An Efficient Weighted Neighborhood Approach for Modern Data Classification

Tamer TULGAR
Department of Computer Engineering
Girne American University
Girne, T.R.N.C., Mersin 10 TURKEY
tamertulgar@gau.edu.tr

Abstract— **In the era of information, businesses need to give base their data driven decisions on efficient and accurate data analysis. To meet this need, many different existing data mining strategies are needed to be adapted to the modern data analysis needs. One of the most important data mining tasks is classification and the K Nearest Neighbor classification is a well-accepted data classification method. It is known that the K-NN is an individual distance based algorithm rather than relying on a general center representation of data classes, which is a cause of false classification decisions. In this study, the K-NN algorithm was modified to remedy the algorithm's classification accuracy degradation, by basing the majority voting decision of the K-NN algorithm on weighted distances of the K nearest neighbors. Specifically, this paper proposes to consider the contribution of each of the nearest neighbors to decide which data class should be chosen as the class of the test data to be classified. To evaluate the performance improvement of the proposed method, several experiments were carried out using different real datasets. The presented results, which are achieved after extensive experiments, prove that the proposed algorithm improves the classification accuracy of the classical K-NN algorithm. The achieved performance was also compared against different recent classification schemes.**

Keywords- Classification, K-Nearest Neighbor, Weighted majority representation

I. INTRODUCTION

Recently, the data mining needed in the era of information, needs to be improved so that the ever-growing volumes of data retrieved from many different online streaming sources can be analyzed and processed [1].

Analyzing data coming from vast amount of data sources, results in working with huge amounts of unstructured raw data, which are big in terms of volume, variety and velocity of acquisition. The process of analyzing such data is recently a popular research area, known as the Big Data Analysis [2].

Classification as the task of assigning data instances to one of several predefined data classes is a pervasive problem that encompasses many diverse applications [3]. To classify data in the big data age, centralized techniques need to be improved so that the current data analysis challenges of working with the high velocity data streams of big data can be tackled. Especially, to meet the timing requirements of the modern data classification, the centralized methods are being distributed on environments like the Apache Hadoop and the Google's MapReduce Framework [4].

K Nearest Neighbor Classification (K-NN) has been one of the most popular classification algorithms [5]. Nevertheless, the classical K-NN algorithm suffers from high computation costs, which makes it inconvenient for modern big data analysis that requires rapid and accurate classification results.

The classical K-NN algorithm is based on calculating the distances between the test data instance to be classified and all of the instances in the training data set and finding the closest K number of training instances. After detecting the K number of closest training instances, the K-NN algorithm applies majority voting which is the process of detecting the data class with the maximum number of instances among the K selected instances.

Traditionally, individual instance distances based classification strategy of the classical K-NN algorithm makes this algorithm perform weakly in terms of the classification accuracy. On the other hand, K-NN's individual instance distances strategy makes K-NN a strong candidate for distributed data classification, which is the basis of achieving acceptably low classification delays while classifying big data.

Taking into account the K-NN's suitability to distributed environments, many K-NN based studies, which try to improve the K-NN algorithms performance, have been proposed in the literature [6-16] to meet the modern data mining needs.

An improved K-NN classification algorithm is proposed in this study, which is named as the Weight Improved K-NN algorithm (WI-KNN), and is expected to improve the classification accuracy performance of the classical K-NN algorithm.

The main contribution of the WI-KNN algorithm is giving classification decisions by representing the strengths of the data instances among the K nearest neighbors so that more accurate data class detection can be made as opposed to the classical majority voting decision step of the classical K-NN.

In other words, the nearest neighbors evaluated by the distance calculations are used to calculate weights, in other words how well each instance contributes to the nearest neighborhood is considered so that the cumulative class strengths can be computed and used to decide which class better represents the test data that is needed to be classified.

To test how the proposed WI-KNN algorithm performs, extensive experiments are carried out and the achieved results are compared against several other studies proposed in the literature.

