An Appraise of KNN to the Perfection

Pooja Rani
Student, Department of Computer Science and Engineering, GJU S&T, Hisar, India

Jyoti Vashishtha
Assistant Professor, Department of Computer Science and Engineering, GJU S&T, Hisar, India

ABSTRACT
K-Nearest Neighbor (KNN) is highly efficient classification algorithm due to its key features like: very easy to use, requires low training time, robust to noisy training data, easy to implement. However, it also has some shortcomings like high computational complexity, large memory requirement for large training datasets, curse of dimensionality and equal weights given to all attributes. Many researchers have suggested various advancements and improvements in KNN to overcome these shortcomings. This paper appraising various advancements and improvements in KNN.

Keywords
K-Nearest Neighbor, KNN, Distance weighted KNN, Attribute weighted KNN.

1. INTRODUCTION

1.1 Data mining
Every moment human being is generating and using a huge amount of data. This data can be present in the type of documents, graphics, texts, numbers, figures, audio or video, hypertext etc. This data is of no use if it doesn't provide any useful information for decision makers. Data mining can be defined as a method of extracting useful information from bulky amount of information and it is a powerful capability with great potential to help organizations spotlight on the most important information in their data warehouses. Data mining tools predict future trends and behaviors which help different organizations to make proactive knowledge-driven decisions [1]. It is used in numerous spheres like in drawing efforts from area including database technology, machine learning, statistics, pattern recognition, information retrieval, neural networks, knowledge-based systems, artificial intelligence, high-performance computing, and data visualization [2].It has various applications like in wholesale business, telecommunication area, criminal inquiry, economics, sales, medication, market, banking, healthcare and insurance, consumer segmentation and do research analysis [3], [1]. Throughout the years, several techniques like Classification, Regression, Clustering, Time Series Analysis, Prediction, Summarization, Sequence Discovery, Spatial Mining, Web mining, Association Rule Mining etc have been designed and developed to extract interesting and hidden knowledge from the various types of datasets[1], [4]. In data mining, classification is supervised learning in which target class is predefined and each record is assigned to one of the several predefined categories or classes [5]. Classification is a two-phase process, first is learning phase where a classification model is constructed and second is classification phase where the already constructed model is used to predict class labels for the given data[6]. Classification can be done using Decision tree induction, Bayesian networks, Rule-based classification, classification by Back-propagation, Support vector machines, by using Association rule mining etc. These all are called as eager learners because when a set of training tuples is given to them they build a classification model before getting new test tuples to categorize[2], [7]. Though, there are some classification algorithms which do not construct a model, but they make the classification judgment by comparing the test set with the training set all time when they act upon classification. These algorithms are known as instance-based learning algorithms or also lazy learners as they just store the instances and they do not do any effort on instances until a test tuple they are given. Lazy learners do a smaller amount work when a training tuple is offered and additional work while make a classification. The K-nearest-neighbor classifiers as well as case-based reasoning classifiers, both are the examples of lazy learners.

1.2 Traditional K-Nearest Neighbor
The K-Nearest-Neighbor (KNN) classifier first came into description in the early 1950s. KNN is applicable in many spheres such as, classification, regression, pattern recognition, mining of text, finance, agriculture, medicine etc.[8]. KNN does not require any prior knowledge about dataset because of its non-parametric nature and its own-self assumes that instances in the datasets are separately and identically spread, therefore, the instances which are more close to each other have the same category[5].As KNN is a lazy learner algorithm in its learning phase it only stores all the training tuples given as input without performing any calculations or does a little processing and it waits unless a test tuple is offered to it to classify. All the computations or processing apply at the time of classification only. It classifies the new tuple by comparing it with all the training tuples that are similar to it. When an unknown tuple is given, KNN searches the sample space for the 'k' nearest-neighbors or nearby tuples to the unknown tuple. The 'k' nearby training tuples to the given unknown tuple can be find out using a choice of distance metrics like Euclidean distance, Minkowski distance, Manhattan distance etc. The standard Euclidean distance is usually regarded as the distance function [9]. Euclidean distance function is given below which calculates the distance between two tuples *x* and *y*.

$$d(x,y) = \sqrt{\sum_{i=1}^{n} \left(a_i(x) - a_i(y)\right)^2}$$

Where 'n' represents the total number of attributes and 'a' is the value of that attribute in instances, here *x* is test tuple and *y* is a set of training tuples.

