the distance function and $\psi$ is the scaling function that normalizes $\rho$
   between 0 and 1.
   Output: $y_t$ be the class label of $\vec{x}_t$.
   Steps:
1  Find the median of each class, say $\vec{C}'_i$, $\forall i = 1, 2, ...m$
2  $S = \varnothing$
3  $\forall \vec{x}_j \in X$ and $\vec{x}_j \in C_k$ do $S = S \bigcup \vec{x}_j$ if $\rho(\vec{x}_j, \vec{x}_t) \leq \rho(\vec{C}'_k, \vec{x}_t)$
4  Rearrange $S$ in increasing order of distance with respect to $\vec{x}_t$
5  $L = \beta$
6  $S_L$ = First $L$ data points from $S$
7  $\forall \vec{x}_j \in S_L$ do $w_j = \{1 - \psi(\rho(\vec{x}_j, \vec{x}_t))\}\{1 - \psi(\rho(\vec{C}'_k, \vec{x}_j))\}$, where $\vec{x}_j \in C_k$
8  $L_i = \sum_{\vec{x}_j \in C_i} w_j, \ \forall i = 1, ..., m$
9  $L_{max_1} \leftarrow \max\{L_1, L_2, ..., L_m\}$
10 $L_{max_2} \leftarrow \max\{\{L_1, L_2, ..., L_m\} - \{L_{max_1}\}\}$
11 if $\frac{L_{max_1}}{|C_{max_1}|} - \frac{L_{max_2}}{|C_{max_2}|} \geq \gamma$ then $y_t = c_{max_1}$ and return $y_t$;
12 else
13     $L = L + 1$
14     if $L \geq |S|$ then return $\vec{x}_t$ as unclassified;
15     else
16         $w_L = \{1 - \psi(\rho(\vec{x}_L, \vec{x}_t))\}\{1 - \psi(\rho(\vec{C}'_k, \vec{x}_L))\}$, where $\vec{x}_L \in C_k$
17         $S_L = S_L \bigcup \vec{x}_L$
18         go to step 8
```

generated as $1-\rho$. This expression is not entirely correct. In case of cosine similarity, the value of the similarity measure lies between $[-1,1]$. This means the similarity value can be at most 1. Therefore, its corresponding distance measure can be rightly computed by subtracting it from the maximum value, i.e. 1. And also, the similarity measure for cosine can be easily generated from the distance value by again subtracting it from 1. Similarly, in case of other distance functions the value of distance may be greater than 1 and often we cannot fix a maximum value for every distance function, for example, say *Euclidean Distance Function.*Therefore we need to scale down the distance values with some scaling function, say ψ in order to normalize it between 0 and 1. Now the normalized distance value can be rightly subtracted from 1 to get the similarity value. The weight of a particular data point $\vec{x}_j$ is now defined as

$$w_j = \{1 - \psi(\rho(\vec{x}_j, \vec{x}_t))\} \times \{1 - \psi(\rho(\vec{C}'_k, \vec{x}_j))\}$$

such that $\vec{x}_j \in C_k$ and C'_k is the medoid of class C_k.
This weighting scheme shows that the data points which are not only close to the test data point but also at the same time close to it's corresponding medoid will be given greater value of weights. Thus, the point being close to the Neighbor (may be a boundary point) is not good enough. It's medoid should also be close to the test data point, which proves that the particular data point is not actually a boundary point.

Chapter 4

Experimental Evaluation

4.1 Description of Data

Eight standard datasets from UCI Machine Learning Repository[1] [4] have been used here for experimental analysis. All of these datasets have different number of samples and it varies from 32 to 4601. The range of attributes of these data sets lies between 4 and 57 and the number of classes varies from 2 to 11. The description of the data sets are presented in Table 4.1. The Pima Indian diabetes dataset has been removed from UCI recently. The dataset used here is collected from Kaggle[2] [23].