The rest of this paper is organized as follows: Some recent literature is summarized in Section II. The Section III presents the proposed WI-KNN algorithm in detail. The experimental setup and the achieved performance results are presented in section IV. Finally, the section V concludes the paper and states the future works.

II. SOme Recent KNN Based Studies

In [10], three schemes for classification are proposed and compared. The proposed schemes are K-nearest Neighbour (KNN), Fuzzy KNN and the Support Vector Machine (SVM). The Fuzzy KNN proposed in [10] employs Gaussian Membership Functions as the representatives of the data clusters, which is one of the details pointed out in [10]. In the results, the author presents the experimental results, which show that among their proposed alternatives the scheme, which combines Support Vector Machine with Soft Labels, produces the better classification accuracies.

Another MapReduce fuzzy data classification scheme is proposed in [11]. In [11], the authors propose four different schemes and compare their performances. The four proposed classification techniques are fuzzy KNN and mode function, SVM classifier and mode function, SVM and soft labels and finally SVM classifier and fuzzy Gaussian membership function. In [11], the four methods mainly differ in the Reducer function part of the MapReduce such that the reducers are implemented using three approaches, which are, the mode, the soft labels and fuzzy Gaussian. The results presented in the study illustrate that the fuzzy techniques perform better than the crisp methods. Especially, the SVM using soft labels produces the better results.

In [13], the authors propose a K-Means variation together with a KNN classification approach. The proposed method in [13] clusters the data using the K-Means algorithm and then for testing, relies on KNN Classification. It is claimed by the authors of [13] that their proposed method is suitable for dealing with big data. The results they present outperforms the results of [14], which will be summarized next in this section

The method proposed in [14] modifies the KNN algorithm with a self-representation of the data clusters ideology. The presented main aim is to learn an optimal k value in KNN to improve the accuracy of the classification. To support their claim. the authors compare their results with three other algorithms named as kNNC, LMMN and ADNN which are summarized in [14]. The results presented in the paper shows better performance when compared to these three algorithms.

Authors of [15] compare and analyze five different existing methods to deduce the strengths and weaknesses of the KNN classification scheme for big data. As evaluation, [15] presents the advantages and disadvantages of the different stages of the compared classification models which are all applied on MapReduce workflow. It is claimed in [15] that the results achieved in the study can be used to tackle different practical KNN problems in the context of big data.

In [16], another KNN based classification scheme is proposed. The proposed study in [16] can be mainly summarized as an iterative version of MapReduce workflow based on SPARK which benefits from the KNN classification. The performance of the method proposed in [16] is evaluated using experiments. The results of the presented experiments illustrate that the method performs better than the KNN approaches based on Hadoop from both accuracy and runtime points of view.

III. The Proposed WI-KNN Algorithm

A. The Classical K-NN Algorithm

In a classification task, if a data instance is considered as a vector of feature values, then a data instance i can be denoted as vi which corresponds to a vector containing p features. Therefore, classifying a data instance means detecting the correct data classes of n test data instances by comparing their similarity to m training data instances using their feature values.

The classical K-NN algorithm is based on the idea of calculating the distances of a test data to be classified to all of the of data instances in the training data set. On computing the distances, the classical K-NN considers the K number of minimum distances and uses the training instances, which produced those K distances. These K training instances are the K Nearest Neighbors of the tested data. The classical Euclidean Distance Calculation used in K-NN is given in (1).

$$dist_{ij} = \sqrt{\sum_{k=1}^{p}\left(tsi_{feat_k} - trj_{feat_k}\right)^2} \qquad (1)$$

After detecting the K nearest neighbors, the classical K-NN detects which data class has the most number of instances among the selected K nearest neighbors, which is known as the process of majority voting. As it can be understood from the given explanation of the classical K-NN algorithm, the classification decision is solely based on the individual instance distances between all testing and training instances.

Nevertheless, when a data set with high number of instances and high number of features per instance needs to be classified, the classical K-NN algorithm's classification accuracy performance becomes lower than its competitors, like K-Means classification [17].

