Celal Bayar University Journal of Science

A Weighted Similarity Measure for k-Nearest Neighbors Algorithm

Bergen Karabulut[1]*, Güvenç Arslan[2], Halil Murat Ünver[3]

[1, 3] Department of Computer Engineering, Kırıkkale University, Kırıkkale, Turkey
[2] Department of Statistics, Kırıkkale University, Kırıkkale, Turkey
*bergen.karabulut@tubitak.gov.tr

Received: 11 September 2019
Accepted: 12 December 2019
DOI: 10.18466/cbayarfbe. 618964

Abstract

One of the most important problems in machine learning, which has gained importance in recent years, is classification. The k-nearest neighbors (kNN) algorithm is widely used in classification problem because it is a simple and effective method. However, there are several factors affecting the performance of kNN algorithm. One of them is determining an appropriate proximity (distance or similarity) measure. Although the Euclidean distance is often used as a proximity measure in the application of the kNN, studies show that the use of different proximity measures can improve the performance of the kNN. In this study, we propose the Weighted Similarity k-Nearest Neighbors algorithm (WS-kNN) which use a weighted similarity as proximity measure in the kNN algorithm. Firstly, it calculates the weight of each attribute and similarity between the instances in the dataset. And then, it weights similarities by attribute weights and creates a weighted similarity matrix to use as proximity measure. The proposed algorithm is compared with the classical kNN method based on the Euclidean distance. To verify the performance of our algorithm, experiments are made on 10 different real-life datasets from the UCI (UC Irvine Machine Learning Repository) by classification accuracy. Experimental results show that the proposed WS-kNN algorithm can achieve comparative classification accuracy. For some datasets, this new algorithm gives highly good results.

Keywords: Classification, Feature Weighting, k-Nearest Neighbors, Weighted Similarity.

1. Introduction

In parallel with technological developments, the importance of the machine learning and the disciplines related to it has increased. In addition, increasing the importance of large-scale data in all areas of life has revealed new demands based on machine learning algorithms [1]. One of the basic learning techniques in machine learning is supervised learning. In supervised learning, a training set consisting of a set of instances $\{\mathbf{x}_n\}_{n=1}^{N}$ with their targets $\{t_n\}_{n=1}^{N}$ is given. It is expected that a model will be developed by using the instances given with the training set and their targets. It is aimed to predict the target t accurately for an input value $\mathbf{x}$, which has not been encountered before, with the help of the model. One of the main problems in machine learning is classification as a supervised learning technique [2]. Classification is the process of placing an object in a predefined class by using attributes [3]. Class labels are treated as targets in the classification problem.

There are many different learning methods in the supervised learning. One of these methods is instance-based learning, also called lazy learning [4]. The classical example of an instance-based learning method is the k-nearest neighbors (k-NN) classification algorithm [5]. In the naive implementation of the nearest neighbors rule, to classify each data point it is necessary to store all the data points previously classified and compare each data point to each stored point [6]. The k-NN method is simple but has proved to be very efficient and effective for solving various classification problems in real life [7]. It has been used in practical applications such as gear crack level identification [8], heart disease prediction [9], diagnosis of breast cancer [10], hand gesture recognition [11], text classification [12]. On the other hand, the new studies have been carried out to improve the method.

The k-NN method requires two parameters. The first parameter is the proximity measure that defines the closest instances. The second one is the k variable representing an upper limit for the number of instances

that will be handled [13]. Distance metrics are generally used as proximity measure. The most widely used distance measure in the k-NN is the Euclidean distance [7]. However, there are many distance metrics in the literature and some studies showed that the distance metric used in the k-NN method affects the performance of it. For example, Mulak and Talhar [14] applied Euclidean, Chebychev and Manhattan distance functions in the k-NN method and they reported that Manhattan distance provided higher performance. In another study, Hu et al. [15] investigated the effect of distance function on the performance of k-NN method by using different medical datasets. They used Euclidean, cosine, Chi square and Makowski distance functions and stated that chi square was the best option.
Depending on the learning process applied in machine learning, each pattern can be represented as a set of features/attributes; however, the extent of their effects is not in equal importance [16]. In many cases, some attributes are more discriminative, while others may be less irrelevant [17]. For this, one of the commonly used methods is feature weighting. Feature weighting assigns real values (instead of 0 or 1) to features that define the relevance to a learning problem [18]. These values can be used to weight similarities of instances depending on the attributes.