Table 4.1: Overview of UCI Benchmark Datasets

Dataset	Domain	#Instances	#Attributes	#Categories
Ecoli	Life	336	7	8
Breast Cancer	Life	569	30	2
Ionosphere	Physical	351	33	2
Spambase	Text	4601	57	2
Vowel	NA	990	14	11
Pima Indian Diabetes	Life	768	8	2
Glass	Physical	214	9	6
Lungs Cancer	Life	32	54	3

The proposed method is also evaluated using 5 text corpora that are taken from the Text REtrieval Conference, co-sponsored by the National Institute of Standards and Technology (NIST) and U.S. Department of Defense[3] [3]. All the text datasets

[1] http://archive.ics.uci.edu/ml

[2] https://www.kaggle.com/uciml/pima-indians-diabetes-database

[3] https://trec.nist.gov/

are developed by Karypis and Han [15]. These text corpora consists of documents as less as 204 to at most 878, and has number of terms ranging from 5804 to 8261. All the text corpora pose a multi class classification problem with number of classes varying from 6 to 10. The detailed overview of the text corpora are provided in Table 4.2.

Table 4.2: Overview of Text Corpora from TREC

Dataset	#Documents	#Terms	#Categories
tr45	690	8261	10
tr41	878	7454	10
tr11	414	6429	9
tr23	204	5832	6
tr12	303	5804	8

4.2 Evaluation Techniques

The performance of the proposed method and the state of the art classifiers are evaluated by using the standard precision, recall and f-measure [25]. The precision and recall for two class classification problem can be computed as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$$

Here TP stands for *true positive* and it counts the number of data points correctly predicted to the positive class. FP stands for *false positive* and it counts the number of data points that actually belong to the negative class, but predicted as positive (i.e., *falsely predicted as positive*). FN stands for *false negative* and it counts the number of data points that actually belong to the positive class, but predicted as negative (i.e., *falsely predicted as negative*). TN stands for *true negative* and it counts the number of data points correctly predicted to the negative class. The f-measure combines recall and precision with an equal weight in the following form:

$$\text{F-measure} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}$$

The closer the values of precision and recall, the higher is the f-measure [2]. F-measure becomes 1 when the values of precision and recall are 1 and it becomes 0 when precision is 0, or recall is 0, or both are 0. Thus f-measure lies between 0 and 1. A high f-measure value is desirable for good classification.

There are two conventional methods to generalize these evaluation functions for multi class classification problem, namely *macro-averaging* and *micro-averaging* [16]. The macro averaged measure finds the precision and recall score for each class from the confusion matrix and then the these scores for all the classes are averaged [25]. The micro averaged measure individually aggregates the true positives, false positives and false negatives over all the classes and then finds the precision and recall [25]. We have used both macro-averaged and micro-averaged f-measure to evaluate the performance of the classifiers.

4.3 Experimental Setup

The performance of the proposed method is compared with support vector machine (SVM) [6], kNN [7] and weighted kNN [9] decision rules. It may be noted that SVM has been widely used for data classification in various applications [24]. The concept of the proposed method has been introduced in spirit of nearest neighbor decision rule and therefore the performance of the proposed method is compared with kNN decision rule and weighted kNN classifier.

The data sets used here have no specific training and test sets. Therefore we have randomly split the data sets into two parts - 80% is considered as training set and the rest as test set. The random split is done in such a way that ensures the representative of each class in both training and test set. The training set is used to train the classifiers and the test set is used to evaluate the performance of individual classifiers.

The proposed algorithm has two major parameters, the first one is β, which is used to initialize the neighborhood of the test data point and the other one is γ, which is used as the bound on the proposed weights of the competing classes. It may be noted that $\beta \in [2, 3, \cdots, |S|]$, where S is the set of selected data points from the training set, which is being used by the proposed weighting scheme. The value of γ is to be fixed experimentally by using 10-fold cross validation technique on the training set.

The parameters of the state of the art classifiers are tuned using 10-fold cross validation on the training set. In case of kNN and weighted kNN classifiers, the value of k is chosen by varying it from 2 to 40. The state of the art classifiers are implemented using scikit-learn[4] [19], a machine learning tool in Python.

