SELCUK UNIVERSITY JOURNAL OF ENGINEERING SCIENCES

Journal homepage: http://sujes.selcuk.edu.tr/sujes

e-ISSN: 2757-8828

International Open Access

Volume 20
Issue 02

August, 2021

Research Article

Comparison of classification performance of kNN and WKNN algorithms

Fatih Tarakci [a,*], ***Ilker Ali Ozkan*** [b]

[a] *Selcuk University, Faculty of Technology, Department of Computer Engineering, Konya, 42130, Turkey*
[b] *Selcuk University, Faculty of Technology, Department of Computer Engineering, Konya, 42130, Turkey*

ARTICLE INFO

Article history:
Received 03 May 2021
Accepted 29 July 2021

Keywords:
kNN
WKNN
UCI
Machine Learning
Classification

ABSTRACT

In this study, K nearest neighbor (kNN) algorithm which is the most popular and widely used among the machine learning classification algorithms and the weighted kNN (WKNN) algorithm which takes the weight of the feature index into consideration, are used. As the weighting method, a weighting is made by taking the inverse of the distance squared ($w = 1 / d^2$). The confusion matrix of the data sets was created by applying the algorithms to five data sets via MATLAB program and the classification success was compared by conducting performance measurements of algorithms. It was observed that in two of the five data sets used in the study kNN algorithm turned out to make a more successful classification than WKNN while in three data sets the WKNN algorithm performed a more successful classification than the kNN.

1. Introduction

The K nearest neighbor (kNN) algorithm is one of the always popular and widely used classification methods due to its conceptual and computational simplicity. The algorithm makes a classification based on the principle of taking the majority vote of k neighbors in determining the class labels by calculating the distance of the unlabeled sample to all the labeled samples in the training set with various distance formulas [1]. Thus, all features contribute equally to the classification. However, this is not always a desired situation. Considering that the feature index has a great influence on the result of the classification and the weight of the classification feature index is not taken into account in the traditional kNN algorithm, this raises an important problem [2]. In addition, factors such as determining the number of k neighborhoods and choosing the distance function which is used to define neighbors are other problems affecting the classification success of kNN algorithm [3]. Therefore, after choosing the appropriate k neighbor number and distance function, using the weighted kNN (WKNN) which takes the weight of the feature index into account, will contribute to the improvement of the classification performance. Over the years, a large number of studies which offer a variety of weighting methods, have been conducted to increase the classification accuracy of the kNN [3]. It is useful to briefly mention some of these studies.

Tran and Ha used the WKNN algorithm to increase the accuracy of indoor visible light positioning systems with simple, real-time and stable methods in their article. They demonstrated that this algorithm provides very high positioning accuracy, is fully suitable for a few special two-dimensional interior positioning applications, and achieves more successful results than the kNN algorithm [4].

Karabulut et al. proposed the Weighted Similarity k-Nearest Neighbors algorithm (WS-kNN) in their article prepared in 2019. First, they calculated the weight of each feature and the similarity between the samples in the data set. Then they created a weighted similarity matrix by weighing the similarities according to the feature weights and using them as a proximity measure. The proposed algorithm was compared with the classical kNN method based on the Euclidean distance. In order to verify the performance of the algorithm, experiments were conducted on 10 different data sets downloaded from UCI. Experimental results showed that the proposed WS-k NN

[*] Corresponding author. E-mail address: *fatihtarakcii@gmail.com*

algorithm could achieve comparative classification accuracy [5].

Fan and et al. designed a new short-term load prediction model based on kNN by analyzing the historical power load data from The National Electricity Market (Australia) with the properties and regulations of electricity, in their article in 2019. They showed that the proposed estimation model could reflect the trend of variation by considering the inverse of the Euclidean distance to weigh the KNN algorithm and had a good adaptability in short-term load estimation [6].

Biswas et al. preferred to apply a weight to each of k neighbors with fuzzy logic method by using the Fuzzy Nearest Neighbor (FKNN) classifier, unlike the k-Nearest Neighbor Classifier (kNN) which treats neighbors equally in their article prepared in 2018. The proposed model was tested on 20 real-world datasets with different properties and compared with 8 cutting-edge and popular classifiers. It was observed that the method used provided a serious advantage over some classification algorithms [3].

