2014 Fourth International Conference on Advanced Computing & Communication Technologies

# An Enhanced K-Nearest Neighbor Algorithm Using Information Gain and Clustering

Shweta Taneja[#1], Charu Gupta[#2], Kratika Goyal[#3] , Dharna Gureja [#4]
[#]CSE Department, Bhagwan Parshuram Institute of Technology,
Guru Gobind Singh Indraprastha University, New Delhi, India
[1]shweta_taneja08@yahoo.co.in
[2]charugupta0202@gmail.com
[3]goyalkratika@ymail.com
[4]dgrockdg@gmail.com

***Abstract*—KNN (k-nearest neighbor) is an extensively used classification algorithm owing to its simplicity, ease of implementation and effectiveness. It is one of the top ten data mining algorithms, has been widely applied in various fields. KNN has few shortcomings affecting its accuracy of classification. It has large memory requirements as well as high time complexity. Several techniques have been proposed to improve these shortcomings in literature. In this paper, we have first reviewed some improvements made in KNN algorithm. Then, we have proposed our novel improved algorithm. It is a combination of dynamic selected, attribute weighted and distance weighted techniques. We have experimentally tested our proposed algorithm in NetBeans IDE, using a standard UCI dataset-Iris. The accuracy of our algorithm is improved with a blend of classification and clustering techniques. Experimental results have proved that our proposed algorithm performs better than conventional KNN algorithm.**

***Keywords- KNN; Dynamic KNN (DKNN); Distance-Weighted KNN (DWKNN); Weight Adjusted KNN; Information Gain***

## I. INTRODUCTION

Data mining is a process of extracting knowledge from large amounts of data stored either in databases or other information repositories. It is an essential process where intelligent methods are applied to extract data patterns which can be viewed from different angles. It is extensively applied in various fields like data analysis, market analysis, fraud detection, sports etc. The other related terms for data mining are knowledge mining from databases, knowledge extraction, data or pattern analysis, data dredging. Classification [1] is a data mining technique to predict class of unknown instances. A training set with their known class labels is used to predict class of future or unknown data.

KNN [6] is a classification algorithm first proposed by T.M. Cover and P.E. Hart. It is frequently used to classify future data due to its simplicity, ease of implementation and effectiveness. It is one of the top ten data mining an algorithm which has been widely applied in various fields of pattern recognition, cancer diagnosis, text classification etc. KNN is a lazy learning or instance-based method. The classifier or model is not built during training period which leads to high computational time and cost during classification period. Only training tuples with their class labels are stored at training time and classifier is build at the classification time for each test instance. It is a non-parametric method for classifying unknown data i.e. it does not make any assumptions on underlying data distribution. An instance is represented by attribute vector $<a_1,a_2,\ldots.. a_n>$, where $a_i$ denotes the value of ith attribute $A_i$ of x.

KNN uses standard Euclidean distance [4] to measure the difference or similarity between training and test instance. The standard Euclidean distance $d(x_i,x_j)$ is defined in equation 1.1 as follows:

$$d(x_i, x_j) = \sqrt{\sum ( a_r (x_i) - a_r (x_j) )^2} \quad (eq1.1)$$

KNN takes into account the most common class of k nearest neighbors to estimate the class of test instance [2]. It is defined in equation 1.2 as follows:

$$c(x) = \arg\max_{c \in C} \sum_{i=1}^{k} \delta(c , c(y_i)) \quad (eq\ 1.2)$$

where $y_1 , y_2 , \ldots, y_k$ are the k nearest neighbors of test instance, k is the number of the neighbors, C represents the finite set of class labels and $\delta(c, c(y_i)) = 1$ if $c = c(y_i)$ and $\delta(c, c(y_i)) = 0$ otherwise.

We have conducted an extensive survey of the work done in KNN algorithm. We have seen that there are three main shortcomings [2] of KNN. They are as follows: 1) The distance function used is Standard Euclidean distance which considers equal participation of all the attributes; 2) Neighborhood size is taken as input parameter which results in inaccurate results in unbalanced data; 3) The class probability estimation is based on simple voting method.

To overcome these shortcomings, we have seen various methods to improve its accuracy of classifying test instance:1) More accurate distance functions can be used which take into account priority of attributes; 2) Find best neighborhood size which produces more accurate results ; 3) More accurate class probability estimation method can be used instead of simple voting.