4.3.1 Document Representation Using Vector Space Model

Vector Space Model is the most common representation of Documents with respect to supervised and unsupervised learning of text data [22]. According to this model, a document (d_i) is represented as a vector $\vec{d_i}$ where, $\vec{d_i} = \{tf_{i1}, tf_{i2}, \ldots, tf_{in}\}$ and $tf_{ij}, \forall j = 1, 2, \ldots, n$ is the frequency of the j^{th} term t_{ij} in document d_i [3]. The number of documents, from a set of N documents, in which the j_{th} term occurs is termed as the document frequency (df_j). The *term frequency* (tf) and the *inverse document frequency* (idf) of a particular term are associated together to generate the weight of that particular term. The inverse document frequency (idf) of a term indicates how often a term appears in a collection of documents. Mathematically, it is defined as

$$idf_j = \log(\frac{N}{df_j}), \quad \forall j = 1, 2, \ldots, n$$

As long as Nearest Neighbor Techniques are concerned, these documents need to compared with each other with respect to some distance function. In the domain of text mining, the concept of *similarity* is used between two documents $\vec{d_i}$ and $\vec{d_j}$. Among the different similarity measures available in literature, the most common measure used is cosine similarity [14], which is expressed as

$$\cos(\vec{d_1}, \vec{d_2}) = \frac{\vec{d_1} \cdot \vec{d_2}}{|\vec{d_1}||\vec{d_2}|} = \frac{\sum_{j=1}^{n}(w_{1j} \times w_{2j})}{\sqrt{\sum_{j=1}^{n} {w_{1j}}^2 \times \sum_{j=1}^{n} {w_{2j}}^2}}$$

where $\vec{d_i} = (w_{i1}, w_{i2}, \ldots, w_{in})$; $i = 1, 2$.
$\cos(\vec{d_1}, \vec{d_2}) = 1$ implies that the two documents are totally similar. As the value of cosine decreases from 1 to 0, so does the similarity between two documents.

[4] http://www.scikit-learn.org

4.4 Analysis of Results Using UCI Benchmark Datasets

The performance of the proposed method and state of the art classifiers on the benchmark data sets mentioned above are shown in Table 4.3 and Table 4.4 using both macro-averaged and micro-averaged fmeasure. Moreover, the value of the parameter k that has been used to perform kNN and weighted kNN are shown in Table 4.3 and Table 4.4 beside individual f-measure values. The values of L, the number of nearest neighbors and γ are also presented in Table 4.3 and Table 4.4.
It can be seen from Table 4.3 and Table 4.4 that the proposed method performs better than the other classifiers for all the data sets except Ionosphere using both macro-averaged and micro-averaged fmeasures. For the Ionosphere data set the other classifiers have an edge over the proposed method using both macro-averaged and micro-averaged fmeasures. It can be seen from Table 4.3 and Table 4.4 that there are 48 comparisons for the proposed method and the proposed one has performed better than the other methods in 42 cases. The statistical significance of these results is to be tested. For example, for Ecoli the macro-averaged fmeasure of kNN is 0.85 and for the proposed method it is 0.86 and we have to test whether this difference is statistically significant.

