A Novel Adaptive K Nearest Neighbor Algorithm

Hamid Nasiri, Saeed Shiry Ghidary, Mohammad Mehdi Ebadzadeh
Computer Engineering and Information Technology Department
Amirkabir University of Technology (Tehran Polytechnic)
Tehran, Iran
h.nasiri@aut.ac.ir, shiry@aut.ac.ir, ebadzadeh@aut.ac.ir

***Abstract*— Classification is a broad ranging research field and different algorithms proposed in this area, one of which is K nearest neighbor (KNN) algorithm. This algorithm has a simple structure and easy implementation. Its performance depends on three main factors including similarity measure for voting, distance function and appropriate value for the parameter K among which the value of K is particularly significant, So that if it is not correctly selected, algorithm performance would remarkably reduce. We proposed a novel method for adaptive selection of parameter k in this paper. In this method, an optimal K-value for each training instance is obtained and used to classify a test instance by KNN algorithm. Evaluation tests on standard datasets and comparing obtained results with conventional methods show that the presented method has an acceptable performance compared to other methods and improves classification accuracy as well.**

Keywords*— *K Nearest Neighbor Algorithm; Adaptive KNN Algorithm; Nearest Neighbor Classification; Pattern Classification

I. INTRODUCTION

In Data Mining we analyze data in different dimensions and we find relationships between them. Nowadays the first priority of companies is the customer acquisition and retention. Therefore, data mining is a popular research area among them [1]. There are different data mining techniques that classification and clustering are among them. Classification is an approach used to classify a test instance and predict its class label. Various methods of classification have been proposed. Some of those techniques include Decision Tree, Naïve Bayes, KNN, Neural Networks, etc [2].

KNN has a simple structure, is easy to implement [3] and its classification error is limited to twice of the classification error in Naïve Bayes method [4]. Such characteristics have led us to use KNN algorithm widely in data mining and pattern recognition. This algorithm was first proposed by Cover and Hart [3]. KNN is based on a distance function which measures the similarity or difference between two data points. Generally, standard Euclidean distance, d(x, y), between two instances x and y is used as distance function. Standard Euclidean distance is defined in (1).

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

As seen in (2), for a given instance x, KNN finds out the class of x's K nearest neighbors and assign most common of it to x [5].

$$c(x)=\arg\max_{c \in C} \sum_{i=1}^{k} \delta\left(c,c(y_i)\right) \quad (2)$$

In (2) y_i,i=1,2,…,K are K nearest neighbors of test instance x, K is the parameter of algorithm (K neighbors are considered) and δ (c, c(yi)) = 1, if c = c(y_i) and δ (c, c(y_i)) = 0, otherwise.

Performance of KNN algorithm depends on three main factors: I. Similarity measure for voting, II. Distance function, III. Appropriate value of parameter K [6]. In the past years, a myriad of studies have been carried out to improve KNN algorithm based on the aforementioned factors. In this study, a novel method is proposed for adaptive selection of K. Evaluation tests on standard benchmarks show that this technique has an acceptable performance compared to other methods.

Rest of the paper is organized as follow: In Section II we discuss the related works, in Section III we describe the proposed method. Experimental results are presented in Section IV and finally the paper is concluded in Section V.

II. RELATED WORKS

A myriad of studies have been previously done to improve the performance of KNN algorithm some of which are reviewed here. In 1978, Bailey and Jain proposed a KNN algorithm with weighting factor. In this algorithm, training data were related to a weight factor which was determined considering the distance between training instance and test instance. The result of this weight factor was elimination of redundant patterns [7]. In 1983, Jowik proposed the concept of fuzzy KNN for the first time. In this algorithm fuzzy membership values for unknown data points were considered in ranges of [0, 1], [8]. Keller and Gray introduced the meaning of fuzzy sets in 1985 and proposed fuzzy KNN algorithm [9]. In 2003, Guo and Wang proposed a KNN model in which a new model of data was formed by a set of representatives of training instances, which was smaller than complete training set. That model was used to classify new test instances. However it was not efficient for large datasets and had low classification accuracy [10]. Therefore, in 2012, Guo et al presented an improved version of the algorithm called (efficient) e-KNN model. The latter performed appropriately on large datasets as well [11]. In 2008, Lu et al proposed a KNN algorithm based on clustering. In such an algorithm they categorized test dataset into isolated points and different clusters. Then to classify a new test instance, a representative was selected from each cluster. While representative sets were smaller than test dataset, classification speed was improved to a large extent [12].