With the combination of these three methods, we have proposed an efficient algorithm using dynamic selected, attribute weighted and distance weighted techniques. We have experimentally tested our algorithm using standard UCI dataset-Iris [15].

978-1-4799-4910-6/14 $31.00 © 2014 IEEE
DOI 10.1109/ACCT.2014.22

The rest of the paper is organized as follows. In section II, we present related work done in the improvements made in KNN algorithm. In section III, we describe our proposed improved KNN algorithm. In section IV, we present experiments conducted and results obtained. Finally we conclude in the next section.

## II. RELATED WORK

Many improved algorithms have been proposed by various authors to overcome shortcomings in conventional KNN algorithm. Some of them are described below.

### A. Dynamic KNN

The selection of the value of k is an essential part of KNN. In practical data sets, some classes have more data points (called majority class) than other classes (called minority class). If k value is a fixed, user- defined one then in most of the cases the result would be biased towards the majority class. To avoid this biasness, many researchers have proposed different algorithms to optimize the value of k.

One of the methods is trying various values of k and selecting the best out of it. Based on this idea, Selective Neighborhood Naive Bayes (SNNB) [2] was proposed. In this method for a test instance, various k values are tested and local naive Bayes is learned for each k value. Then the most accurate classifier is used for classification of test instance. This method is very time consuming to be used in real world applications.

Other effective approach to learn best k value at training time is Dynamic KNN (DKNN) [3]. It is based on leave-one-out cross-validation method, a combination of eager and lazy learning method. This method has been implemented in Weka [2].

### B. Weight Adjusted KNN

KNN uses standard Euclidean distance to measure the difference or similarity between training and test instance. It takes into account equal participation of all attributes of the instance whether relevant or not. Hence, when there are large numbers of irrelevant attributes, the value of distance function become inaccurate and is known as Curse of dimensionality [13]. An effective approach to overcome this problem is to assign degree of importance to all attributes i.e. to weight each attribute differently for calculating distance between two instances.

Weighted Euclidean distance function [5] can be defined in equation 2.1 as:

$$d(x_i, x_j) = \sqrt{\sum w_i \left( a_r(x_i) - a_r(x_j) \right)^2} \qquad \text{(eq 2.1)}$$

where $w_i$ ( i =1,2,.....,n) is the weight of attribute $A_i$

Thus when attributes are nominal, the attribute-weighted distance function [4] can be defined in equation 2.2 as:

$$c(x) = \sum_{i=1}^{n} I_p(A_i : C)\ \delta(a_i(x), a_i(y)) \qquad \text{(eq 2.2)}$$

where $I_p$ ( $A_i$ ; C) is the mutual information of the attribute variable $A_i$ and the class variable C, and $\delta(a_i(x), a_i(y)) = 0$ if $a_i(x) = a_i(y)$ and $\delta(a_i(x), a_i(y)) = 1$ otherwise.

### C. Distance-weighted KNN

KNN uses simple voting method for estimating class of the test instance. It is very sensitive to unbalanced data. To improve this, an improved method is to weight vote of k nearest neighbors differently according to their distance from test instance. It is known as k-nearest neighbor with distance weighted (KNNDW) [13].

Hence, weighted class probability estimation method [3] is given in equation 2.3 as follows:

$$c(x) = \arg\max_{c \in C} \sum_{i=1}^{k} w\, \delta(c, c(y_i)) \qquad \text{(eq 2.3)}$$

where $y_1, y_2, \ldots, y_k$ are the k nearest neighbors of test instance, k is the number of the neighbors, C represents the finite set of class labels and $\delta(c, c(y_i)) = 1$ if $c = c(y_i)$ and $\delta(c, c(y_i)) = 0$ otherwise.

## III. PROPOSED ALGORITHM

We have studied different modifications of KNN algorithm and have proposed a novel algorithm. This algorithm is expected to reduce the inefficiency of traditional K nearest neighbor algorithm.

The proposed algorithm is divided in two parts:

- Data pre-processing: The data pre-processing part provides the weight to different attributes, determines *k* value for the test samples and also divides the training data set into different clusters. It results in creation of model used to classify future data. This part runs only once, so it would not affect the efficiency.
- Classification: The actual classification of test data is done in this part. This part runs every time when it does classification.

We have designed a flow chart to explain these two parts of our algorithm as shown in Figure. 1 and 2 respectively.