Table 4.3: Macro Averaged Fmeasure of the proposed classifier and state-of-the-art classifiers on various benchmark datasets

Dataset	SVM	k	wkNN	k	kNN	L(avg)	γ	Proposed
Ecoli	0.67	3	0.68	6	0.85	6	1	**0.86**
Breast Cancer	**0.96**	4	0.94	8	0.94	4	0.9	**0.96**
Ionosphere	**0.92**	2	0.90	2	0.88	3	0.05	0.87
Spambase	0.84	3	0.81	6	0.79	5	0.8	**0.85**
Vowel	0.15	2	0.26	2	0.28	3	0.05	**0.33**
Pima Indian Diabetes	**0.78**	5	0.70	31	0.73	6	0.9	**0.78**
Glass	0.67	2	0.7	3	0.55	2	0.025	**0.76**
Lungs Cancer	0.33	8	0.3	3	0.48	5	0.05	**0.52**

Table 4.4: Micro Averaged Fmeasure of the proposed classifier and state-of-the-art classifiers on various benchmark datasets

Dataset	SVM	k	wkNN	k	kNN	L(avg)	γ	Proposed
Ecoli	0.82	3	0.86	6	0.85	6	1	**0.87**
Breast Cancer	0.96	4	0.94	8	0.94	4	0.9	**0.97**
Ionosphere	**0.92**	2	0.91	2	0.90	3	0.05	0.89
Spambase	**0.85**	3	0.82	6	0.80	5	0.8	**0.85**
Vowel	0.20	2	0.28	2	0.31	3	0.05	**0.39**
Pima Indian Diabetes	0.79	5	0.73	31	0.77	6	0.9	**0.81**
Glass	0.63	2	**0.72**	3	0.63	2	0.025	**0.72**
Lungs Cancer	0.43	8	0.29	3	0.43	5	0.05	**0.52**

4.5 Analysis of Results Using Standard Text Collections

The performance of the proposed method and state of the art classifiers on the different text corpora are shown in Table 4.5 and Table 4.6 using both macro-averaged and micro-averaged fmeasure. The text data are transformed into feature vectors using the Vector Space Model as described in Section 4.3.1. Here also, the value of the parameter k that has been used to perform kNN and weighted kNN are shown in Table 4.5 and Table 4.6 beside individual f-measure values. The values of L, the number of nearest neighbors and γ are also presented in Table 4.5 and Table 4.6.

Table 4.5: Macro Averaged Fmeasure of the proposed classifier and state-of-the-art classifiers on various text corpora

Dataset	SVM	k	wkNN	k	kNN	L(avg)	γ	Proposed
tr45	0.86	4	0.87	3	0.62	2	0.025	**0.90**
tr41	**0.95**	3	0.84	7	0.82	4	0.1	0.94
tr11	0.78	3	0.75	5	0.74	5	0.05	**0.81**
tr23	0.88	4	0.91	6	0.83	7	0.1	**0.92**
tr12	0.85	5	0.71	2	0.63	6	0.025	**0.86**

It can be seen from Table 4.5 and Table 4.6 that the proposed method performs better than the other classifiers for all the data sets except tr41 using both macro-averaged and micro-averaged fmeasures. For the tr41 data set almost all the other classifiers outperform the proposed method using both macro-averaged and micro-

Table 4.6: Micro Averaged Fmeasure of the proposed classifier and state-of-the-art classifiers on various text corpora

Dataset	SVM	k	w*k*NN	k	*k*NN	L(avg)	γ	Proposed
tr45	**0.93**	3	0.91	6	0.87	6	1	**0.93**
tr41	**0.96**	4	0.95	8	0.91	4	0.9	0.95
tr11	0.90	2	0.87	2	0.87	3	0.05	**0.92**
tr23	0.87	3	0.90	6	0.85	5	0.8	**0.91**
tr12	0.85	2	0.82	2	0.84	3	0.05	**0.85**

averaged fmeasures. It can be seen from Table 4.5 and Table 4.6 that there are 30 comparisons for the proposed method and the proposed one has performed equally or better than the other methods in 28 cases. The statistical significance of these results is to be tested. For example, for tr12 the macro-averaged fmeasure of SVM is 0.85 and for the proposed method it is 0.86 and we have to test whether this difference is statistically significant.

Chapter 5

Discussion

A paired *t-test* is suitable for testing the equality of means when the variances are unknown. A suitable test statistic is described and tabled in [21] and [20] respectively. The statistic uses the null hypothesis of equal means assuming unequal variance on same sample size. The statistic t is measured as $t = \frac{\mu_1 - \mu_2}{\sqrt{\sigma_1^2/n_1 + \sigma_2^2/n_2}}$, where μ_1, μ_2 are the means, σ_1, σ_2 are the standard deviations and n_1, n_2 are the number of observations [21]. It has been found that the results are statically significant in 30 out of 42 cases, where the proposed technique performs better than the other methods for the level of significance 0.05. The test results are statistically significant in 3 out of 6 cases for the same level of significance, when other methods have an edge over the proposed one. Thus in 90.90% cases the performance of the proposed technique is significantly better than the other classifiers. The effectiveness of the proposed method can be observed from these results.