Many other studies have been carried out to improve KNN algorithm focusing on how to choose K for the KNN algorithm. In 2003, Hand and Vinciotti showed that how it could be possible to select K when the classes were unbalanced [13]. In 2010, Hamerly and Speegle proposed an efficient method based on cross validation to select K [14]. In 2007, Ougiaroglou et al used certain heuristics for dynamic selection of K to enhance classification accuracy [15]. In 2014, Bhattacharya et al

proposed a novel non-parametric method to estimate K. In this method first a hyper-sphere was constructed around each test instance to obtain the local distribution of the training instances. Then such information was used to classify the data [16]. Time complexity is one of the issues in fuzzy KNN algorithm, as the algorithm calculates membership at the classification phase. Therefore, in 2015, Taneja et al proposed MFZ-KNN which was a modified version of fuzzy based KNN. MFZ-KNN improved time complexity to a large extent [2].

In this paper we proposed an adaptive method for selecting K. As mentioned before, the value of K plays a significant role in the accuracy of KNN algorithm. In this method, an optimal K for each training instance is obtained and for each new test instance we set K as optimal K of its nearest neighbor. In addition, if it is not possible to determine such an optimal K, we average optimal K of its nearest neighbors and use it as parameter K of KNN algorithm.

III. Proposed Method

Although KNN algorithm has many advantages compared to other classification methods and normally presents an acceptable performance, selection of K is of great importance and if that parameter is not appropriately selected, the performance of the algorithm would remarkably reduce. Conventional choices for K are 1, 3, 5 and 7 [17] any of which are appropriate for a specific problem and might not be suitable for other problems. Therefore, in current study, a method has been proposed in which selection of K could be adaptive based on the problem conditions. Evaluation tests on UCI standard datasets [18] show that current method has an acceptable performance compared to other methods and it improves classification accuracy of KNN algorithm.

In this technique, an optimal K-value is obtained for each training data which is called Koptimal. For each test instance according to Koptimal of its nearest neighbor, we consider Koptimal neighbors of this test instance and perform classification by these points. In other word we carry out KNN algorithm for test instance with K equal to Koptimal of its nearest neighbor. Koptimal calculation procedure for each training instance is described below:

To calculate Koptimal for each training instance, first its nearest neighbors are obtained based on Euclidean distance and then classification of training instance is done based on conventional KNN algorithm with K = 1. Finally, predicted class is compared with the true class of that instance. If the training instance is classified correctly, then Koptimal is set to 1 for that training instance; otherwise K-value would be considered 2 and aforementioned steps should be repeated. This time, if classification is done correctly, then Koptimal is set to 2. The same procedure is continued until we reach to K = 9. If none of the classifications are correct, then Koptimal would be considered -1 and all steps should be repeated for each training instance and Koptimal would be calculated for all of them. At the second step, Koptimal for the training instances which their Koptimal has not been obtained, is determined by the following procedure:

For each training instance, x, first the nearest neighbor is obtained. If Koptimal of this training instance is unknown (i.e. Koptimal = -1), the Koptimal of x would be considered 9, otherwise we consider Koptimal neighbors of x, get their Koptimals and use average of their Koptimals as x's Koptimal. Considering the proximity of these points, their behavior is probably similar. Thus, their appropriate Koptimal would also be closed to each other. It should be kept in mind that Koptimal might not be specified for some of the nearest neighbors. In that case, data with unknown Koptimal is removed from averaging process and if Koptimal of all neighbors are unknown, Koptimal of x would be considered equal to 9. When the values of Koptimal are obtained for all training data, then classification is carried out by the following procedure:

First the nearest neighbor of that test instance from training set is obtained. Then the value of Koptimal for this nearest neighbor is extracted. Finally, KNN algorithm with parameter K = Koptimal is carried out and test instance is classified.

IV. Experimental Results

To analyze the performance of the presented method in this work, 12 datasets from UCI Machine Learning dataset [18] were used. Details of these datasets are presented in table 1. To evaluate proposed algorithm using each dataset from table (1), we used 70% of the data as training set and 30% of the remained data, were considered as test set. Train and test set were selected randomly from whole data set. Then classification accuracy was calculated on test sets and such a procedure was repeated 10 times for each dataset and the average and standard deviation of classification accuracy were reported.

