Empirical Evaluation of KNN Classifier Using Various K-Values

Obunadike Georgina N. 1*
Department of Computer Science and IT
Federal University, Dutsinma
Katsina State, Nigeria
gobunadike@fudutsinma.edu.ng

Joushua Abah 2
Department of Computer Engineering,
University of Maiduguri,
Maiduguri,
Borno State, Nigeria.

Dima R. M[3]
Department of Computer Science and IT
Federal University, Dutsinma
Katsina State, Nigeria
gobunadike@fudutsinma.edu.ng

Abstract - *KNN is famous K- Nearest Neighbour algorithm and one of the top popular algorithms in machine learning and data mining. It is simple to use and have application in different area of life endeavour. It searches for the most similar elements in a dataset using similarity function. Unlike Naïve Bayes, Support Vector Machine, Neural network and Decision Tree classifiers, KNN is good for data stream because it builds not its classifier or model in advance. It usually applies different similarity methods like Euclidean distance, simple matching and squared Euclidean distance in making its classification. KNN is often very accurate and assumes all attributes are of equal importance. The performance of the classifier depends on choosing the optimal number of k value (neighbours). Thus, the challenge of the KNN classifier is how to choose the optimal k values. If k is large it is believed to give better result but not in all cases. This work discussed empirical evidence of how various K-values of KNN algorithms influences its performance using performance evaluation metrics such as accuracy, time, kappa statistic and ROC curve.*

Keywords— KNN Algorithm, classification, similarity measures, Instance based learning, Rote learning

I. INTRODUCTION

Ref [1] in their work was the first to introduce the Nearest Neighbour algorithm, since then KNN has been used in classification and regression problems of different forms. Though considered as a lazy classifier; has proven to be effective algorithm that can compete favourably with other leading classifiers [2]; [3] and [4]. It is perhaps one of the simplest machine learning algorithms and is widely used [4] and [5]. It has been applied successfully in different areas such as computer vision, recommendation system, pattern recognition, prediction and it is a supervised learning algorithm [6] and [7]. KNN gets its name from the fact that it uses information from a record's k-nearest neighbours to classify unlabelled record [4]. The letter k is variable which means that any number of nearest neighbours can be used. KNN identifies k-records in training data that are closest in similarity and assigned to the unlabelled data the class of the closest neighbour [8]. It uses distance measures to measure similarities between two data items. The common distance measures are Euclidean distance, Manhattan, simple matching, squared Euclidean distance and Minkowski [9]. Suppose there are two elements $x_1, x_2 \cdots$ and $y_1, y_2, \cdots$ The similarity function of these two elements can be stated as:

$$d(x, y,) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (1)$$

KNN does not apply model in classification and is memory based. Given a data item, the k numbers of data items closest to the data item to be classified are identified. It uses neighbourhood vote as the prediction value for the new data item. Normally, k-value of the KNN are empirically chosen by trying different number of nearest data items and the data item with the best value is usually chosen for the classification [10]. Picking the optimal K value is usually a challenge and this usually affects the performance of KNN [11]. When the data items are not uniformly distributed, the choosing of the optimal k value also becomes a challenge [12]. Therefore, the purpose of this work is to study the effect of different k values on classification accuracy and other performance metrics such as kappa Statistics, ROC Curve and time on different UCI dataset and to make recommendation accordingly. The rest section discussed the challenges of KNN, related works, data normalization, experimental setup, result discussion and conclusion

II. CHALLENGES OF KNN CLASSIFIER

KNN classifier is usually referred to as lazy learner because is not really learning anything but merely stores the training data verbatim, thus it just allows the training phase without training anything thereby slows down the prediction process. That is to say it makes predictions base on training instances rather on abstracted model. Thus, KNN is classified an Instance-based-learning or rote learning [10]. The strengths and weaknesses of KNN are enumerated in Table 1:

TABLE I. STRENGTHS AND WEAKNESSES OF KNN CLASSIFIER

Strengths	Weaknesses
Simple and effective **Makes no assumptions about the underlying data distribution** **Fast training phase**	**Does not produce a model, limiting the ability to understand how the features are related to the class** **Requires selection of an appropriate k** **Slow classification phase** **Nominal features and missing data require additional processing**

Source (Brett, 2015)

III. RELATED WORKS

Before The choice of k parameter in KNN algorithm usually affect the classifier performance, depending on the nature of the problems, different value of k parameters is usually applied in the classification and the value that gave the optimal performance is usually chosen for the classification. It has been revealed in literature that when the dataset is not uniformly distributed, the choice of k value is usually affected [12].