When it is given an unknown tuple, the KNN classifier searches the sample space for the 'k' training tuples that are nearby to the new tuple and assigns the most nearest

majority class in 'k' nearest neighbors of test tuple by using the formula[10]:

$$c(x) = \arg max \sum_{c \varepsilon Ci=1}^{k} \delta\big(c, c(y_i)\big)$$

Where y_i represents y_1, y_2 ….y_k and these are the 'k' nearest neighbors of x, 'k' represents the number of neighbors and $\delta\big(c, c(y_i)\big)$=1 if c= $c(y_i)$ and $\delta\big(c, c(y_i)\big)$=0 otherwise. The selection of 'k' is very significant in constructing the KNN model. The key factor of this model is that it can stoutly influence the quality of classification. For any given data, a small value of 'k' will direct to a large variation in predictions instead, adjusting 'k' to a large value may lead to a large model bias. As a result, 'k' should be adjusted to a value large adequate to minimize the probability of misclassification and small enough with respect to the number of tuples in the dataset so that the 'k' nearest point is close enough to the query point. When 'k'=1, the given unknown tuple is assigned the category of the training tuples that is closest to it in sample space. In binary where we have only two classes it is useful to adjust 'k' to be an odd number as this avoids ties between votes [8].The value of 'k' should not be chosen as a multiplier of the number of classes. It avoids ties when number of classes is greater than two. All algorithms have its own pros and cons. Although KNN is broadly used algorithm due to its straightforwardness, high efficiency, simplicity, less training time, very simple to comprehend, stout to noisy training data, efficient if the training data is large and its robustness. KNN has some pros like to decide the number of nearest neighbors or the value of 'k' parameter , distance metric is not clear which type of distance metric should be used; which attributes are better to include while producing the best results? Shall we include all attributes or certain attributes only?, it can be easily fooled by unrelated attributes, slow at query time, computational cost is quite high because it is needed to compute distances of each query instance to all training samples; the large memory to implement in amount with size of training set; low accuracy rate in multidimensional datasets etc.

2. ADVANCEMENT IN KNN

Traditional KNN is very simple, highly efficient, easy to implement, comprehensive, but it has some disadvantages like curse of dimensionality, selection of distance metric, computational cost, memory limitation, deciding the value of 'k' parameter, biasing towards majority class, equal impact of all attributes etc. To overcome these limitations and to improve the classification performance of KNN, some weighting techniques are developed. The traditional KNN with weighting strategy is known as weighted KNN. The weights can be provided to attributes, instances or both attributes as well as instances at same time. As we have discussed about the role of 'k' parameter value on its classification performance, some distance weighted techniques are developed to reduce the impact of 'k' value on its performance and it also reduces the effect of outliers present in training datasets. Distance weighted techniques also used to meet the problem of simple majority voting in its classification phase so that the majority class prediction , which does not always give better performance in case of imbalanced datasets, can be improved. Another problem in KNN is the curse of dimensionality and same impact of all attributes in the process of classification but data classification is dependent more upon some attributes than others, when those relevant attributes get more weight, then it is called attribute weighting and it is used to reduce the curse of dimensionality.

Attribute weighted, Distance weighted and the combination of these two techniques are discussed here which are used to enhance the performance of KNN classifier.