One of the most important factors affecting the performance of the kNN is the distance function. Several studies have been carried out to determine the appropriate distance function. In these studies, the effect of distance function on classification performance has been investigated by using the existing *distance* functions with kNN. In addition, there are studies that suggest a new distance metric and use it with k-NN to improve the performance of kNN.

Jiado et al. [17], proposed a new evidential k-nearest neighbors classification with weighted attributes (WEK-NN) to overcome the limitations of evidential kNN method. In WEK-NN, the class-conditional weighted Euclidean distance function was used to assess the similarity of test samples with the stored training samples. They used both the heuristic rule and the parameter optimization procedure to determine attribute weights. They demonstrated the superiority of the WEK-NN method over several classical kNN methods as a result of their experiments with simulated and real datasets.

Hassanat [19] introduced a new similarity measure to be used in general work, including supervised learning. The k-NN method was used to test the viability of the proposed similarity measure for different applications and classification was performed with test samples from several real data sets. According to the experimental results, the proposed metric is a promising distance measure for the k-NN classifier compared to some other well-known metrics.

Alkasassbeh et al. [20] used Hassanat distance measure to improve performance of some nearest neighbors classifiers. As a result of this study, they stated that this distance measure shows its superiority over traditional and most used distances such as Manhattan and Euclidean distance. They also proved that this distance measure was invariant against data scaling, noise and outliers.

Mulak and Talhar [14] analyzed and compared Euclidean, Chebychev and Manhattan distance functions using k-nearest neighbors. They compared these distance measures with regard to accuracy, specificity, sensitivity, false positive rate and false negative rate using KDD (knowledge discovery and data mining) dataset. It was observed that Manhattan distance had better results than others. They also stated that the performance of the Euclidean distance was lower than the Chebychev distance.

Chamboon et al. [21] examined the performance of the kNN classification method by using 11 different distance measures; Euclidean, Standardized Euclidean, Mahalanobis, City block, Minkowski, Chebychev, Cosine, Correlation, Hamming, Jaccard, and Spearman. They performed a set of experimentations performed on 8 synthetic datasets that have different kinds of distribution. They concluded that Hamming and Jaccard techniques were affected by the ratio of members in each class, while the other techniques were not affected by such phenomenon. They also stated the highest accuracy on classify data with k-Nearest Neighbors was obtained from City-block, Chebychev, Euclidean, Mahalanobis, Minkowski, and Standardized Euclidean techniques.

Hu et al. [15] examined whether the distance function effect the kNN performance over different medical data sets. They used four different distance measures including Euclidean, Cosine, Chi square and, Minkowsky and their experiments were based on three different types of medical datasets containing categorical, numerical and mixed types of data. As a result of their study, they showed that the selected distance function can affect the classification accuracy of the kNN classifier. In addition, they stated that k-NN method based on Chi square distance function gives the best results for medical datasets including numerical, categorical and mixed types of data.

Prasath et al. [7] conducted a comprehensive review about the effect of distance measures on the performance of the k-nearest neighbors classifier. In addition, they attempted to find the most appropriate distance measure that can be used with kNN in general. They evaluated the performance of the kNN using many distance measures on clean and noisy datasets. Various results were obtained from the study. They stated that the performance of the kNN classifier was dependent

significantly on the distance used, and there were big gaps between the performance of different distances. They also stated that there is no optimal distance measure appropriate for all datasets and each dataset supported a certain distance measure.
Rodrigues [13] proposed a new distance that combines Minkowski and Chebychev distances. To evaluate the efficiency of the proposed distance using kNN an experiment was performed using 33 datasets from the UCI. In this experiment, 15 different distance measures were used, and the k value ranged from 1 to 200. As a result of the study, it is stated that the combination of Minkowski and Chebychev distances provides an efficient distance metric in terms of processing times and accuracy.

This work proposes a new similarity measure called as weighted similarity as proximity measure. It is motivated by the weighted similarity expressions defined for the numerical attributes that are used by Kayaalp and Arslan [22, 23]. To evaluate the performance of this measure, it was used in the k-nearest neighbors algorithm as proximity measure. This new version of kNN was called as weighted similarity k-nearest neighbors (WS-kNN) algorithm. An experimental analysis was conducted on some real-world datasets.