Ref [13] is of the opinion that the performance of KNN algorithm is optimized when large k value is applied. Thus, they applied large values of k such as 30, 45 and 60 and their result was tabulated [13] and [14].

Ref [15] in their work stated that many factors such as data size, dimension and distribution determines the choice of k value. Ref [16] in their work tried to increase the KNN classifier accuracy rather than parameter tuning for optimal k value. Ref [2] applied bootstrap method to enhance the performance of the KNN. The evaluation result showed that their methodology outperformed the traditional KNN classifier.

It has been observed from literature that most work used only accuracy to evaluate performance of KNN against the various k values. This work used different k values starting from.

$$k = 1, 3, 5 \text{ and } k = \sqrt{n} \text{ where } n \text{ is the number of training set to large } k \text{ values as } k = 45, 60 \text{ and } 100$$

The classifier applied majority vote for the classification. The classifier performance was evaluated using four common machine learning performance evaluation metrics such as accuracy, time, ROC curve and kappa statistic.

IV. METHODOLOGY

This section discussed the process of the analysis from data preparation methodologies to data set selection and analysis using KNN.

A. Data Preparation in KNN

Data normalization is one of the major data preparation task when applying KNN classifier for prediction or classification problems. The reason for this is that distance formula is highly dependent on measurement range of the data items. When a data item have a dominant data range values than others, the distance measurement will also be dominated by that data item. The best approach is usually to scale the data values upwards or downwards so that each one will contribute equally in the distance formula. There are many methods of data scaling or normalization. The conventional approach is Min-Max normalization which scales data to fall between the range of 0 and 1. The formula is given as shown in Equation 2.

$$Z_{new} = \frac{Z - Min(Z)}{Max(Z) - Min(Z)} \quad (2)$$

The letter represents the data values. Another method is called the Z-score standardization. This subtracts the mean value of data and divides the output by the standard deviation; the formula is shown in Equation 3.

$$Z_{new} = \frac{Z - \mu}{\sigma} = \frac{Z - mean(z)}{std(Z)} \quad (3)$$

The result usually fall in a range of negative and positive numbers usually called the Z-score.

The distance function is not made for nominal data, thus, to calculated distance function for nominal data it needs to be converted to numeric format. The common method of transforming the nominal to numeric is the dummy approach where a value of 1 indicates one category and 0 indicate the other category [17]. Equation 4 indicates the dummy coding for gender variable.

$$female = \begin{cases} 1 \text{ if } x = female \\ 0 \text{ other wise} \end{cases} \quad (4)$$

B. Experimental Setup

The work applied Euclidean distance in the conventional KNN methodology. The KNN classifier is used in each of the iteration with different k value. The value of k is taking from the range k = 1 to k =5; k = square root of k. where n is the number of instances in the dataset. In addition, large no of k neighbours such as K = 45, 60, 100 were also applied to check if it will help to increase the accuracy of the classifier as argued in the literature that performance of KNN improves when using large k values [13] and [14]. It uses neighbourhoods vote for classification. The KNN classifier then uses the majority vote to identify class label for the new instance. Thus, the class with majority votes is usually chosen for the classification on the unlabelled data item. The k value

is limited to be equal to the selected k values. This is to reduce computation time and obey the standard rule that states that k should not be greater than the square root of the data size [16]. The highest accuracy with the lowest computational cost is chosen as optimal result. This work used odd numbers of k values to avoid ties in the votes and to increase classification speed. 10-fold cross validation split was use for the analysis. Five UCI data set namely: glass, Irish, vote, diabetes and segment have been chosen to study the effect of different k values on the classification accuracy on KNN on these datasets.

C. Dataset Used

The KNN classifier was applied on these five-dataset shown in Table 2 which was collected from UCI data repository

TABLE II. DESCRIPTION OF DATASET USE

Dataset Name	No of instances	Data Type	No of Attributes	No of Class
Glass	**214**	**Numeric**	**9**	**6**
Diabetes	**768**	**Numeric**	**8**	**2**
Irish	**150**	**Numeric**	**4**	**3**
Vote	**399**	**Numeric**	**10**	**2**
segments	**1500**	**Numeric**	**19**	**7**

V. RESULT DISCUSSION

The Table 3 to Table 7 below showed the result of the KNN classification using different K values on five UCI datasets. The evaluation metrics used are accuracy, time, kappa statistic and ROC Curve as discussed below:

Accuracy: this is the percentage of the correctly classified data item

Time: this is the time taken for the classification using different k values

Kappa: measures the relationship between the classification and the real class value. A value of 1 implies perfect correlation while 0 implies guessing.