B. The proposed CR-KNN Algorithm

WI-KNN offers improvement over the classical KNN algorithm mainly in the classification decision step where traditionally the majority voting is used. The proposed WI-KNN algorithm relies on calculating weights by finding how much a training instance's distance contribute to the cumulative weight of all K nearest neighbors.

The proposed WI-KNN starts classification by calculating all distances between the test data instance to be classified and all other training instances. Then the K nearest neighbors are found which are the training instances having the closest proximity to the tested data i.

Upon finding the K nearest neighbors, the WI-KNN calculates the overall weight by summing all of the distances from the K Nearest Neighbors, given in (2).

$$Total\ Weight = \sum_{j=1}^{n} dist_{ij} \qquad (2)$$

Then the weight of each training instance among the K nearest Neighbors is calculated by dividing the overall distance to the distance of the training instance as in (3). Hence, the nearest neighbor with the smallest distance has the greatest weight and the nearest neighbor with the greatest distance has the smallest weight.

$$W_j = \frac{Total\ Weight}{dist_{ij}} \qquad (3)$$

After this step, WI-KNN, unlike classical KNN, finds which data classes exist among the K nearest neighbors and groups the nearest neighbors according to which data classes they belong to.

After the class detection, the WI-KNN finds the cumulative weight of each class and classifies the test instance to the class with the highest cumulative weight.

Please consider the following simplified example, where test instance 9 actually belongs to the class B, and the given are the detected K nearest neighbors. Looking at the distance values, the total weight is 0.11+0.14+0.14+0.17+0.21 = 0.75.

<ts_9, <tr_3, 0.11, A>, $W_3 = 0.75 / 0.11 = 6.82$

<ts_9, <tr_{11}, 0.14, B>, $W_{11} = 0.75 / 0.14 = 5.35$

<ts_9, <tr_{27}, 0.14, B>, $W_{27} = 0.75 / 0.14 = 5.35$

<ts_9, <tr_{23}, 0.17, C>, $W_{23} = 0.75 / 0.17 = 4.41$

<ts_9, <tr42, 0.21, A>, $W_{42} = 0.75 / 0.19 = 3.57$

In the example given above, 5 nearest neighbors are from classes A, B and C where tr_3 and tr_4 are from class A, tr_{11} and tr_{27} are from class B and tr_{23} is from class C. Hence, the cumulative weight for class A is W_3+W_{42} =10.39, the cumulative weight for class B is $W_{11}+W_{27}$ = 10.7 and the cumulative weight for class C is W_{23} = 4.41.

Upon calculating the cumulative weights, the WI-KNN calculates the maximum of the class weights for the testing instance to be classified and classifies the test data to the class with the maximum cumulative weight, in the example, the class B.

Calculating the weights for the testing instance contribution of the WI-KNN improves the classification accuracy of the classical K-NN by eliminating such misclassifications:

Please consider the simple example given at the top. In such a case, because of majority voting, the classical K-NN algorithm will conclude that the test instance 9 belongs to class A, since class A and class B has equal number of instances among the K nearest neighbors and the class A contains an instance, which has the closest proximity to the testing instance.

On the other hand, the same class A has an instance, which is further away from the class B instances to the test instance 9. To represent the similarity of the test instance to the training classes found among the k nearest neighbors in a more realistic and accurate way, the WI-KNN proposes to represent the classes A, B and C of the K nearest neighbors by cumulative weights and baseing the classification decision on the class with the maximum cumulative weight instead of relying on the individual data members. In this way, if class B elements look more alike the test instance, than the cumulative weight of class B will be higher and the proposed WI-KNN will correctly decide that the test instance should belong to class B.

The experimental results, which are presented in the next section, demonstrate that the proposed improvement enhances the performance of the classical K-NN algorithm for datasets with different natures.

IV. EXPERIMENTAL SETTING AND THE RELUTS

The proposed WI-KNN algorithm was implemented in Sun JAVA JDK 1.8 [18]. To evaluate and compare the performance of the proposed algorithm several datasets from the UCI machine learning repository [19] was used.