2. Materials and Methods

This section presents basic notation and the formulations of similarity function and attribute weighting definitions. More details about notions may be found in [24].

2.1. Definitions and Notation

Consider a dataset X consisting of data points; $X = \{x_1, x_2, \cdots, x_n\}$ (n: number of instances in the dataset). Each data point is defined by m attributes in the attributes set; $A = \{a_1, a_2, \cdots, a_m\}$. The value of a data point x_i in the dataset for an attribute a_j is represented by $f(x_i, a_j)$.

Similarity Function

Definition1. The *similarity* value between instances x_i and x_j with respect to an attribute $a \in A$ is defined as:

$$sim_a(x_i, x_j) = 1 - \frac{|f(x_i,a) - f(x_j,a)|}{max(a) - min(a)} \quad \textbf{(2.1)}$$

Definition2. The *weighted similarity* value between instances x_i and x_j with respect to all attributes $a = 1, 2, 3, \ldots, m$ is defined as:

$$SIM_A(x_i, x_j) = \sum_{a \in A} w_a sim_a(x_i, x_j) \quad \textbf{(2.2)}$$

where w_a corresponds to the attribute $a \in A$.

Attribute Weighting

Suppose that there are t classes in the dataset. For $a \in C$, let $A_i(a) = \{x_k \in X: \min(C_i(a)) \leq f(x_k, a) \leq \max(C_i(a))\}$

where $C_i(a)$ is the set of values for attribute a belonging to class i, $1 \leq i \leq$. Denoting by

$$B_j(a) = A_j(a) - \bigcup_{\substack{i=1 \\ (i \neq j)}}^{t} A_i(a) \quad \textbf{(2.3)}$$

the weights w_a is defined by

$$w_a = \frac{\bigcup_{i=1}^{t} |B_i(a)|}{n} \quad \textbf{(2.4)}$$

where $|B_i(a)|$ denotes the number of elements in set B_i. Weights are normalized to see the impact of that attribute with respect to each class. Therefore,

$$w_a^* = \frac{w_a}{\sum_a w_a} \quad \textbf{(2.5)}$$

2.2. Datasets

In the experimental analysis of the study, 10 real world datasets from UCI (UC Irvine Machine Learning Repository) [https://archive.ics.uci.edu/ml/index.php] were used. The datasets used contain only numerical type of data and they are summarized in Table 1.

Table1. Summary of data sets, where #NI: number of instances; #NN: number of numerical attributes; #NC: number of classes; N/A: missing value.

#	Dataset	#NI	#NN	#NC	N/A	Type
1	Iris	150	4	2	No	Numerical
2	Wine	178	13	3	No	Numerical
3	Glass	214	10	7	No	Numerical
4	Sonar	208	60	2	No	Numerical
5	Vehicle	846	18	4	No	Numerical
6	Ionosphere	351	34	2	No	Numerical
7	Transfusion	748	4	2	No	Numerical
8	*Escherichia coli*	336	7	8	No	Numerical
9	Haberman	306	3	2	No	Numerical
10	Breast Cancer Wisconsin	699	9	2	Yes	Numerical

Celal Bayar University Journal of Science
Volume 15, Issue 4, 2019 p 393-400
Doi: 10.18466/cbayarfbe.618964

As shown in Table 1 only the Breast-cancer dataset includes missing values. This dataset contains 2.288984% missing values of an attribute. Before the classification process, imputation was performed for this dataset and the missing values were imputed. The mean imputation operation was applied because the dataset is numerical.

The proposed method was implemented with the R programming language on the R studio platform. To test the performance of the proposed method accuracy measure was used.

3. Results and Discussion

This section presents proposed Weighted Similarity k-Nearest Neighbors (WS-kNN) algorithm and experimental analysis of it.

WS-kNN Algorithm

Input:

$\boldsymbol{X} = \{\boldsymbol{x_i} \epsilon \mathbb{R}^{\boldsymbol{d}}\}_{\boldsymbol{i=1}}^{\boldsymbol{n}}$: the training set with n training instances.

$\boldsymbol{k}$: the neighbors size.

$\boldsymbol{z} = \{\mathbf{x}', \boldsymbol{y}'\}$: the given query sample.

$\boldsymbol{A} = \{\boldsymbol{a_1}, \boldsymbol{a_2}, \boldsymbol{a_3}, \dots, \boldsymbol{a_m}\}$: the feature set with m classes.