Li and et al. applied the WKNN algorithm to the wine data set and reduced unrelated features in their study in 2015. And it was observed that the classification performance of the algorithm improved by determining the weight of each feature with sensitivity method [7].

A weighting approach was proposed for the kNN algorithm in the study prepared by Yiğit in 2013. The purpose of the proposed approach was to find the most suitable weights with the Artificial Bee Colony (ABC) algorithm. It was observed that the algorithm improved the correct classification performance and it was concluded that the ABC algorithm was applicable to the kNN algorithm [1].

The WKNN algorithm, which was created by weighting the kNN algorithm and thought to improve the classification performance, was applied to 5 data sets in this study. As the weighting method, a weighting is made by taking the inverse of the distance squared ($w = 1 / d^2$) and the results were compared with the classification performance of the kNN algorithm. While introducing the subject in the first part of the study, the material and method were included in the second part. After the findings and results were explained in detail in the third section, discussions and suggestions were presented in the fourth and last section.

2. Material and Method

2.1. Properties of the Data Set

The properties of the data sets used in this study are as follows:

The data set named “Rice Cammeo Osmancik” was downloaded from https://www.muratkoklu.com/datasets. The data set was created by taking the image of 3810 rice grains. The samples in the data set were classified into 2 classes (Osmancık and Cammeo) according to 7 morphological features [8].

The data set named "Raisin Grains" was downloaded from https://www.muratkoklu.com/datasets. The data set, including a total of 900 pieces were created from grapes grown in equal numbers from the two different kinds of raisins in Turkey. Raisins were classified into 2 classes (Keçimen and Besni) according to 7 characteristics [9].

The data set named “Heart Failure Clinical Records” was downloaded from UCI machine Learning Repository. The dataset included medical records of 299 (105 females, 194 males) heart failure patients collected at the Faisalabad Institute of Cardiology and Allied Hospital in Faisalabad (Punjab, Pakistan). There were 13 features in the data set. The data were classified into 2 classes according to the risk of death [10].

The data set named "Cervical Cancer Behavior Risk Data Set" was obtained from UCI Machine Learning Repository. The data set included 19 features and sample data from 72 people living in Jakarta, the capital city of Indonesia. While 22 people from the samples were at risk of cervical cancer, 50 people were healthy individuals. While creating the data set, 18 questionnaire questions were asked to the aforementioned individuals and classified into 2 classes as “at risk” or “not at risk” according to the risk of cervical cancer [11].

The data set named "Breast Cancer Coimbra Data Set" was obtained from UCI Machine Learning Repository. There were 166 samples and 10 features in the data set. Samples in the data set were classified into 2 classes as "breast cancer" or "healthy".

2.2. Data Pre-Processing

In cases where there is a large difference between the data, reducing the data into a single order gives more accurate results. For this purpose, the properties in the data sets used in the study were distributed between 0 and 1 by applying min-max normalization. Min-max normalization is given in Equation 1 [12].

$$x' = \frac{x-\min(x)}{\max(\mathrm{x})-\min(\mathrm{x})} \tag{1}$$

2.3. Performance Evaluation

Commonly used evaluation metrics such as accuracy, specificity, sensitivity, recall, and F1 score can be used to measure the performance of a classification algorithm. These measurements are calculated using the confusion matrix. The confusion matrix is a table often used to describe the performance of the classification model with a set of known test data. The confusion matrix consists of 4 parameters. These are: TP: True Positives, TN: True Negatives, FP: False Positives and FN: False Negatives. Table 1 shows the structure of the confusion matrix [13].

Table 1. Confusion matrix

		Predicted Class	
		0	1
Actual Class	0	TP (True Positives)	FP (False Positives)
	1	FN (False Negatives)	TN (True Negatives)

The performance measurement formulas calculated on the basis of the confusion matrix are given in Table 2 [14].

Table 2. Performance measurement formulas

Measure	Formula
Accuracy	(TP + TN) / (TP + FP + FN + TN)
Error Rate	(FP + FN) / (TP + FP + FN + TN)
Specificity	TN / (TN + FP)
P: Precision	TP / (TP + FP)
R: Recall	TP / (TP + FN)
F1 Score	(2 * P * R) + (P+ R)

2.4. Performance Evaluation

Cross validation is a standard assessment technique developed to increase the security of classification in machine learning. Cross validation is the division of the data set into k subgroups as training and testing. While one of the subgroups is used as a test set, the system is trained with the remaining sets. This process is repeated for the specified number of k and the system is tested [15]. Cross validation is shown in Figure 1 [9].