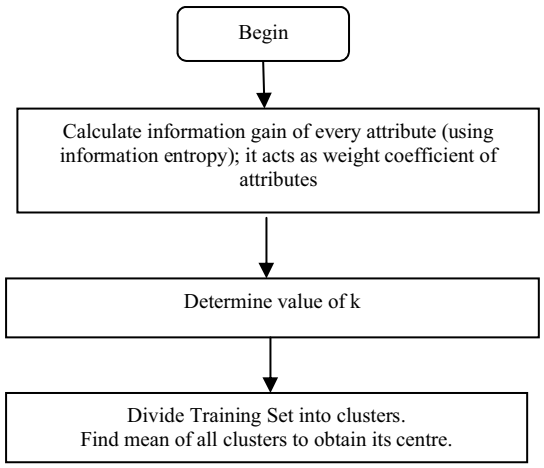


Figure 1. Data Preprocessing

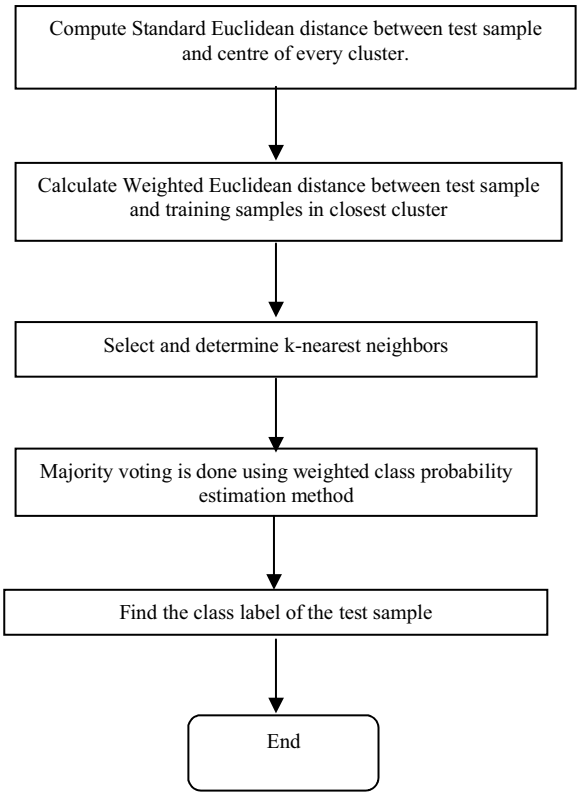


Figure 2. Classification Process

The enhanced K-NN Algorithm can be explained in the following steps as follows:

Step 1: Calculate the information entropy of each attribute which is used to compute information gain of every attribute. It acts as the weight of attributes to assign priorities to them.

Step 2: Find the k value for the training set.

Step 3: Divide training set into number of clusters.

Step 4: Find mean of all the clusters to obtain center of every cluster.

Step 5: Determine cluster closest to test sample by using Euclidean Distance formula.

Step 6: Weighted Euclidean Distance formula is used to calculate the distance between the test sample and each sample in cluster. Hence, k nearest neighbors are determined.

Step 7: The class label of selected k neighbors with maximum probability is chosen as class label of test sample.

Our proposed KNN algorithm is better than the traditional KNN algorithm. This proposed algorithm improves the accuracy of classification and greatly reduces the execution time as it is a blend of classification and clustering techniques.

## IV. Experimental Analysis And Results

To test the performance of our proposed KNN, we have used it to classify the Iris data set taken from the UCI Machine Learning Repository [15]. This data set is comprised of 150 instances. Each instance is characterized by 4 attributes-sepal length, sepal width, and petal length and petal width and classified as either "Iris-setosa" or "Iris-versicolor" or "Iris-virginica" classes. We select 141 instances randomly as the training instances to calculate the attribute weight sets and then classify the rest instances in the set.

All the experiments are performed on an Intel Core 2 Duo processor, 2.66GHz PC with 2.00GB RAM and NetBeans IDE 7.3.1 and Oracle Database 10g Express Edition.

Our proposed KNN algorithm is based on weighted entropy of attribute value where information gain of every attribute is calculated and Euclidean distance formula is modified.