$\boldsymbol{w}$: attribute weight vector.

Step 1: Compute the weight of each attribute

for all $\boldsymbol{a} \subset \boldsymbol{A}$ **do**

for all $\boldsymbol{f}(\boldsymbol{x_k}, \boldsymbol{a}) \subset \boldsymbol{C_i}(\boldsymbol{a})$ **do**

if $\boldsymbol{A_i}(\boldsymbol{a}) = \{\boldsymbol{x_k} \in \boldsymbol{X}: \mathbf{min}(\boldsymbol{C_i}(\boldsymbol{a})) \leq \boldsymbol{f}(\boldsymbol{x_k}, \boldsymbol{a}) \leq \mathbf{max}(\boldsymbol{C_i}(\boldsymbol{a}))\}$ **then**

$$\boldsymbol{B_j}(\boldsymbol{a}) = \boldsymbol{A_j}(\boldsymbol{a}) - \bigcup_{\substack{\boldsymbol{i=1} \\ (\boldsymbol{i} \neq \boldsymbol{j})}}^{\boldsymbol{t}} \boldsymbol{A_i}(\boldsymbol{a})$$

end if

end for

$$\boldsymbol{w_a} = \frac{\cup_{\boldsymbol{i=1}}^{\boldsymbol{t}} |\boldsymbol{B_i}(\boldsymbol{a})|}{\boldsymbol{n}}$$

end for

Step 2: Normalize the weights

$$\boldsymbol{w_a^*} = \frac{\boldsymbol{w_a}}{\sum_{\boldsymbol{a}} \boldsymbol{w_a}}$$

Step 3: Compute the weighted similarity between $\mathbf{x}'$ and every sample in $\boldsymbol{X}$.

for all $\boldsymbol{a} \subset \boldsymbol{A}$ **do**

$$\boldsymbol{sim_a}(\boldsymbol{z}, \boldsymbol{x_i}) = \mathbf{1} - \frac{|\boldsymbol{f}(\boldsymbol{z}, \boldsymbol{a}) - \boldsymbol{f}(\boldsymbol{x_i}, \boldsymbol{a})|}{\boldsymbol{max}(\boldsymbol{a}) - \boldsymbol{min}(\boldsymbol{a})}$$

end for

$$\boldsymbol{SIM_A}(\boldsymbol{z}, \boldsymbol{x_i}) = \sum_{\boldsymbol{a} \in \boldsymbol{A}} \boldsymbol{w_a} \boldsymbol{sim_a}(\boldsymbol{z}, \boldsymbol{x_i})$$

Step 4: Select $\boldsymbol{X_z} \boldsymbol{\epsilon} \boldsymbol{X}$, the set of k nearest neighbors of $\mathbf{x}'$ by the top k highest similarities.

Step 5: Assign a majority weighted voting class label to the query $\mathbf{x}'$.

$$\boldsymbol{y'} = \mathbf{arg\,max}_{\boldsymbol{v}} \sum_{(\boldsymbol{x_i}, \boldsymbol{y_i}) \epsilon \boldsymbol{X_z}} \boldsymbol{I}(\boldsymbol{v} = \boldsymbol{y_i})$$

Output: $\boldsymbol{y'}$**,** class label of test sample.

The proposed method, WS-kNN, was compared with the classical kNN algorithm using the Euclidean distance. A 10-fold cross validation was applied for the both methods to obtain overall classification accuracy. The neighborhood size k ranged from 1 to 15. The size of datasets used in experimental analysis is small. Therefore, the k value greater than 15 was not used. Keeping the k value larger will reduce the effectiveness of both methods as it can increase the processing time. The best accuracy of each method with corresponding k value is presented in Table 2.

As shown in Table 2, WS-kNN has comparable performance with the classical kNN method using

Euclidean distance. The proposed method provided better performance for the wine and the sonar datasets. It is seen for the other datasets, which are iris, vehicle, transfusion and breast-cancer datasets, the performance of both methods is close to each other and the Euclidean-kNN method provides better performance on Ionosphere, *E. coli* and Haberman datasets. The accuracies (for k=1 to 15) were summed and the average accuracies were calculated for each dataset (Table 3). Table 3 shows similar results to Table 2.