Obtained results of the performance of the proposed algorithm (classification accuracy) and obtained results of conventional KNN method with K-values equal to 1, 3, 5 and 7 as well as Bhattacharya technique [16] are reported in table 2.

As seen in table 2, the proposed method in comparison to KNN algorithm with K=1, K=3, K=5, K=7 and Bhattacharya algorithm [16] has better performance in 12, 10, 9, 10 and 8 datasets respectively. Also, this technique has the highest average accuracy (85.23) among all compared methods.

TABLE I. TESTED DATASETS FROM UCI

Dataset	Number of Instances	Number of Features	Number of Classes
Iris	150	4	3
Wine	178	13	3
Glass Identification	214	10	7
Pima-Diabetes	768	8	2
Connectionist Bench (Sonar, Mines vs. Rocks)	208	60	2
Ionosphere	351	34	2
Statlog (Vehicle Silhouettes)	846	18	4
Breast Cancer Wisconsin (Diagnostic)	569	32	2
SPECTF Heart	267	44	2
Musk (Version 1)	476	166	2
Breast Tissue	106	9	6
Ecoli	336	8	8

TABLE II. COMPARISION OF PROPOSED KNN ALGORITHM WITH OTHER METHODS

Accuracy using Euclidean Distance						
Dataset	**K = 1**	**K = 3**	**K = 5**	**K = 7**	**Bhattacharya Method [16]**	**Proposed Method**
	Mean (S.D.)	**Mean (S.D.)**	**Mean (S.D.)**	**Mean (S.D.)**	**Mean (S.D.)**	**Mean (S.D.)**
Iris	94.00 (4.58)	92.28 (5.86)	94.81 (5.77)	95.50 (5.29)	94.94 (5.30)	**96.67 (2.40)**
Wine	95.34 (5.11)	95.67 (5.01)	96.85 (4.04)	96.47 (4.43)	97.46 (3.83)	**97.88 (2.59)**
Glass Identification	68.81 (8.56)	70.21 (8.08)	68.27 (9.40)	64.61 (9.74)	69.40 (8.38)	**88.44 (1.68)**
Pima-Diabetes	70.82 (5.27)	73.72 (4.43)	**74.00 (4.69)**	73.76 (4.52)	73.37 (4.39)	73.13 (2.05)
Connectionist Bench (Sonar, Mines vs. Rocks)	86.42 (8.04)	84.52 (7.96)	81.01 (8.89)	79.15 (9.07)	86.16 (7.68)	**87.58 (3.49)**
Ionosphere	86.77 (5.83)	84.74 (5.66)	84.09 (6.40)	83.23 (6.47)	87.77 (5.02)	**88.19 (2.70)**
Statlog (Vehicle Silhouettes)	69.88 (4.20)	71.48 (4.01)	71.87 (3.88)	71.73 (4.39)	**72.76 (3.70)**	71.85 (2.51)
Breast Cancer Wisconsin (Diagnostic)	95.13 (2.53)	96.47 (2.47)	96.85 (2.30)	96.64 (2.33)	96.87 (2.10)	**97.10 (1.25)**
SPECTF Heart	70.47 (8.23)	71.56 (8.33)	72.79 (8.95)	74.36 (8.26)	73.94 (7.46)	**76.12 (3.93)**
Musk (Version 1)	88.53 (4.69)	89.08 (4.85)	88.84 (4.90)	86.62 (5.31)	**92.17 (3.87)**	89.58 (3.34)
Breast Tissue	68.25 (14.3)	69.08 (14.8)	70.70 (14.3)	62.65 (13.79)	71.61 (14.11)	**72.19 (6.82)**
Ecoli	80.79 (6.78)	84.08 (6.38)	85.83 (6.40)	**86.54 (6.28)**	85.17 (6.28)	84.06 (2.66)
Overall Mean & Standard Deviation	81.27 (6.51)	82.07 (6.49)	82.16 (6.66)	80.49 (6.66)	83.47 (6.01)	**85.23 (2.95)**

In addition, the presented method has remarkable performance compared to MFZ-KNN [2] which has only reported its results on Wine dataset.