The robustness of different classification algorithms can be determined by using the idea of Friedman [10]. Robustness of a classifier h for a particular data set is defined as $E_h = E_h/E_0$, where E_h is either macro averaged or micro averaged fmeasure of h and $E_0 = \max_h E_h$. The best classifier for a particular data set will have $E_h = 1$, while the other competing algorithms will have $E_h \leq 1$. Lower values of E_h indicate the lack of robustness of the algorithm h. We have computed this ratio for all the classifiers and for all the data sets using macro averaged and micro averaged fmeasure and they are graphically shown by boxplots respectively in Figure 5.1 and Figure 5.4. It can be observed from these figures that the proposed method outperforms the competing classifiers.

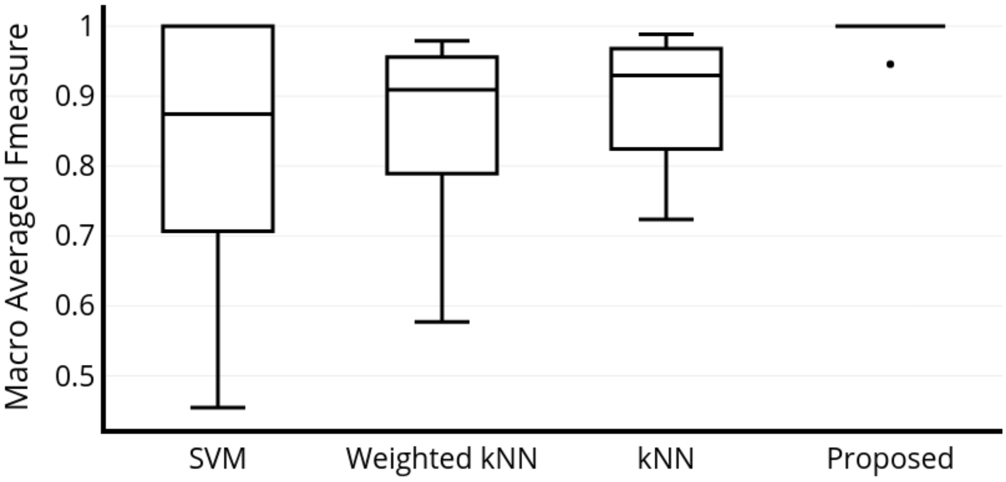

Figure 5.1: Robustness of Different Classifiers Using Macro Averaged Fmeasure on UCI Datasets

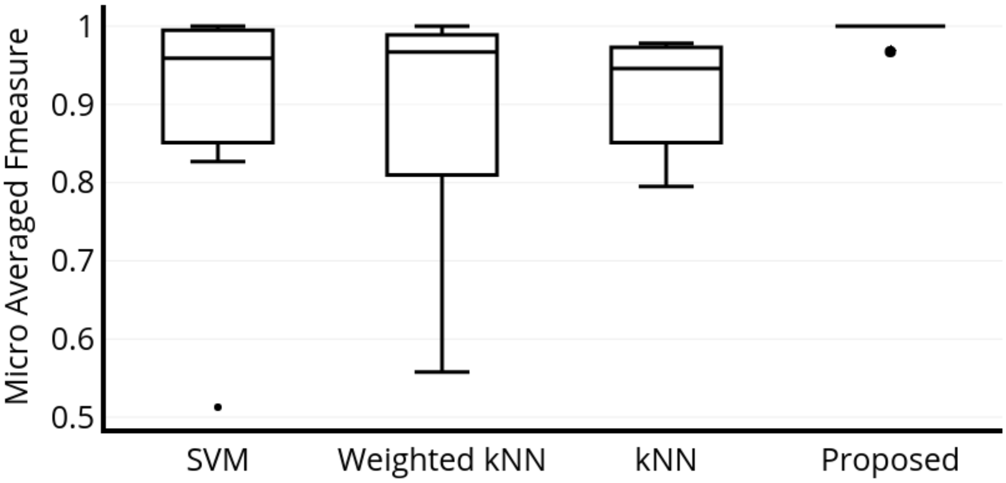

Figure 5.2: Robustness of Different Classifiers Using Micro Averaged Fmeasure on UCI Datasets

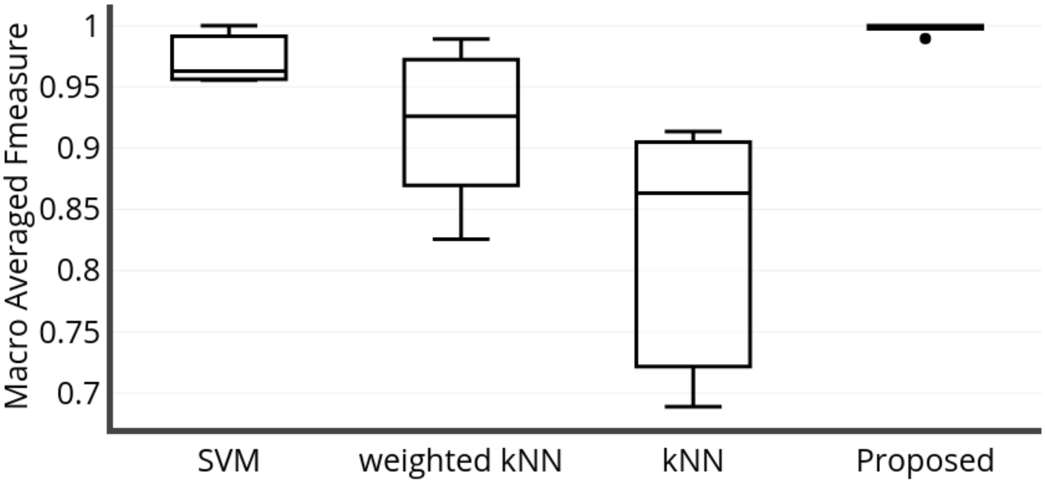

Figure 5.3: Robustness of Different Classifiers Using Macro Averaged Fmeasure on various text collections