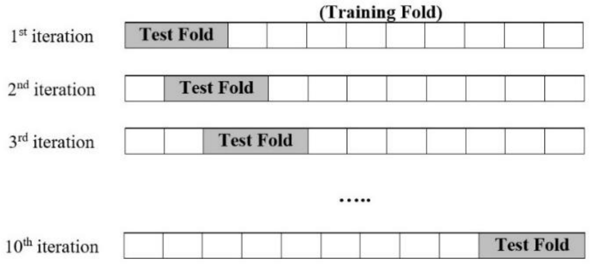

Figure 1. Cross validation

2.5. Classification Algorithms

The concept of classification can simply be expressed as distributing the data to classes defined in the data set. Classification algorithms, after learning the distribution of data from the data set, try to classify the class of the newly arrived test data, whose class is not known [16].

Since the classification algorithms K Nearest Neighbor Algorithm (kNN) and Weighted K Nearest Neighbor Algorithm (WKNN) are discussed in this study, it is useful to give brief information about these algorithms.

2.6. K Nearest Neighbor Algorithm (kNN)

K-Nearest Neighbor (kNN) algorithm proposed by T. M. Cover and P. E. Hart is a simple, effective and popular classification algorithm among machine learning algorithms. The main reasons of preference for the classification applications of the kNN algorithm are its lack of training, being easily monitored analytically, being easy to perform and resistant to noisy training data. The kNN algorithm calculates the distance of the new sample to be included in the data set from the existing data and determines the class of the sample data by looking at the class of the nearest k neighbors [17]. The structure of the kNN algorithm is shown in Figure 2 [18].

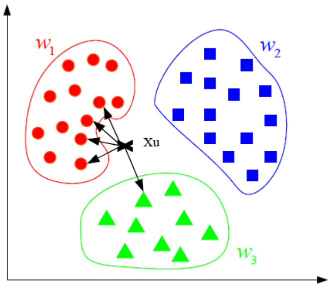

Figure 2. The structure of the kNN algorithm

Different calculation methods are used to calculate the distance between samples in the kNN algorithm. Euclidean distance, Manhattan distance, Chebyshev distance, Minkowski distance are among the calculation methods used. While different results occur with different distance measurements, different distance measurements are required for different data. The equations of these distances are as follows: [7].

Euclidean Distance: Measures the distance between two points on the plane. It is the most preferred method for the closest neighborhood algorithm.

$$Dist_{xy} = \sqrt{\sum_{k=1}^{m} (x_{ik} - y_{jk})^2} \tag{2}$$

Manhattan Distance: Sum of the absolute differences of two points in N dimensions.

$$Dist_{xy} = \sum_{i=1}^{n} |x_i - y_i| \tag{3}$$

Minkowski Distance: Equals Manhattan distance for case p = 1 and equals Euclidean distance for p = 2.

$$Dist_{xy} = (\sum_{i=1}^{n} |x_i - y_i|)^{1/p} \tag{4}$$

Chebyshev Distance: The greatest value of the absolute difference of two points.

$$Dist_{xy} = max_{x=1}^{n}|x_i - y_i| \tag{5}$$

2.7. Weighted K Nearest Neighbor Algorithm (WKNN)

Weighted kNN (WKNN) algorithm is a classification algorithm developed to reduce the error rate of the kNN algorithm. While classifying a new data in the kNN algorithm, after calculating the distance, the closest k neighbors' classes are examined. Various distance determination functions are used in the kNN algorithm. However, that this distance metric does not consider the suitability of the feature to solve the classification task, poses a problem. For this reason, the distances of all features contribute equally to choosing k nearest neighbors. The WKNN algorithm is used to eliminate this problem and make a better classification [19]. When the literature is scanned, it is seen that the features are weighted in various ways. Some of these are as follows: Determining appropriate weights with the Artificial Bee Colony algorithm, weighting neighbors with fuzzy logic method, weighting features using a kNN algorithm based on particle swarm optimization (PSO), weighting by taking the inverse of distance (w = 1 / d) and squaring the inverse of the distance. Weighting by The working structure of the WKNN algorithm is as follows [20]. K parameter is determined. The distances between the new sample and all other samples are calculated one by one. Calculated distances are sorted from small to large and the smallest k is selected among them. The weights of k selected samples are determined by calculating using Equation 6. taking (w = 1 / d^2). In this study, a weighting is made by the formula shown in Equation 6 by taking the inverse of the distance squared.