Table 2. The best accuracy of each method with corresponding k in the parentheses (the accuracy rates in bold-face are the best performance among the methods).

#	Dataset	WS-kNN	Euclidean-kNN
1	Iris	0.97 (7)	0.97 (7)
2	Wine	0.98 (1)	**0.77 (2)**
3	Glass	**0.68 (1)**	0.72 (1)
4	Sonar	1 (1)	**0.84 (2)**
5	Vehicle	0.66 (7)	0.66 (4)
6	Ionosphere	**0.64 (2)**	0.87 (1)
7	Transfusion	0.77 (15)	0.77 (5)
8	*E. coli*	**0.78 (11)**	0.87 (7)
9	Haberman	**0.72 (10)**	0.75 (11)
10	Breast Cancer	0.96 (13)	0.96 (8)

Table 3. The average accuracy of each method (the accuracy rates in bold-face are the best performance among the methods).

#	Dataset	WS-kNN	Euclidean-kNN
1	Iris	0.96	0.96
2	Wine	**0.98**	0.70
3	Glass	0.64	0.64
4	Sonar	**1.00**	0.75
5	Vehicle	0.64	0.64
6	Ionosphere	0.58	**0.83**
7	Transfusion	**0.76**	0.75
8	*E. coli*	0.77	**0.85**
9	Haberman	0.69	**0.73**
10	Breast Cancer	0.96	0.96

In addition, in Table 3 the comparison of the processing times of methods are presented for some k values. It is seen that the proposed method is not as fast as Euclidean-kNN algorithm. However, in this method presented as a preliminary study, by using code optimization it is expected that the results can be further improved.

Table 3. The processing times in seconds for each method.

#	Dataset	k	WS-kNN	Euclidean-kNN
1	Iris	5	0.43	0.32
		10	0.43	0.31
		15	0.43	0.32
2	Wine	5	1.84	1.01
		10	1.88	1.95
		15	1.99	1.03
3	Glass	5	2.06	1.13
		10	1.92	2.10
		15	3.68	1.13
4	Sonar	5	16.75	10.10
		10	21.93	5.27
		15	16.76	6.39
5	Vehicle	5	99.95	54.00
		10	57.90	56.73
		15	89.78	56.50
6	Ionosphere	5	30.44	8.81
		10	33.57	16.92
		15	33.95	8.62
7	Transfusion	5	6.54	5.35
		10	6.31	5.37
		15	6.59	5.44
8	*E. coli*	5	3.12	2.32
		10	3.12	2.20
		15	3.12	4.29
9	Haberman	5	1.17	0.77
		10	0.84	0.81
		15	0.81	0.77
10	Breast-cancer	5	13.40	8.12
		10	13.35	8.21
		15	12.97	8.08

For both methods, the change of accuracy value according to k value is examined (for k= 1 to 15). In Figures 1-10, accuracies are shown depending on the k value. When the figures are analyzed, the accuracy

Celal Bayar University Journal of Science
Volume 15, Issue 4, 2019 p 393-400
Doi: 10.18466/cbayarfbe.618964

value obtained in both methods, as it seen, can change depending on the k value.

Iris dataset. In both methods, the change in k value changes the accuracy (Figure 1).

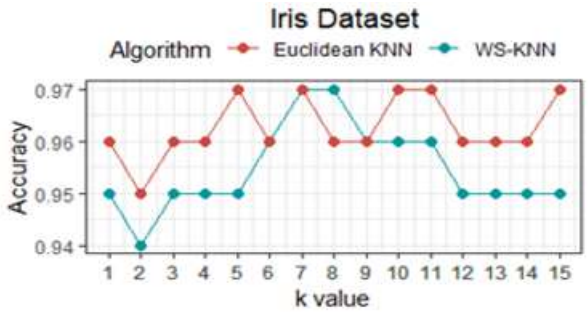

Figure 1. The accuracies via k value for Iris dataset.

Wine dataset. In both methods, the change in k value does not change the accuracy much (Figure 2).

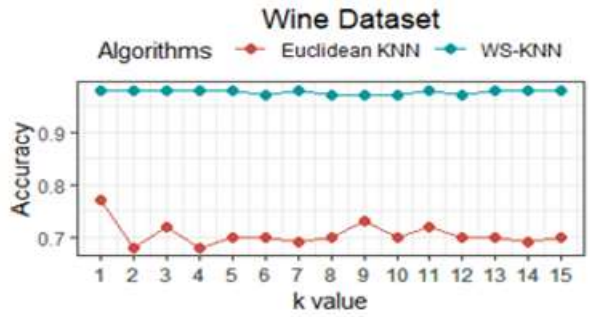

Figure 2. The accuracies via k value for Wine dataset.