2.1 Attribute weighted KNN

To reduce the equal impact of all attributes and curse of dimensionality, weights are assigned to each attribute where more relevant attributes get more weight as compared to others. The traditional KNN with attribute weighting is known as Attribute weighted KNN. Some attribute weighted techniques are discussed below:

Jiang *et al* in 2006devised a Dynamic KNN Naive Bayes along with Attribute weighting technique to enhance the performance of KNN. Attributes are assigned weights using common information between each attribute and the class attribute. In this method eager learning and lazy learning are combined together to find out the most excellent value of 'k' parameter at the moment of training and at the time of classification for each given test instance, a local Naive Bayes within the best 'k' nearest neighbor is lazily constructed. It significantly enhance the performance of KNN [11]. A chi square statistical test based feature weighting method was developed by D. P. Vivencio *et al in* 2007. The chi-square χ^2methodbased formula which is used to measure common information between variables *i* and *j* is given below where *i* is the discrete variable which can assume *l* possible values, another variable is *j* which can assume c possible values, n_{ij} is the observed frequency an e_{ij} is the expected frequency[12].

$$\chi^2 = \sum_{i=1}^{l} \sum_{j=1}^{c} \frac{\left(n_{ij} - e_{ij}\right)^2}{e_{ij}}$$

This weighting method has good performance in datasets when huge number of irrelevant features is randomly inserted into datasets.

In 2008, an attribute weighted technique developed by Bao *et al* which uses information gain and extension relativity method to assign weights. The attributes with more information gain are called primary attributes as well as the attributes with less information gain are called secondary attributes and the attributes with zero information gain are called irrelevant attributes. The n-dimensional attribute space is divided into sub-spaces with primary, secondary and irrelevant attribute samples. The extension relativity is calculated between unknown sample and sub-spaces. Then, after calculating the Euclidean distance between unknown sample and searching sub-spaces, the class of majority voting is assigned to an unknown sample. This technique improves the anti-jamming ability and classification accuracy of KNN classifier. It highly decreases the time complexity of traditional KNN [13]. Xiao and Ding in 2012 suggested a weighting method based on weighted entropy of attribute value which enhances the accuracy of classification [14]. It first calculates the information entropy of each attribute and then calculates the weighted Euclidean distance using information gain as the weight coefficient, the formula is:

$$d(x,y) = \sqrt{\sum_{i=1}^{m} info(v_i)\,(x_i - y_i)}$$

Where x and y are two tuples, $info(v_i)$ is the information gain of attribute value and it is calculated using the formula as given:

$$info(v_i) = D(v_i) - \sum_{i=1}^{n} \frac{T_i}{k} \ln\left(\frac{T_i}{k}\right)$$

Where $D(v_i)$ is information entropy, is given by

$$D(v_i) = -\sum_{j=1}^{m} \ln(p_{ij})$$

M.E. Syed devised a chi-squared based attribute weighting technique in 2014 with two variations in weighting, one of them is Normal weighting technique in which a single weight is calculated for each attribute and second is class-wise weighting method in which different weights are calculated for different classes for every attribute. The class-wise weighting method is very helpful in recognizing the minor classes and in enhancing the rate of accuracy for minor classes. It uses the Heterogeneous Euclidean Overlap Metric (HEOM) for calculating the distance between two participation vectors. HEOM distance function to calculate the distance between two vectors x and y is [5]:

$$HEOM(x,y) = \sqrt{\sum_{a=1}^{m} \sqrt{d(x_a, y_a)^2}}$$

Where '*a*' represents an attribute and '*m*' represents the total number of attributes. The distance between two values of given input vectors x and y of a given attribute '*a'* is given by:

$$d(x_a, y_a) = \begin{cases} 1, (if\, x_a or y_a is unknown) \\ overlap(x_a, y_a), \quad (if\, a is nominal) \\ rn_{diff}(x_a, y_a) \end{cases}$$

Where $overlap(x_a, y_a) = \begin{cases} 0, & (if\, x_a = y_a) \\ 1, & (otherwise) \end{cases}$

$$rn_{diff}(x_a, y_a) = \left|\frac{x_a - y_a}{range_a}\right|$$

$$range_a = max_a - min_a$$

Here, max_a and min_a are the highest and lowest values of attribute '*a'* which is taken from the training set. Li *et al* proposed an attribute weighting method where the unrelated attributes are reduced and weights are assign to each attribute by using the method of sensitivity which increase the efficiency of algorithm [15]. A new algorithm based on dynamic weighting to enhance the classification accuracy of the KNN algorithm was devised by K Maryam. It allocates weights to attributes so that the impact of less important attributes can be reduced. It enhances the classification accuracy of KNN algorithm [16].