The summary of the used datasets are given in Table 1. While selecting the datasets, the factors like being binary or multi-class classification problem, various instance counts and different data distributions were considered so that the strengths and weaknesses of the proposed WI-KNN algorithm can be analyzed.

Also, it is worth to mention that the experimental model was validated using 10-fold cross validation and the presented classification accuracy values are the averages of the results found after cross validations.

TABLE I. REAL DATASETS USED IN THE EXPERIMENTS

Dataset	Instances	Features	Classes
ionosphere	351	34	2
wdbc	569	32	2
satimage	6435	36	7
pendigits	10992	16	10

A. Experimental Results

The results presented in this section demonstrate the classification accuracy of the proposed CR-KNN algorithm and other two recent classification schemes SR-KNN [14], which was summarized in section II of this paper and KMEANS-MOD [17]. The classification accuracy can be defined as the ratio of the number of correct classifications decisions to the number of total classification decisions for a given dataset's total number of testing instances.

The KMEANS-MOD proposed in [17], is an improved KMEANS clustering based algorithm which benefits from the effect of variance and introduces a membership strength as the metric to base its classification decisions. Since the algorithm proposed in [17] is a KMEANS based algorithm, the classification idea is completely dependent on strong class representations using whole class data centers.

Table II provides the classification accuracy results, which demonstrate the WI-KNN performance for different K values.

TABLE II. CR-KNN WITHOUT THE VARIANCE EFFCT VS INCLUDING VARIANCE EFFECT TO CR-KNN

	WI-KNN		
	K=5	**K=7**	**K=9**
pendigits	0.9771	0.9762	0.9759
satimage	0.9045	0.9000	0.8955
ionosphere	0.9035	0.8735	0.8529
wdbc	0.9655	0.9485	0.9385

By investigating the results, it can be seen that, reasonably small K values are enough to achieve high classification accuracy performance from the proposed WI-KNN algorithm.

The proposed WI-KNN algorithm was also compared against the classification schemes SR-KNN [14] and KMEANS-MOD [17] in terms of the classification accuracy performance to investigate how the proposed WI-KNN algorithm compares against both a recent Nearest Neighbor based classification scheme and a completely centroid representation based classification approach.

The comparative results mentioned above can be observed in Table III.

TABLE III. CR-KNN CLASSIFICATION ACCURACY PERFORMANCE COMPARISONS

	KMEANS-MOD	**Classical-KNN**	**SR-KNN**	**WI-KNN**
pendigits	0.9770	0.9078	0.9452	0.9771
satimage	0.8980	0.9065	0.8806	0.9045
ionosphere	0.9123	0.6286	0.8971	0.9035
wdbc	0.9628	0.6548	0.9650	0.9655

The comparative results presented in Table III illustrate that the proposed WI-KNN improves the performance of the classical KNN algorithm for the majority of the tested datasets.

Against the other K-NN based algorithm SR-KNN [14], the WI-KNN shows better classification accuracy performance for all datasets proving that the weighted cumulative distance representation for the data classes provide the expected improvement on the classification accuracy.

When compared with the KMEANS classification based KMEANS-MOD [17], which may have a better complete data class center representation with the cost of higher computation complexities, the proposed WI-KNN demonstrates a similar accuracy performance, if not better.

V. CONCLUSION

In this paper, a new Nearest Neighbor based classification algorithm is presented. The proposed algorithm is designed to improve the classical K-NN classification scheme with weighted distance representation of the data classes.

The proposed algorithm was tested with extensive experiments conducted on real datasets with different natures. The experimental results achieved after experiments show that the proposed algorithm improves the classification accuracy for the majority of the datasets. Also, the results demonstrate that the proposed algorithm is able to compete against other recently proposed classification schemes successfully.

The future works include focusing on the classification delay performance and testing the proposed algorithm on other datasets which include higher number of instances, in the margin of millions. Another planned future work is further improving the distance calculation, by replacing the distance concept with a novel similarity measure capable of providing similarity of mixed data including both continuous and categorical features.