The working structure of the WKNN algorithm is as follows [20].

- K parameter is determined.
- The distances between the new sample and all other samples are calculated one by one.
- . Calculated distances are sorted from small to large and the smallest k is selected among them.
- The weights of k selected samples are determined by calculating using Equation 6.

$$w = 1 / d^2 \qquad (6)$$

- The weights of the same classes are added together and the class of the new sample is determined by looking at the total weights of the classes of the closest neighbors.

3. Findings and Conclusions

By applying min-max normalization to the data sets used in the study, all data were distributed between 0-1In all of the models used, the number of cross validation and the number of closest neighbors was determined as 10. Euclidean distance was used to calculate the distance between samples in kNN and WKNN algorithms. The complexity matrices of the algorithms for each data set were extracted using the MATLAB program, and the performance measurements were calculated. The confusion matrix of the data sets is shown in Table 3.

The accuracy, error rate, specificity, sensitivity, recall and F1 score of the algorithms that classify the data sets were calculated using the confusion matrix, and performance measurements are shown in Table 4.

Table 3. Confusion matrix of algorithms for data sets

Algoritmalar (kNN - WKNN)			
Rice Cammeo Osmancik Dataset			
		Predicted Class	
		Cammeo	Osmancık
Actual Class	Cammeo	**1461** **1462**	**169** **168**
	Osmancık	**114** **149**	**2066** **2031**
Raisin Grains Dataset			
		Predicted Class	
		Besni	Kecimen
Actual Class	Besni	**374** **371**	**76** **79**
	Kecimen	**54** **54**	**396** **396**
Heart Failure Clinical Records			
		Predicted Class	
		0 (alive)	1 (dead)
Actual Class	**0** (alive)	**195** **185**	**8** **18**
	1 (dead)	**74** **62**	**22** **34**
Cervical Cancer Behavior Risk			
		Predicted Class	
		0 (no cancer)	**1** (cancer)
Actual Class	**0** (no cancer)	**51** **51**	**0** **0**
	1 (cancer)	**11** **8**	**10** **13**
Breast Cancer Coimbra Data Set			
		Predicted Class	
		1 (Healthy)	**2** (Patients)
Actual Class	**1** (Healthy)	**38** **40**	**14** **12**
	2 (Patients)	**23** **17**	**41** **47**

Tablo 4. Performance measures of algorithms for data sets

Data sets	**Algorithm**	**Accuracy**	**Error rate**	**Specificity**	**Precision**	**Recall**	**F1 Score**
Rice Cammeo Osmancik	kNN	92.60	7.40	94.77	92.76	89.63	91.17
	WKNN	91.70	8.30	93.16	90.75	89.69	90.21
Raisin Grains	kNN	85.60	14.40	88.00	87.38	83.11	85.19
	WKNN	85.22	14.78	88.00	87.29	82.44	84.80
Heart Failure Clinical Records	kNN	72.60	27.40	22.91	72.49	96.05	82.62
	WKNN	73.20	26.80	35.41	74.89	91.13	82.22

Cervical Cancer Behavior Risk	kNN	84.70	15.30	47.61	82.25	100	90.26
	WKNN	88.90	11.10	61.90	86.44	100	92.72
Breast Cancer Coimbra	kNN	68.10	31.90	64.06	62.29	73.07	67.25
	WKNN	75.00	25.00	73.43	70.17	76.92	73.39

Performance measurements of kNN and WKNN algorithms, which are among the machine learning classification algorithms, were made on 5 data sets and the results are shown in Table 4. It has been observed that the kNN algorithm is more successful than the WKNN algorithm in “Rice Cammeo Osmancik” and “Raisin Grains” data sets. While the kNN algorithm has classified 92.60% correctly in the “Rice Cammeo Osmancik” data set, it has made a more successful classification than WKNN with an accuracy of 85.60% in the "Raisin Grains" data set.