Glass dataset. The accuracy generally decreases in both methods when the k value increases (Figure 3).

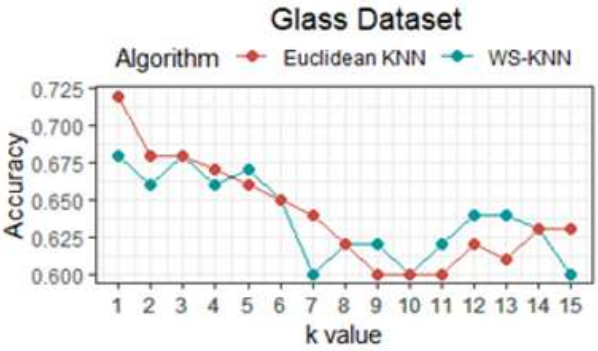

Figure 3. The accuracies via k value for Glass dataset.

Sonar dataset. The change of the k value in the WS-kNN method does not affect accuracy. However, in the Euclidean kNN when the k value increases, the accuracy generally decreases (Figure 4).

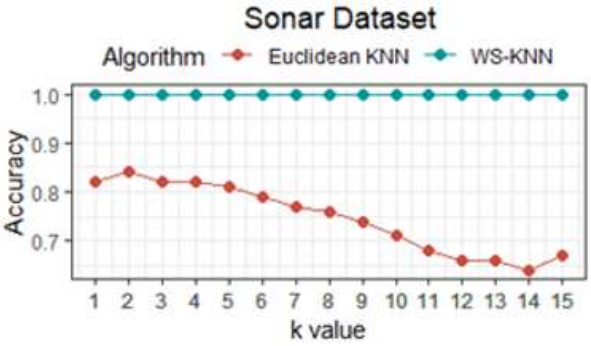

Figure 4. The accuracies via k value for Sonar dataset.

Vehicle dataset. In both methods, the change in k value changes the accuracy (Figure 5).

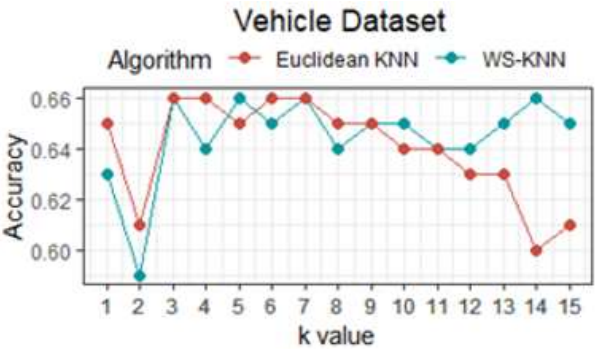

Figure 5. The accuracies via k value for Vehicle dataset.

Ionosphere dataset. In both methods, the change in k value changes the accuracy (Figure 6). However, there is more clear change in the accuracies of WS-KNN method.

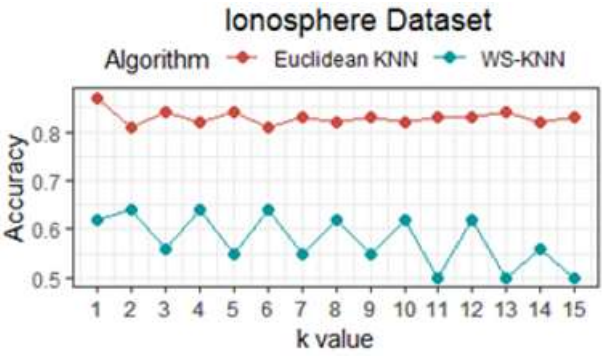

Figure 6. The accuracies via k value for Ionosphere dataset.

Transfusion dataset. In the WS-kNN, the change in k value does not change the accuracy value much. In the Euclidean, when k value is greater than 5, the accuracy does not change (Figure 7).