ROC Curve: it is used to visualize the performance of a classifier, value of 1 implies perfect model while 0.5 implies guessing model.

TABLE III. .. IRISH DATASET CLASSIFIED USING DIFFERENT K VALUES

Parameters	K= 1	K= 3	K= 5	K=sqrt (150)	K= 45	K=60	K=100
Accuracy	**95.3**	**95.3**	**95.3**	**96.7**	**93.3**	**89.3**	**66,7**
Time	**0 secs**	**0.01 secs**	**0 secs**	**0 secs**	**0 secs**	**0 secs**	**0 secs**
Kappa	**0.93**	**0.93**	**0.93**	**0.95**	**0.90**	**0.84**	**0.5**
ROC Curve	**0.966**	**0.974**	**0.995**	**0.997**	**0.99**	**0.93**	**0.83**

TABLE IV. GLASS DATASET CLASSIFIED USING DIFFERENT K VALUES

Parameters	K= 1	K= 3	K= 5	K=sqrt (214)	K= 45	K=60	K=100
Accuracy	**70.6**	**72.0**	**67.8**	**66.6**	**57.9**	**54.7**	**34.5**
Time	**0 secs**	**0 secs**	**0 secs**	**0.01 secs**	**0 secs**	**0 secs**	**0 secs**
Kappa	**0.60**	**0.61**	**0.55**	**0.47**	**0.39**	**0.35**	**0.01**
ROC Curve	**0.792**	**0.847**	**0.853**	**0.871**	**0.810**	**0.780**	**0.750**

TABLE V. VOTE DATASET CLASSIFIED USING DIFFERENT K VALUES

Parameters	K= 1	K= 3	K= 5	K= sqrt (435)	K= 45	K=60	K=100
Accuracy	**92.4**	**92.6**	**92.6**	**91.6**	**90.5**	**90.1**	**89.4**
Time	**0 secs**	**0 secs**	**0 secs**	**0 secs**	**0 secs**	**0 secs**	**0 secs**
Kappa	**0.84**	**0.85**	**0.85**	**0.83**	**0.81**	**0.80**	**0.78**
ROC Curve	**0.965**	**0.975**	**0.981**	**0.984**	**0.980**	**0.980**	**0.980**

TABLE VI. SEGMENT DATASET CLASSIFIED USING DIFFERENT K VALUES

Parameters	K= 1	K= 3	K= 5	K= sqrt (1500)	K= 45	K=60	K=100
Accuracy	**96.2**	**95.1**	**94.8**	**88.6**	**88.2**	**86.1**	**83.9**
Time	**0 secs**	**0 secs**	**0.01 secs**	**0.02 secs**	**0 secs**	**0 secs**	**0 secs**
Kappa	**0.96**	**0.94**	**0.94**	**0.94**	**0.86**	**0.83**	**0.81**
ROC Curve	**0.978**	**0.989**	**0.991**	**0.991**	**0,99**	**0.987**	**0.98**

TABLE VII. DIABETES DATASET CLASSIFIED USING DIFFERENT K VALUES

Parameters	K= 1	K= 3	K= 5	K= sqrt (768)	K= 45	K=60	K=100
Accuracy	**70.2**	**72.6**	**73.1**	**74.1**	**74.7**	**74.8**	**72.7**
Time	**0 secs**	**0 secs**	**0 secs**	**0 secs**	**0 secs**	**0 secs**	**0 secs**
Kappa	**0.33**	**0.38**	**0.39**	**0.38**	**0.39**	**0.38**	**0.81**
ROC Curve	**0.65**	**0.742**	**0.766**	**0.809**	**0.812**	**0.811**	**0.807**

The result showed that there is no optimal k value. Thus, no specific numbers of k-values are suitable for all the datasets used in the experiment; each dataset classification accuracy performs well at different k values (neighbour). On the other hand, setting higher value for k did not yield optimal result in most cases as argued in literature that the performance of KNN algorithm increases as k value increases; perhaps that might depend on the nature of the dataset.