In “Heart Failure Clinical Records”, “Cervical Cancer Behavior Risk” and “Breast Cancer Coimbra” data sets, the WKNN algorithm has achieved more successful classification than the kNN algorithm. The classification success of the WKNN algorithm is 73.20% in the "Heart Failure Clinical Records" dataset, 88.90% in the "Cervical Cancer Behavior Risk" dataset, and 75% in the "Breast Cancer Coimbra" dataset.

When the 5 data sets used in the study are considered, while the kNN algorithm is more successful than the WKNN algorithm in 2 data sets, the WKNN algorithm has made a more successful classification than the kNN algorithm in 3 data sets. Considering these results, it is understood that the WKNN algorithm makes a more successful classification, but the classification success is not the same for each data set. Therefore, in a classification to be made with the kNN algorithm, it is thought that calculating the classification success of the WKNN algorithm and making a classification with the more successful algorithm is more appropriate.

Classification accuracy rates of algorithms are shown graphically in Figure 3.

4. Discussion and Suggestions

A new classification can be made by changing the number of cross validation used in the study, the number of nearest neighbors and the Euclidean distance used to calculate the distance between samples. In addition, the success of kNN and WKNN algorithms can be compared by testing them on more data sets. In a classification with the kNN algorithm, the WKNN algorithm may be preferred if it is more successful, considering the contribution of the weighting to the success of the classification.

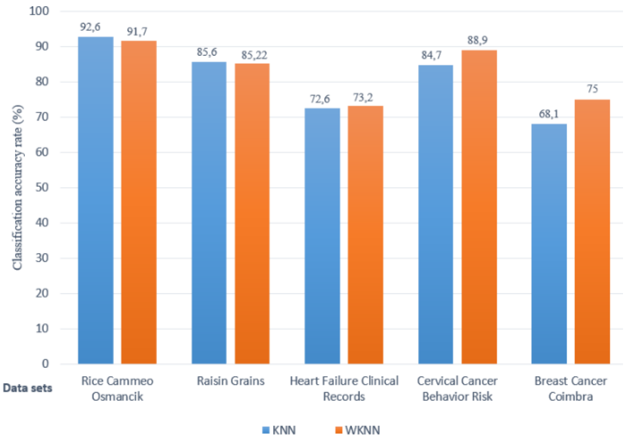

Figure 3. Classification accuracy rates of algorithms

References

[1] H. Yigit, "A weighting approach for KNN classifier," in 2013 International Conference on Electronics, Computer and Computation (ICECCO), 7-9 Nov. 2013 2013, pp. 228-231, doi: https://doi.org/10.1109/ICECCO.2013.6718270.

[2] H. Zhang, K. Hou, and Z. Zhou, "A Weighted KNN Algorithm Based on Entropy Method," in Intelligent Computing and Internet of Things, Pt Ii, vol. 924, (Communications in Computer and Information Science. Berlin: Springer-Verlag Berlin, 2018, pp. 443-451, doi: https://doi.org/10.1007/978-981-13-2384-3_41.

[3] N. Biswas, S. Chakraborty, S. S. Mullick, and S. Das, "A parameter independent fuzzy weighted k-Nearest neighbor classifier," Pattern Recognition Letters, vol. 101, pp. 80-87, 2018/01/01 2018, doi: https://doi.org/10.1016/j.patrec.2017.11.003.

[4] T. Huy Quang and H. Cheolkeun, "High Precision Weighted Optimum K-Nearest Neighbors Algorithm for Indoor Visible Light Positioning Applications," IEEE Access, vol. 8, pp. 114597-114607, 2020, doi: https://doi.org/10.1109/ACCESS.2020.3003977.

[5] B. Karabulut, G. Arslan, and H. M. Ünver, "A Weighted Similarity Measure for k-Nearest Neighbors Algorithm," Celal Bayar University Journal of Science, vol. 15, pp. 393 - 400, 2019, doi: https://doi.org/10.18466/cbayarfbe.618964.

[6] G.-F. Fan, Y.-H. Guo, J.-M. Zheng, and W.-C. Hong, "Application of the Weighted K-Nearest Neighbor Algorithm for Short-Term Load Forecasting," Energies, vol. 12, no. 5, p. 916, 2019, doi: https://doi.org/10.3390/en12050916.