2.2 Distance weighted KNN

Distance weighting methods are also known as instance weighting KNN. An instance weighting method proposed by Schliep Hechenbichler in 2004[17] in which the instances obtain more weight on the basis of their proximity to the new instance than such neighbors that are distant away from the new instance. It assign weights to tuples depending on the distance from the unknown tuple by using the triangular kernel function .In this method the effect of 'k' parameter can be reduced up to some extent. A correlation based weighted KNN was proposed by Li and Xiang in 2012. For each query instance the correlation is calculated between the query instance and each training sample. To identify a given class it uses the weighted average probability. The formula is given below to calculate weighted average probability is as follow:

$$P = \frac{P_{i-k}}{P_i}$$

Where P stands for the probability of the query instance that belongs to the identified class *i*. P_{i-k} is the probability of training samples of classes *i* among its 'k' most relevant samples. P_i is the probability of class *i* in whole training set [18]. It first calculates the attribute weights using information gain, then divide the dataset into clusters and standard Euclidean distance between the given test sample and the centre of every cluster is calculated. When an unknown tuple is given the weighted Euclidean distance between the test samples and the training samples in its closest cluster, is calculated and using weighted class probability estimation method the class label is assigned to the unknown tuple. This technique reduces the computing time and avoids errors that are generated by the samples within unbalanced class in one training set. Gou *et al* in 2012 developed a distance weighted KNN rule using the dual distance weighted function to reduce the dependency problem of the selection of neighborhood size 'k'. This technique can deal with outliers in the neighborhood area of a data space. The degree of the sensitivity of different choices of 'k' can be degraded [19].

2.3 Hybrid KNN

A combination of these two techniques, called Hybrid dynamic KNN with attribute weighted and distance weighted algorithm was devised by Jia Wu *et al* in 2012. It weights the distance with mutual information and the most excellent value of 'k' is eagerly learned. It also weights the 'k' nearest neighbors using mutual information[20]. Another combined technique of attribute weighted and distance weighted with a different weighting method was suggested by Shweta Taneja *et al* in 2014. It assigns weights to each attribute based on the method of information gain and then divides the training dataset into clusters. Means of all the clusters is found out to obtain the centre of cluster and Euclidean distance is then calculated between the test sample and the centre of every cluster. The weighted Euclidean distance is calculated between the test sample and the training samples in its closest cluster. The classification is done using majority weighted class probability estimation method and it increases the accuracy of classification and decreases the execution time[21].