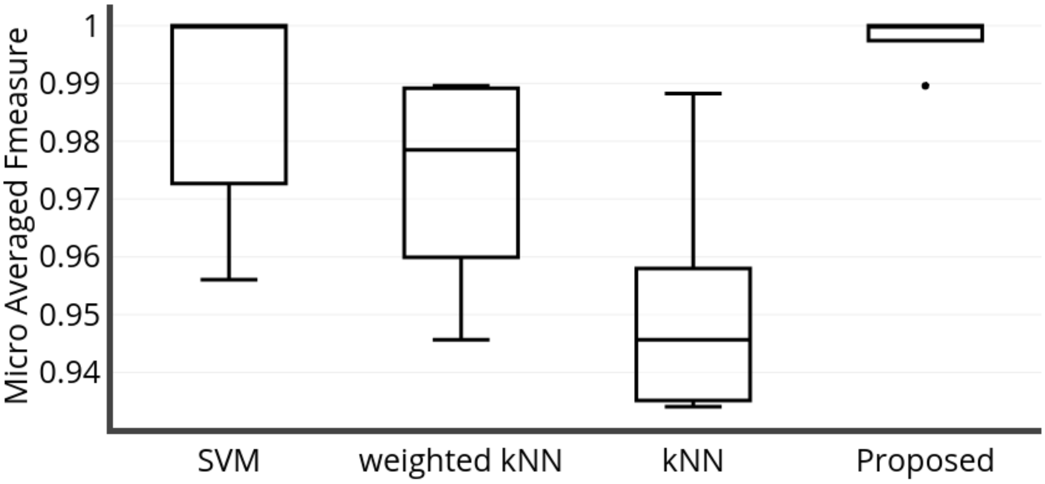

Figure 5.4: Robustness of Different Classifiers Using Micro Averaged Fmeasure on various text collections

Chapter 6

Conclusions and Scope of Further Research

The kNN method is one of the most fundamental and simple classification methods for statistical pattern recognition. A method has been introduced in this article to overcome some of the limitations of the state of the art nearest neighbor decision rules. The performance of the method is tested on different standard benchmark data sets collected from UCI repository and other text collections from TREC. The method uses a parameter γ to provide a bound the difference between the weights of the competing classes. It has been shown in the empirical analysis that the method outperforms the state of the art techniques in most of the cases. It has been observed from the experimental results that no data point remain unclassified by the proposed technique for all the data sets used here. This proves the effectiveness of the method. However, in future, we have to test the performance of the proposed method in different other applications e.g., image classification, face recognition etc.