V. CONCLUSION AND FUTURE WORKS

In this study, a KNN algorithm with adaptive selection of parameter K was proposed in which an optimal K-value was obtained for each training instance and it was used to classify each new test instance. To evaluate the proposed method, standard UCI dataset was used and the proposed algorithm was tested on 12 various datasets. Obtained results show that the aforementioned method is efficient and it improves classification accuracy of KNN algorithm.

For the future works, testing the proposed algorithm on more datasets is proposed (especially datasets with missing values and datasets with high dimensions). Moreover, it is possible to test the presented method on real world problems and analyze its performance.

REFERENCES

[1] J. Han and M. Kamber, Data Mining: Concepts and Techniques, 2nd ed. Elsevier publications.

[2] S. Taneja, C. Gupta, S. Aggarwal and V. Jindal, "MFZ-KNN — A modified fuzzy based K nearest neighbor algorithm," International Conference on Cognitive Computing and Information Processing (CCIP), Noida, 2015, pp. 1-5.

[3] T.M. Cover and P.E. Hart, "Nearest neighbor pattern classification," IEEE Transactions on Information Theory, vol. 13, 1967, pp. 21-27.

[4] K. Fukunaga and L. D. Hostetler, "k-nearest-neighbor Bayes-risk estimation," IEEE Transactions on Information Theory, vol. 21, 1975 pp. 285–293.

[5] L. Jiang, "Survey of Improving K-Nearest-Neighbor for Classification," IEEE Fourth International Conference Fuzzy Systems and Knowledge Discovery, vol.1, 2007, pp. 679–683.

[6] G. Bhattacharya, K. Ghosh, and A. S. Chowdhury, "An affinity-based new local distance function and similarity measure for kNN Algorithm," Pattern Recognition Letters, vol. 33, 2012, pp. 356–363.

[7] T. Bailey and A. K. Jain, "A note on Distance weighted k-nearest neighbor rules," IEEE Transactions on Systems, Man and Cybernetics, vol. 8, 1978, pp.311-313.

[8] A. Jówik, "A learning scheme for a fuzzy k-NN rule," Pattern Recognition Letters, vol. 1, no. 5-6, 1983, pp.287–289.

[9] J. M. Keller and M. R. Gray, "A Fuzzy K Nearest Neighbor Algorithm," IEEE Transactions on Systems, Man and Cybernetics, vol. 15, 1985, pp. 580-585.

[10] G. Guo and H. Wang, "KNN model based approach in classification," Proc. of On The Move to Meaningful Internet Systems, CooIS, DOA, and ODBASE Lecture Notes in Computer Science, Springer, vol. 2888, 2003, pp. 986-996.

[11] L. Chen, G. Guo, and S. Wang, "Nearest neighbor classification by partially fuzzy clustering," IEEE conference on Advanced Information Networking and Applications workshop, 2012, pp. 789- 794.

[12] J. Lu, L. Wang, J. Lu, and Q. Sunl, "Research and application on kNN method based on cluster before classification," IEEE conference on Machine Learning and Cybernetics, vol. 1, 2008, pp. 12-15.

[13] D. J. Hand, and V. Vinciotti, "Choosing k for two-class nearest neighbor classifiers with unbalanced classes," Pattern Recognition Letters, vol. 24, 2003, pp. 1555–1562.

[14] G. Hamerly and G. Speegle, "Efficient Model Selection for Large-Scale Nearest-Neighbor Data Mining," Proc. of 27th British national conference on Data Security and Security Data (BNCOD), 2010, pp. 37-54.

[15] S. Ougiaroglou, A. Nanopoulos, A. Papadopoulos, N. Apostolos, M. Yannis, and T. Welzer-Druzovec, "Adaptive k-Nearest Neighbor Classification Based on a Dynamic Number of Nearest Neighbors," Proc. of the 11th East European conference on Advances in Databases and Information Systems (ADIBS), 2007, pp. 66-82.

[16] G. Bhattacharya, K. Ghosh and A. S. Chowdhury, "Test Point Specific k Estimation for kNN Classifier," 22nd International Conference on Pattern Recognition (ICPR), Stockholm, 2014, pp. 1478-1483.

[17] K. Kozak, M. Kozak, and K. Stapor, (2005). "Weighted k-Nearest-Neighbor Techniques for High Throughput Screening Data," International Journal of Biological and Life Sciences, vol. 1, 2005, pp. 155–160.

[18] "UCI Repository", Retrieved on 16 April 2016, at "https://archive.ics.uci.edu/ml/datasets.html".