Table 1. Comparison of Nearest Neighbor Techniques

Sr. No.	Technique Name	Key idea	Merit	Demerit
1.	Dynamic KNN Naive Bayes with attribute weighted	It uses common information to assign weights to attributes	Significantly outperforms in terms of accuracy	It is time consuming to learn the best 'k' value at the time of training
2.	Feature-weighted K-Nearest Neighbor	It uses chi-squared statistical test to assign weights to attributes	1. It gives better performance in datasets with a large number of irrelevant attributes 2. Normalized weighting vector generation achieves best results as compared to other weighting vector generation criterions	No improvement in accuracy when all or most of the attributes are relevant or having equal weights
3.	An Enhancement of KNN using Information Gain and Extension Relativity	Information gain and extension relativity to assign weight coefficients	1. The anti-jamming ability improved 2. Accuracy highly improved 3. Time complexity reduced 4. Reduces searching space	Uses simple voting to classify the sample which biasing towards majority class
4.	Enhancement of KNN, depending upon Weighted Entropy of Attribute Value	Information entropy is used to assign weights to attributes	Effectively improved the classification accuracy of traditional KNN	Consumes more time in categorical datasets
5.	Attribute weighting in KNN	It uses chi-squared statistical test to assign weights to attributes with two variations, normal weighting and class wise weighting care used	The class wise weighting is very useful to predict the smaller class and it improves the classification accuracy in imbalanced datasets	The majority class cannot be correctly classified by using Class-wise weighting method
6.	Weighted KNN and its application on UCI	It uses method of sensitivity to assign weights to each attribute	It improves the classification performance	
7.	Weighted KNN techniques and ordinal classification	Weights to instances are assigned according to their closeness to the test tuple	Automatic adjustment of 'k' takes place so the results for higher 'k' reaches to the optimal solution	Classification accuracy degraded in case high dimensional datasets
8.	Correlation based KNN	Weights to instances are assigned according to their correlation between the test tuple and training tuples	It decreases the complexity of arithmetic and it also saves the accounting time	1. Computational cost increases in calculating weights 2. Applicable for nominal datasets
9.	A new distance weighted KNN	Weights to instances are assigned according to their closeness to the test tuple	1. It can deal with outliers in local region 2. Sensitivity of KNN for 'k' value can be reduced up to some extent Classification performance increases	1. Computational cost increases in calculating weights 2. Slow in execution
10.	An enhanced KNN using Information Gain and Clustering	Attribute weights are assigned using information gain	1. It enhances the accuracy of classification 2. It decreases the execution time.	Tested only for one dataset
11.	Hybrid dynamic KNN with attribute and distance weighting	Uses mutual information to assign weights	1. It improves the classification accuracy It reduces the execution time	Learning the best value of 'k' is time consuming

3. CONCLUSION

This paper gives a brief review of variety of KNN techniques. This review would be helpful for the researchers in focusing on the various issues of KNN algorithm like curse of dimensionality, memory limitations, computational cost, too much time consumption in searching 'k' nearest neighbors for a test tuple in large or multimedia training datasets, slow at classification process etc. Most of the previously done studies on KNN applications in various fields, researchers use different types of data like spatial, categorical, continuous, text, graphics, geo- data, moving objects etc. For these different data types different techniques are developed by many researchers like class wise attribute weighting technique is developed to solve the problems in imbalanced datasets. Distance weighted techniques play an important role to solve the sensitivity problem of selecting neighborhood size 'k' in KNN and it can deal with outliers in local region. Attribute weighting techniques are very useful to reduce the curse of dimensionality and also overcome the problem of equal impact of all attributes. This paper concludes that the factors which don't let the work to be completely successful are curse of dimensionality, selection of 'k' parameter value and biasing towards majority class prediction etc. The researchers were doing well in making the outcomes of their research work successful but the problem of getting most accurate and general KNN classification model is still there. Attribute weighting, distance weighting methods are most commonly used to enhance its performance. In future work Attribute weighting and Distance weighting methods can be combined with different weighting strategies to enhance the performance of KNN.

4. REFERENCES

[1] N. Padhy 2012. "The Survey of Data Mining Applications and Feature Scope," Int. J. Comput. Sci. Eng. Inf. Technol., vol. 2, no. 3, pp. 43–58, Jun.

[2] J. Han and M. Kamber 2006. Data mining: concepts and techniques, 2nd ed. Amsterdam ; Boston : San Francisco, CA: Elsevier.

[3] S. Bagga and G. N. Singh 2012. "Applications of Data Mining," Int. J. Sci. Emerg. Technol. Latest Trends, vol. 1, no. 1, pp. 19–23.

[4] S. Sethi, D. Malhotra, and N. Verma . 2006. "Data Mining: Current Applications & Trends," Int. J. Innov. Eng. Technol. IJIET, vol. 6, no. 14, pp. 667–673, Apr.