Bibliography

[1] Subhash C Bagui, Sikha Bagui, Kuhu Pal, and Nikhil R Pal. Breast cancer detection using rank nearest neighbor classification rules. *Pattern recognition*, 36(1):25–34, 2003.

[2] T. Basu and C. A. Murthy. A feature selection method for improved document classification. In *Proceedings of the International Conference on Advanced Data Mining and Applications*, pages 296–305, 2012.

[3] T. Basu and C.A. Murthy. Towards enriching the quality of k-nearest neighbor rule for document classification. *International Journal of Machine Learning and Cybernetics*, 5(6):897–905, 2014.

[4] T. Basu, C.A. Murthy, and H. Chakraborty. A tweak on k-nearest neighbor decision rule. In *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition, IPCV'13*, pages 929–935, Las Vegas, USA, 2012.

[5] Nitin Bhatia et al. Survey of nearest neighbor techniques. *arXiv preprint arXiv:1007.0085*, 2010.

[6] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[7] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[8] B. V. Dasarathy. *Nearest Neighbor NN Norms: NN Pattern Classification Techniques.* McGraw-Hill Computer Science Series. IEEE CS Press, 1991.

[9] Sahibsingh A Dudani. The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, (4):325–327, 1976.

[10] Jerome H Friedman et al. Flexible metric nearest neighbor classification. Technical report, Technical report, Department of Statistics, Stanford University, 1994.

[11] Geoffrey Gates. The reduced nearest neighbor rule (corresp.). *IEEE transactions on information theory*, 18(3):431–433, 1972.

[12] K Gowda and G Krishna. The condensed nearest neighbor rule using the concept of mutual nearest neighborhood (corresp.). *IEEE Transactions on Information Theory*, 25(4):488–490, 1979.

[13] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. Knn model-based approach in classification. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 986–996. Springer, 2003.

[14] Eui-Hong Han and George Karypis. Fast supervised dimensionality reduction algorithm with applications to document categorization retrieval. In *CIKM*, 2000.

[15] George Karypis and Eui-Hong Sam Han. Fast supervised dimensionality reduction algorithm with applications to document categorization & retrieval. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 12–19. ACM, 2000.

[16] C. D. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval.* Cambridge University Press, New York, 2008.

[17] C Chandra Mouli, P Jyothi, and K Nagabhushan Raju. Comparative study of supervised classification algorithms for wosras. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (An ISO 3297: 2007 Certified Organization) Vol*, 3:7188–7193, 2014.

[18] Hamid Parvin, Hosein Alizadeh, and Behrouz Minaei-Bidgoli. Mknn: Modified k-nearest neighbor. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 1. Citeseer, 2008.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos,

D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[20] C. R. Rao, S. K. Mitra, A. Matthai, and K. G. Ramamurthy, editors. *Formulae and Tables for Statistical Work*. Statistical Publishing Society, Calcutta, 1966.

[21] Graeme D Ruxton. The unequal variance t-test is an underused alternative to student's t-test and the mann-whitney u test. *Behavioral Ecology*, 17(4):688–690, 2006.

[22] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

[23] Everhart J.E. Dickson W.C. Knowler W.C. Johannes R.S. Smith, J.W. Using the adap learning algorithm to forecast the onset of diabetes mellitus. *In Proceedings of the Symposium on Computer Applications and Medical Care*, pages 261–265, 1988.

[24] I. H. Witten, E. Frank, and M. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, third edition, 2011.

[25] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval, Kluwer Academic Publishers*, 1(1-2):69–90, 1999.

[26] Zhou Yong, Li Youwen, and Xia Shixiong. An improved knn text classification algorithm based on clustering. *Journal of computers*, 4(3):230–237, 2009.