Using Center Representation and Variance Effect on K-NN Classification

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

Abstract— **The K-Nearest Neighbor classifier is a well-known and widely applied method in data mining applications. Nevertheless, its high computation and memory usage cost makes the classical K-NN not feasible for today's Big Data analysis applications. To overcome the cost drawbacks of the known data mining methods, several distributed environment alternatives have emerged. Recently, several K-NN based classification algorithms have been proposed which are distributed methods suitable for distributed computing environments and applicable for emerging data analysis needs. In this work, a new CR-KNN algorithm is proposed, which improves the classification accuracy performance of the well-known K-Nearest Neighbor (K-NN) algorithm by benefiting from the center representation of the instances belonging to different data classes. The proposed algorithm relies on the data class representations which are the closest to the test instance. The CR-KNN algorithm was tested using several real-datasets belonging to different application areas. The performance results acquired after extensive experiments are presented in this paper and they prove that the proposed CR-KNN algorithm is a competitive alternative to other studies recently proposed in the literature.**

Keywords- Classification, K-Nearest Neighbor, Center Representation, Variance

I. INTRODUCTION

In the current information age, businesses tend to base their policies and forecasting on the analysis and processing continuously growing volumes of data retrieved from many different online streaming sources [1].

To achieve precise business decisions, data coming from different environments (e.g. different social media tools, data warehouses, cloud storages etc.) need to be analyzed using efficient data mining algorithms. Analyzing data coming from high number 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].

One of the crucial data mining tasks is classification. Classification, which is 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 lack the low classification delay performance, which is vital to cope with the high velocity data streams of big data. To tackle with this timing requirement of the modern data classification, several distributed environments have been tested and used by different researchers. One of these distributed environments is 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.

In this paper, a new K Nearest Neighbor classification algorithm, which will be referred as the Center Representation KNN algorithm (CR-KNN) hereafter, is proposed. The CR-KNN algorithm tries to improve the classification accuracy performance of the classical K-NN algorithm.

The main idea of the CR-KNN algorithm is to base the classification decision of the classical K-NN algorithm on the class representations by calculating the centroids of the training instances belonging to the same classes among the K closest instances detected by the K-NN algorithm. In other words, the classical majority voting decision step of the classical K-NN is refined by a better class representation based classification decision, which is also computationally efficient. Furthermore, the CR-KNN is further improved by introducing the variance effect to the distance measurement so that the algorithm can adapt to the varying data distributions of different datasets.

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

II. THE PROPOSED CR-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.

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 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].

Hence, it can be concluded that to become a classification algorithm suitable for modern data analysis needs, the K-NN's classification accuracy performance should be improved. Taking into account these needs, the CR-KNN algorithm is proposed.

B. *The proposed CR-KNN Algorithm*

The proposed CR-KNN algorithm improves the classical K-NN algorithm with contributions in the distance measurement and classification decision steps.

The classical Euclidean Distance Calculation is given in (1).

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

Like classical K-NN, the first step of the CR-KNN algorithm is calculating the distances between the testing data instance and all training data instances using (1).

After computing all the distances, the CR-KNN finds the K nearest neighbors with the minimum distances to the tested data instance. At this point, unlike classical KNN, the proposed CR-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 CR-KNN finds the centers of the training data instances for each detected class. To clarify the introduced idea, consider the following simplified example, as the detected K nearest neighbors:

<ts9, <tr3, 0.11, A>

<ts9, <tr11, 0.14, B>

<ts9, <tr27, 0.15, B>

<ts9, <tr23, 0.12, C>

<ts9, <tr42, 0.12, A>

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 listed training instances of each of the listed classes will be used to represent each of the classes by their means (in other words the centers).

The mean μ of a class with N instances from the K nearest neighbors are calculated using (2). In other words, centroids are found that represent classes.

$$\mu = \frac{1}{N}\sum_{i=1}^{N} tr_i \quad (2)$$

Upon calculating the means, the CR-KNN calculates the minimum of the distances of the testing instance to be classified to the calculated means using (3) for all classes j. As the classification decision which is the main contribution of the CR-KNN, the algorithm relies on the minimum T of all centroids j rather than classical majority voting.

$$T = \min_j\left(\frac{\|Tsi-\mu j\|^2}{{\delta_j}^2}\right) \quad (3)$$

The second contribution, the variance effect can also be seen in (3). The variance ${\delta_j}^2$ is defined in (4).

$${\delta_j}^2 = \frac{1}{N}\sum_{i=1}^{N}\left(tri-\mu_j\right)^2 \quad (4)$$

Calculating the distance of the testing instance to centroid contribution of the CR-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.

Nevertheless, the same class A has an instance, which is further away from the class B instances to the test instance. 9. To give equal chances to classes in the classification decision, CR-KNN proposes to represent the classes A, B and C of the K nearest neighbors by centroids and base the classification decision on the distance of the test instance to the centroids of the classes rather than relying on the individual data members. In this way, if class B has a centroid closer to the test instance, the CR-KNN can correctly decide that the test instance should belong to class B.

Furthermore, CR-KNN addresses another classification problem that is the fact that the different datasets belonging to different application areas may have data distributions that are not easy to deal with. The main difficulty encountered in different datasets is a data distribution which includes radical data instances belonging to different classes may have close proximity to other class instances. In such cases, false classification decisions can be easily produced by the classical K-NN algorithm.

To compensate the distance calculation based decisions of K-NN, the proposed CR-KNN algorithm includes the effect of how far the instances are spread around the centroids to the classification decision by including the variance to the decision.

This compensation is achieved by dividing the calculated distances to the centroids to the variance. This results in concluding that a smaller variance will produce a stronger membership strength versus a greater variance will produce a weaker membership strength.

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

III. EXPERIMENTAL SETTING AND THE RELUTS

The proposed CR-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 CR-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] 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 SR-KNN, which was 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 [14] paper show better performance when compared to these three algorithms.

The KMEANS-MOD proposed in [17] on the other hand, is an improved KMEANS clustering based algorithm which also 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 effect of adding variance effect to the distance calculation of the CR-KNN for different K values.

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

	CR-KNN without Variance Effect			CR-KNN with Variance Effect		
	K=5	K=7	K=9	K=5	K=7	K=9
pendigits	0.9614	0.9542	0.9459	**0.9614**	0.9542	0.9459
satimage	0.8815	0.8555	0.8310	**0.8815**	0.8555	0.8310
ionosphere	0.8729	0.7647	0.6764	**0.8975**	0.7853	0.7353
wdbc	0.9107	0.8928	0.8928	**0.9685**	0.9285	0.9285

By investigating the results, it can be observed that the variance effect improves the classification accuracy performance of the CR-KNN when the dataset used in the classification shows outlying instances in the data distribution of the class instances like explained in section II. What is more, the results also show that the variance addition to the distance calculation does not have a degrading effect on the tested datasets, which have a clearer data separation among the class instances.

Furthermore, it can be observed that, reasonably small K values are enough to achieve improved classification accuracy performance from the proposed CR-KNN algorithm.

The proposed CR-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 CR-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	CR-KNN
pendigits	0.9770	0.9078	0.9452	0.9614
satimage	0.8980	0.9065	0.8806	0.8815
ionosphere	0.9123	0.6286	0.8971	0.8975
wdbc	0.9628	0.6548	0.9650	0.9685

The comparative results presented in Table III illustrate that the proposed CR-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 CR-KNN shows better classification accuracy performance for all datasets proving that the variance effect and the centroid representation for the data classes provide the expected improvement on the classification accuracy.

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

IV. 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 two improvement contributions.

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 Repository Online. Available: https://archive.ics.uci.edu/ml/