[5] M. E. Syed .2014. "Attribute weighting in k-nearest neighbor classification," University of Tampere.

[6] L. Jiang, H. Zhang, and Z. Cai. 2006. "Dynamic k-nearest-neighbor naive bayes with attribute weighted," in International Conference on Fuzzy Systems and Knowledge Discovery, 2006, pp. 365–368.

[7] I. H. Witten and E. Frank. 2005. Data mining: practical machine learning tools and techniques, 2nd ed. Amsterdam ; Boston, MA: Elsevier.

[8] S. B. Imandoust and M. Bolandraftar. 2013. "Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background," Int. J. Eng. Res. Appl., vol. 3, no. 5, pp. 605–610.

[9] R. Kumar and R. Verma. 2012. "Classification algorithms for data mining: A survey," Int. J. Innov. Eng. Technol. IJIET, vol. 1, no. 2, pp. 7–14.

[10] L. Jiang, Z. Cai, D. Wang, and S. Jiang. 2007. "Survey of improving k-nearest-neighbor for classification," presented at the Fourth International Conference on Fuzzy Systems and Knowledge Discovery, 2007, vol. 1, pp. 679–683.

[11] Liangxiao Jiang, Harry Zhang, and Zhihua Cai. ,2006. "Dynamic k-nearest-neighbor naive bayes with attribute weighted," presented at the International Conference on Fuzzy Systems and Knowledge Discovery, 2006, pp. 365–368.

[12] D. P. Vivencio, E. R. Hruschka, M. do Carmo Nicoletti, E. B. dos Santos, and S. D. Galvao. 2007. "Feature-weighted k-nearest neighbor classifier," presented at the Foundations of Computational Intelligence,2007, pp. 481–486.

[13] W. Baobao, M. Jinsheng, and S. Minru. 2008. "An enhancement of K-Nearest Neighbor algorithm using information gain and extension relativity," presented at the International Conference on Condition Monitoring andDiagnosis, 2008, pp. 1314–1317.

[14] X. Xiao and H. Ding. 2012. "Enhancement of K-nearest neighbor algorithm based on weighted entropy of attribute value," presented at the Fourth International Conference on Advanced & Communication Technologies, 2012, pp. 1261–1264.

[15] Z. Li, Z. Chengjin, X. Qingyang, and L. Chunfa. 2015. "Weigted-KNN and its application on UCI," presented at the International Conference on Information and Automation, 2015, pp. 1748–1750.

[16] X. Li and C. Xiang.2012. "Correlation-based K-nearest neighbor algorithm," presented at the 3rd International Conference on Software Engineering and Service Science, 2012, pp. 185–187.

[17] K. Hechenbichler and K. Schliep. 2004. "Weighted k-nearest-neighbor techniques and ordinal classification," Institue for statistik sonderforschungsbereich, 386.

[18] Maryam Kuhkan. 2016. "A Method to Improve the Accuracy of K-Nearest Neighbor Algorithm," Int. J. Comput. Eng. Inf. Technol. IJCEIT, vol. 8, no. 6, pp. 90–95, Jun.

[19] J. Gou, L. Du, Y. Zhang, T. Xiong, and others. 2012. "A new distance-weighted k-nearest neighbor classifier," J Inf Comput Sci, vol. 9, no. 6, pp. 1429–1436.

[20] J. Wu, Z. hua Cai, and S. Ao. 2012. "Hybrid dynamic k-nearest-neighbour and distance and attribute weighted method for classification," Int. J. Comput. Appl. Technol., vol. 43, no. 4, pp. 378–384.

[21] S. Taneja, C. Gupta, K. Goyal, and D. Gureja. 2014. "An Enhanced K-Nearest Neighbor Algorithm Using Information Gain and Clustering," presented at the Fourth International Conference on Advanced Computing & Communication Technologies, 2014, pp. 325–329.