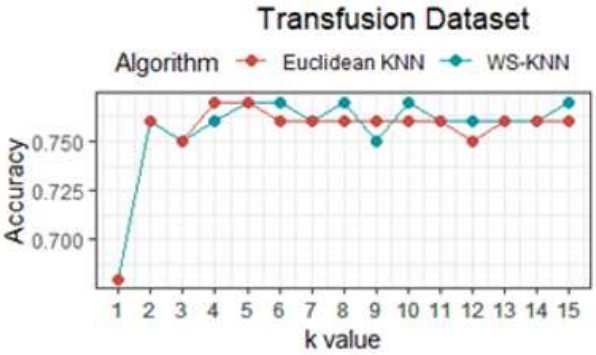

Figure 7. The accuracies via k value for Transfusion dataset.

E. coli dataset. In both methods, the change in k value changes the accuracy (Figure 8).

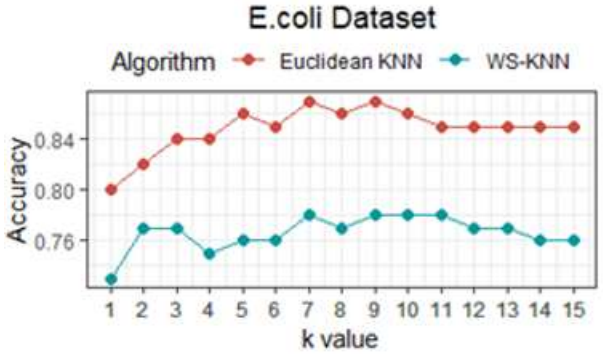

Figure 8. The accuracies via k value for *E. coli* dataset.

Haberman dataset. In both methods, when the k value increases, the accuracy generally increases (Figure 9).

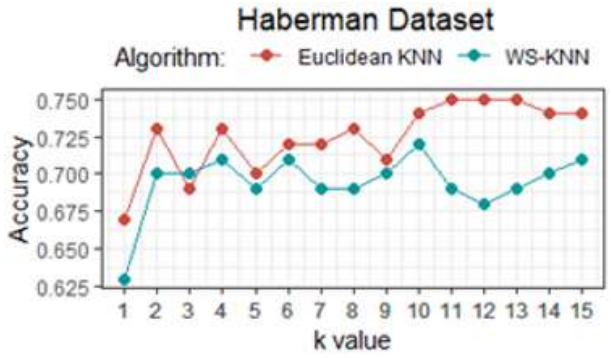

Figure 9. The accuracies via k value for Haberman dataset.

Breast-cancer dataset. In both methods, when k value is greater than 5, the accuracy does not change (Figure 10).

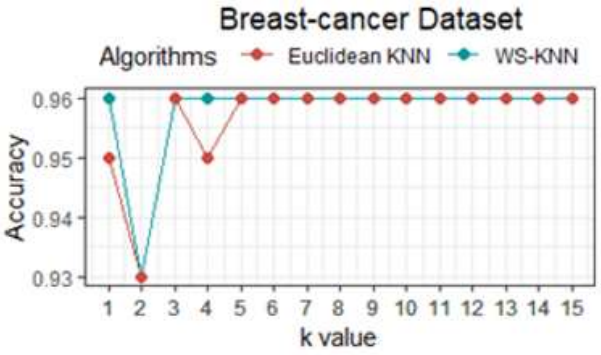

Figure 10. The accuracies via k value for Breast-cancer dataset.

4. Conclusion

In this study, the WS-kNN method has been proposed by using a weighted similarity as a proximity measure in the k-nearest neighbors method. In the proposed method, the similarities between the samples are weighted by the weights calculated for the attributes in the dataset. The method was compared with the classic k-nearest neighbors method using Euclidean distance. Classification accuracies were calculated by giving k = 1 to 15 values on 10 different data sets. According to the results, WS-kNN produces better results in half of the data sets compared to the Euclidean-kNN method. In addition, the proposed WS-kNN method showed a clear improvement for the sonar and wine data sets. These results show that the weighted similarity measure can be used as an alternative proximity measure in different classification methods.

Ethics

There are no ethical issues after the publication of this manuscript.

References

1. Jordan, MI, Mitchell, TM. 2015. Machine learning: Trends, perspectives, and prospects. *Science*; 349(6245): 255-260.

2. Singh, A, Thakur, N, Sharma, A. A review of supervised machine learning algorithms. In 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), IEEE, March 2016, pp. 1310-1315.