VI. CONCLUSION

Machine learning is concerned with the development of algorithms that transforms data into intelligence information. This makes machine learning an ally to data mining especially in this era of big data. Data mining and machine learning are usually misrepresented. Machine learning is concerned with teaching the computer how to used data to solve problem and data mining on the other hand is concerned with teaching the computer to identify patterns from data that can help in decision making. Mostly all data mining activities involves the use of machine learning but not all machine learning that involve data mining. Machine learning algorithms are usually applied in data mining activities. Among the machine learning algorithms usually applied for data mining is the K-Nearest Neighbour (KNN) algorithm. KNN is a popular classification algorithm and also known as instance based learning or rote learning classifier. The performance of the KNN is usually dependent on the choice of k value applied. This work performs empirical study on how various k values affects KNN algorithm performance. It is observed that there is no optimal k value that works perfectly across the dataset. The performance of the classifier was observed to depend mainly on different k values for each dataset. Moreover, it was observed that the use of various k values has no significance effect on the classification time. No significance difference was also observed from most of the ROC curve values and kappa statistic values using different k values in each of the iteration. Thus, it can be inferred that there is no optimal k value that work for all dataset and classification problems. It can also be inferred that optimal k occur in classification with a particular dataset when all the metrics are optimal.

REFERENCES

[1] Fix, E. and Hodges, J. (1951). Discriminatory Analysis: Nonparametric Discrimination: Consistency Properties, 4,

[2] Hamamoto, Y. Uchimura, S. and Tomita, S. (1997). A Bootstrap Technique for Nearest Neighbor Classifier Design, IEEE Transactions On Pattern Analysis And Machine Intelligence, vol. 19, no. 1, pp. 73-79

[3] Weinberger, K. Q. and Saul, L. K. (2009). Distance Metric Learning for Large Margin Nearest Neighbor Classification, Journal of Machine Learning Research, vol. 10, pp. 207-244

[4] Kataria, A. and Singh, M. D. (2013). A Review of Data Classification Using K-Nearest Neighbour Algorithm, International Journal of Emerging Technology and Advanced Engineering, vol. 3, no. 6, pp. 354-360

[5] Bhatia, N. and Vandana, A. (2010) Survey of Nearest Neighbor Techniques, International Journal of Computer Science and Information Security, vol. 8, no. 2, pp. 302-305

[6] Hassant, A. B. A (2011). Visual Speech Recognition, in Speech Technologies, I. Ipsic, Ed. Rijeka: InTech - Open Access Publisher, vol. 2, ch. 14.

[7] Hassanat, A. B. A. (2014). Visual Passwords Using Automatic Lip Reading, International Journal of Sciences: Basic and Applied Research (IJSBAR), vol. 13, no. 1, pp. 218-231.

[8] Kubat, M. and Jr, M. (2000). Voting Nearest-Neighbour Subclassifiers," in Proceedings of the 17th International Conference on Machine Learning, ICML-2000, Stanford, CA, pp. 503-510.

[9] Parvin, H. Alizadeh, H. and Minaei, B. (2010) A Modification on KNearest Neighbor Classifier, Global Journal of Computer Science and Technology, vol. 10, no. 14, pp. 37-41.

[10] Guo, G., Wang, H., Bell, D., Bi, Y. and Greer, K. (2003). KNN Model-Based Approach in Classification, Lecture Notes in ComputerScience, vol. 2888, pp. 986-996

[11] Song, Y., Huang, J., Zhou, D., Zha, H. and Giles, C. L. (2007). Iknn: Informative k-nearest neighbor pattern classification, in Proceedings of the 11th European conference on Principles and Practice of Knowledge Discovery in Databases, pp. 248-264.

[12] Latourrette, M. (2000). Toward an explanatory similarity measure for nearest-neighbor classification," in Proceedings of the 11th European Conference on Machine Learning, London, pp. 238-245.

[13] Yang, Y. and Liu, X. (1999). A re-examination of text categorization methods, in Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval, Berkeley, pp. 42-49

[14] Yang, Y. (1999). An evaluation of statistical approaches to text categorization," Information Retrieval, vol. 1, pp. 69-90

[15] Enas, G. G. and Choi, S. C. (1986). Choice of the smoothing parameter and efficiency of k-nearest neighbor classification, Computers &Mathematics with Applications, vol. 12, no. 2, pp. 235-244

[16] Jirina, M. and Jirina, M. J. (2011). Classifiers Based on Inverted Distances, in New Fundamental Technologies in Data Mining, K. Funatsu, Ed. InTech, vol. 1, ch. 19, pp. 369-387

[17] Brett, L. (2015). Machine Learning with R:Second Edition, PACKT Publishing, pp. 65 - 87

.