[7] L. Zhang, C. Zhang, Q. Xu, and C. Liu, "Weigted-KNN and its application on UCI," in 2015 IEEE International Conference on Information and Automation, 8-10 Aug. 2015 2015, pp. 1748-1750, doi: https://doi.org/10.1109/ICInfA.2015.7279570.

[8] I. Cinar and M. Koklu, "Classification of Rice Varieties Using Artificial Intelligence Methods," International Journal of Intelligent Systems and Applications in Engineering, vol. 7(3), pp. 188-194, 2019, doi: https://doi.org/10.18201/ijisae.2019355381.

[9] I. Cinar, M. Koklu, and S. Tasemir, "Classification of Raisin Grains Using Machine Vision and Artificial Intelligence Methods," Gazi Mühendislik Bilimleri Dergisi, vol. 6(3), pp. 200-209, 2020, doi: https://doi.org/10.30855/gmbd.2020.03.03.

[10] T. Ahmad, A. Munir, S. H. Bhatti, M. Aftab, and M. A. Raza, "Survival analysis of heart failure patients: A case study," PLOS ONE, vol. 12, no. 7, p. e0181001, 2017, doi: https://doi.org/10.1371/journal.pone.0181001.

[11] R. M. Sobar and W. Adi, "Behavior determinant based cervical cancer early detection with machine learning algorithm," Advanced Science Letters, vol. 22, pp. 3120-3123, 10/01 2016, doi: https://doi.org/10.1166/asl.2016.7980.

[12] S. Jain, S. Shukla, and R. Wadhvani, "Dynamic selection of normalization techniques using data complexity measures," Expert Systems with Applications, vol. 106, pp. 252-262, 2018, doi: https://doi.org/10.1016/j.eswa.2018.04.008.

[13] O. Altay, M. Ulaş, and K. E. Alyamaç, "Prediction of the Fresh Performance of Steel Fiber Reinforced Self-Compacting Concrete Using Quadratic SVM and Weighted KNN Models," IEEE Access, Article vol. 8, pp. 92647-92658, 2020, doi: https://doi.org/10.1109/access.2020.2994562.

[14] M. Hossin and N. Sulaiman, "A Review on Evaluation Metrics for Data Classification Evaluations," International Journal of Data Mining & Knowledge Management Process, vol. 5, pp. 01-11, 03/31 2015, doi: https://doi.org/10.5121/ijdkp.2015.5201.

[15] C. Bergmeir and J. M. Benitez, "Forecaster performance evaluation with cross-validation and variants," in 2011 11th International Conference on Intelligent Systems Design and Applications, 22-24 Nov. 2011 2011, pp. 849-854, doi: https://doi.org/10.1109/ISDA.2011.6121763.

[16] A. Soofi and A. Awan, "Classification Techniques in Machine Learning: Applications and Issues," Journal of Basic & Applied Sciences, vol. 13, pp. 459-465, 08/29 2017, doi: https://doi.org/10.6000/1927-5129.2017.13.76.

[17] A. E. Taşçı and A. Onan, "K En Yakın Komşu Algoritması Parametrelerinin Sınıflandırma Performansı Üzerine Etkisinin İncelenmesi," presented at the Akademik Bilişim, Aydın,Türkiye, 2016.

[18] G.-f. Fan, Y.-H. Guo, J.-M. Zheng, and W.-C. Hong, "Application of the Weighted K-Nearest Neighbor Algorithm for Short-Term Load Forecasting," Energies, vol. 12, p. 916, 03/09 2019, doi: https://doi.org/10.3390/en12050916.

[19] Y. Gao and F. Gao, "Edited AdaBoost by weighted kNN," Neurocomputing, vol. 73, no. 16, pp. 3079-3088, 2010/10/01/ 2010, doi: https://doi.org/10.1016/j.neucom.2010.06.024.

[20] W. H. Jung and S. G. Lee, "An Arrhythmia Classification Method in Utilizing the Weighted KNN and the Fitness Rule," Irbm, Article vol. 38, no. 3, pp. 138-148, Jun 2017, doi: https://doi.org/10.1016/j.irbm.2017.04.002.