3. Cekik, R, Telceken, S. 2018. A new classification method based on rough sets theory. *Soft Computing*; 22(6): 1881-1889.

4. Soofi, AA, Awan, A. 2017. Classification Techniques in Machine Learning: Applications and Issues. *Journal of Basic and Applied Sciences*; 13: 459-465.

5. Aggarwal, CC. 2014. Instance-Based Learning: A Survey. *Data Classification: Algorithms and Applications*, 157.

6. Angiulli, F, Narvaez, E. 2018. Pruning strategies for nearest neighbors competence preservation learners. *Neurocomputing*; 308: 8-20.

7. Prasath, VB, Alfeilat, HAA, Lasassmeh, O, Hassanat, A. 2017. Distance and Similarity Measures Effect on the Performance of K-Nearest Neighbors Classifier-A Review. arXiv preprint arXiv:1708.04321.

8. Lei, Y, Zuo, MJ. 2009. Gear crack level identification based on weighted K nearest neighbors classification algorithm. *Mechanical Systems and Signal Processing*; 23(5): 1535-1547.

9. Khateeb, N, Usman, M. Efficient Heart Disease Prediction System using K-Nearest Neighbors Classification Technique, In Proceedings of the International Conference on Big Data and Internet of Thing ACM, December 2017, pp. 21-26.

10. Li, Q, Li, W, Zhang, J, Xu, Z. 2018. An improved k-nearest-neighbors method to diagnose breast cancer. *Analyst*; 143(12): 2807-2811.

11. Liu, Y, Wang, X, Yan, K. 2018. Hand gesture recognition based on concentric circular scan lines and weighted K-nearest neighbors algorithm. *Multimedia Tools and Applications*; 77(1): 209-223.

12. Kılınç, D. 2016. The Effect of Ensemble Learning Models on Turkish Text Classification. Celal Bayar Üniversitesi Fen Bilimleri Dergisi, 12(2).

13. Rodrigues, ÉO. 2018. Combining Minkowski and Cheyshev: New distance proposal and survey of distance metrics using k-nearest neighbours classifier. *Pattern Recognition Letters*; 110: 66-71.

14. Mulak, P, Talhar, N. 2015. Analysis of Distance Measures Using K-Nearest Neighbors Algorithm on KDD Dataset. *International Journal of Science and Research*; 4(7): 2101-2104.

15. Hu, LY, Huang, MW, Ke, SW, Tsai, CF. 2016. The distance function effect on k-nearest neighbors classification for medical datasets. *SpringerPlus*; 5(1): 1304.

16. Dialameh, M, Jahromi, MZ. 2017. A general feature-weighting function for classification problems. *Expert Systems with Applications*; 72: 177-188.

17. Jiao, L, Pan, Q, Feng, X, Yang, F. An evidential k-nearest neighbors classification method with weighted attributes, In Proceedings of the 16th International Conference on Information Fusion, IEEE, July 2013, pp. 145-150.

18. Marchiori, E. Class dependent feature weighting and k-nearest neighbors classification, In IAPR International Conference on Pattern Recognition in Bioinformatics, Springer, Berlin, Heidelberg, 2013, June, pp. 69-78.

19. Hassanat, AB. 2014. Dimensionality invariant similarity measure. *Journal of American Science*; 10(8).

20. Alkasassbeh, M, Altarawneh, GA, Hassanat, A. 2015. On enhancing the performance of nearest neighbour classifiers using hassanat distance metric. *Canadian Journal of Pure and Applied Sciences (CJPAS)*; 9(1).

21. Chomboon, K, Chujai, P, Teerarassamee, P, Kerdprasop, K, Kerdprasop, N. An empirical study of distance metrics for k-nearest neighbors algorithm, In Proceedings of the 3rd International Conference on Industrial Application Engineering, March 2015.

22. Kayaalp, N, Arslan, G. 2014. A Fuzzy Bayesian Classifier with Learned Mahalanobis Distance. *International Journal of Intelligent Systems*; 29(8): 713-726.

23. Kayaalp, N. Arslan, G. A New Fuzzy Bayesian Classification Approach, The 4th International Fuzzy Systems Symposium, İstanbul, 5-6 November 2015.

24. Greco, S, Matarazzo, B, Slowinski, R. 2001. Rough sets theory for multicriteria decision analysis. *European journal of operational research*; 129(1): 1-47.