ACKNOWLEDGMENT

The author of this paper thanks Prof.Dr.Ali HAYDAR and Assoc.Prof.Dr. Ibrahim ERSAN from Girne American University, Department of Computer Engineering, for their continuous support and valuable discussions on this study.

REFERENCES

[1] Klaus Schwab, "The Fourth Industrial Revolution", Crown Business, 2017.

[2] Singh and .K. Reddy, ”A survey on platforms for big data analytics”, Journal of Big Data vol. 1, no. 8, 2014.

[3] P. Tan, M. Steinbach and V. Kumar, ”Introduction to Data Mining”, 1st ed., Reading, MA: Addison-Wesley, 2005.

[4] J. Dean, S. Ghemawat , ”MapReduce: A Flexible Data Processing Tool”, Communications of the ACM, vol. 53 no. 1, pp.72-77, 2010.

[5] X. Wu et. Al., ”Top 10 algorithms in data mining”, Knowledge and Information Systems,vol. 14, no. 1, pp 137, 2008.

[6] Fahad et. AL., ”A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis”, IEEE Trans.on Emerging Topics in Computing, vol. 2, no.3, pp. 267-279, 2014.

[7] S. Zhang, M. Zong and D. Cheng, ”Learning k for KNN Classification”, ACM Transactions on Intelligent Systems and Technology, vol. 8, no. 3, pp. 43:1-19, 2017.

[8] K. Niu, F. Zhao and S. Zhang, ”A Fast Classification Algorithm for Big Data Based on KNN”, Journal of Applied Sciences, vol. 13,no. 12, pp. 2208-2212, 2013.

[9] Bifet, J. Read, B. Pfahringer and G. Holmes, ”Efficient Data Stream Classification via Probabilistic Adaptive Windows”, in Proc. 28th Annual ACM Symposium on Applied Computing, 2013, pp. 801-806.

[10] S. S. Labib, ”A Comparative Study to Classify Big Data Using fuzzy Techniques”, in Proc. 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA), 2016.

[11] M. El Bakry, S. Safwat and O. Hegazy, ”A Mapreduce Fuzzy technique of Big Data Classification, in Proc. SAI Computing Conference 2016, pp. 118-128.

[12] B. Quost and T. Denoeux, ”Clustering and Classification of fuzzy data using the fuzzy EM algorithm”, Fuzzy Sets and Systems, vol. 286, pp. 134-156, 2016.

[13] Z. Deng, X. Zhu, D. Cheng, M. Zong and S. Zhang, ”Efficient kNN classification algorithm for big data”, Neurocomputing, vol.195, pp. 143-148, 2016.

[14] S. Zhang, D. Cheng, M. Zong and L. Gao, ”Self representation nearest neighbour search for classification”, Neurocomputing, vol.195, pp. 137-142, 2016

[15] G. Song, J. Rochas, L. El Beze, F. Huet and F. Magoules, ”K Nearest Neighbour Joins for Big Data on MapReduce:A Theoretical and Experimental Analysis”, IEEE Trans. on Knowledge and Data Engineering, vol. 28, no. 9, pp. 2376-2392, 2016.

[16] J. Maillo, S. Ramirez, I. Triguero and F. Herrera, ”kNN-IS: An Iterative Spark-based design of the k-Nearest Neighbours classifier for big data”, Knowledge-Based Systems, vol. 117, pp. 3-15, 2017.

[17] T.Tulgar, A.haydar and İ.Erşan, "Data Distribution Aware Classification Algorithm based on K-Means", International Journal of Advanced Computer Science and Applications, vol. 8, no.9, pp.328–334, September 2017.

[18] J. Gosling, B. Joy, G. Steele, G. Bracha, A. Buckley, (2017,AUG 01). The Java Language Specification-Java SE 8 Edition Online. Available: https://docs.oracle.com/javase/specs/jls/se8/html/index.html

[19] UCI Center for Machine Learning and Intelligent Systems, (2017, AUG 01). UC Irvine Machine Learning RepositoryOnline.Available: https://archive.ics.uci.edu/ml/