Firstly, we calculate the entropy of every attribute. An instance is represented by attribute vector $<a_1,a_2,\ldots.. a_n>$, where $a_i$ denotes the value of ith attribute $A_i$ of x. Assume S is a set of training samples, and |S| is the number of training samples, then the information amount [12] of S is E(S).

$$E(S) = - \sum p_i \log_2 ( p_i )$$

$$\text{where } p_i = |C_i,S|/|S|$$

$p_i$ is the probability of an arbitrary tuple in S and it belongs to class $C_i$ .

Assuming that A is an attribute that has different values, the set of training samples can be divided into m sub-sets using these values. Then the conditional entropy [12] of categories divided by A can be obtained as E(S|A).

$$E(S|A) = \sum ( |S_j| /|S| ) * E(S_j)$$

The information gain of attributes is calculated using entropy. Information Gain is defined as difference between information of whole dataset and information contained in an attribute [1].

$$\text{Gain } (A) = E(S) - E (S|A) \geq 0$$

The selection of the k value is an essential part of KNN. Leave-one-out cross-validation method is used to determine k value. It considers single training tuple as validation data and rest tuples as training data. This process is repeated for every training tuple as validation data.

In traditional KNN algorithm, the distance between the test point and every training point needs to be calculated. This process consumes a lot of time [2]. Therefore, we divided the complete training set into different clusters having similar data points. Dividing the training set into different sub-sets reduces computation time greatly. Now the distance between the test point and the data point in sub-set needs to be computed. To determine in which cluster, the test point belongs to, cluster centre is obtained. Cluster centre is the mean of all the data points in a cluster.

Weighted Euclidean Distance [5] formula is used to calculate distance of test sample from each training sample in closest cluster. Weight of an attribute is the information gain of that attribute. The formula is as follows:

$$d(x_i, x_j) = \sqrt{\sum w_i \left( a_r(x_i) - a_r(x_j) \right)^2}$$

where $w_i$ is weight associated with every attribute of dataset.

Then distances obtained are sorted and k-nearest neighbors are determined. Then, weight the vote of k nearest neighbors differently according to their distance from test instance. Majority of class labels of selected cluster is the test sample's class label. Weight is inversely proportional to square of distance from test instance .

$$w = 1/(d)^2$$

Hence, weighted class probability estimation method [2] is as follows:

$$c(x) = \arg\max_{c \in C} \sum_{i=1}^{k} w\, \delta(c, c(y_i))$$

where $y_1, y_2, \ldots, y_k$ are the k nearest neighbors of test instance, k is the number of the neighbors, C represents the finite set of class labels and $\delta(c, c(y_i)) = 1$ if $c = c(y_i)$ and $\delta(c, c(y_i)) = 0$ otherwise.

Hence, majority of class labels of selected k-nearest neighbors is the test sample's class label. We have performed the experiment many times and shown comparison of performance between our proposed algorithm and traditional KNN in Table 1.

TABLE I
COMPARISON OF PROPOSED KNN ALGORITHM WITH CONVENTIONAL KNN ALGORITHM

| | **Proposed KNN** | **Conventional KNN** |
|---|---|---|
| **Time taken to build model** | 1 sec | 0 sec |
| **Time taken to classify a new data instance** | 1 sec | 2 sec |
| **Accuracy** | 25/25 | 25/25 |

Conventional KNN algorithm is a lazy learning method so it does not create a model before classifying test instance and delays its learning until classification time. Hence, to classify every test instance, it takes 2 seconds. Whereas, our proposed algorithm creates a model during data preprocessing period which reduces the actual classification time and thereby increases the efficiency of the algorithm. It takes only 1 sec to classify every data point. Therefore, our algorithm proves to be more efficient than conventional KNN algorithm.

For accuracy of classification, attributes are weighted to solve the problem of curse of dimensionality and distance is weighted as information gain of the attributes. Vote of k-nearest neighbors is weighted to improve the accuracy of classification of algorithm.

For time efficiency, the training set is preprocessed, mapped to different clusters and a suitable model for accurate classification is created. Hence, to classify the test samples, searching will be done in sub-space instead of entire training set, which greatly reduces the classification time.

## V. CONCLUSION AND FUTURE WORK

In this paper, we have shown the major shortcomings affecting the traditional KNN algorithm and reviewed some improvements made to overcome them. Based on the analysis, we present our proposed KNN algorithm using dynamic selected, attribute weighted and distance weighted techniques. This proposed algorithm improves the accuracy of classification and reduces the execution time. It is a blend of classification and clustering techniques.

We have experimentally tested our algorithm in NetBeans IDE, using a standard UCI dataset-Iris. Experimental results have proved that our proposed algorithm performs better than conventional KNN algorithm. In future, we will implement our proposed algorithm on various other standard UCI datasets and also incorporate soft computing techniques like Fuzzy logic to suit the real world scenarios.

## REFERENCES


[1] W. Baobao, M. Jinsheng, and S. Minru, "An Enhancement of K-Nearest Neighbor Algorithm Using Information Gain and Extension Relativity," Proc. International Conference on Condition Monitoring and Diagnosis (CMD 2008), Apr. 2008, pp. 1314-1317, doi:10.1109/CMD.2008.4580218.

[2] L. Jiang, Z. Cai, D. Wang, and S. Jiang, "Survey of Improving K-Nearest-Neighbor for Classification," Proc. 4rth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007), vol. 1, Aug. 2007, pp. 679-683, doi:10.1109/FSKD.2007.552.

[3] J. Wu, Z. Cai and Z. Gao, "Dynamic K-Nearest-Neighbor with Distance and Attribute Weighted for Classification," Proc. International Conference on Electronics and Information Engineering (ICEIE 2010), vol. 1, Aug. 2010, pp. V1-356 - V1-360, doi:10.1109/ICEIE.2010.5559858.

[4] S. Bo, D. Junping, and G. Tian, "Study on the Improvement of K-Nearest-Neighbor Algorithm," Proc. International Conference on Artificial Intelligence and Computational Intelligence (AICI 09), vol. 4, IEEE Computer Society, Nov. 2009, pp. 390-393, doi:10.1109/AICI.2009.312.

[5] S. Sun and R. Huan, "An Adaptive k-Nearest Neighbor Algorithm," Proc. 7th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2010), vol. 1, IEEE Press and IEEE Circuits and System Society, Aug. 2010, pp. 91-94, doi:10.1109/FSKD.2010.5569740.

[6] X. Xiao, and H. Ding, "Enhancement of K-nearest Neighbor Algorithm Based on Weighted Entropy of Attribute Value," Proc. 5th International Conference on BioMedical Engineering and Informatics (BMEI 2012), IEEE Press, Oct. 2012, pp. 1261-1264, doi:10.1109/BMEI.2012.6513101.

[7] H. Hong, G. Juan and W. Ben, "An Improved KNN Algorithm Based on Adaptive Cluster Distance Bounding for High Dimensional Indexing," Proc. 3rd Global Congress on Intelligent Systems (GCIS), Conference Publishing Services, Nov. 2012, pp. 213-217, doi:10.1109/GCIS.2012.86.

[8] X. Li and C. Xiang, "Correlation-based K-Nearest Neighbor Algorithm," Proc. IEEE 3rd International Conference of Software

Engineering and Service Science (ICSESS), June 2012, pp. 185-187, doi:10.1109/ICSESS.2012.6269436.

[9] J. Gou, L. Du, Y. Zhang and T. Xiong, "A New Distance-weighted k-nearest Neighbor Classifier," Proc. Journal of Information & Computational Science, June 2012, pp. 1429-1436.

[10] M. A. Amal and B. A. Riadh, "Survey of Nearest Neighbor Condensing Techniques," Proc. International Journal of Advanced Computer Science and Applications (IJACSA), vol. 2, No. 11, 2011, pp. 59-64.

[11] N. Bhatia and Vandana, "Survey of Nearest Neighbor Techniques," Proc. International Journal of Computer Science and Information Security (IJCSIS), vol. 8, 2010, pp. 302-305.

[12] J. Han and M. Kamber, Data Mining Concepts and Techniques 2nd ed., Elsevier, 2006.

[13] S. Taneja, C. Gupta, D. Gureja and K. Goyal, "K Nearest-Neighbor Techniques for Data Classification-AReview," Proc. International Conference on Computing, Informatic and Network (ICCIN-2K14), Jan. 2014, pp. 69-73.

[14] S. Chen and H. Hsiao, "A New Approach for Fuzzy Query Processing Based on Automatic Clustering Techniques," Information and Management Sciences, vol. 18, 2007, pp. 223-240.

[15] The UCI website [Online] http://archive.ics.uci.edu/ml/datasets/Iris