JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A Heuristic Optimal K-value Determination in K-Nearest Neighbor (KNN) Classification using Decision Tree Technique

Dr. D. Mabuni
Assistant Professor
Dept. of Computer Science and Technology
Dravidian University, Kuppam, Andhra Pradesh, India

Abstract: In data analytics K-nearest neighbor classification algorithm is given much importance for data classification. Finding optimal K-value and then for a given test tuple searching for all the K-nearest neighbors are two separate and critical tasks in K-nearest neighbor classification algorithm. Many methods exist for optimal K-value determination including cross validation technique. In this paper a new technique is proposed for optimal K-value determination using decision tree classifier model constructed from the given training dataset. Optimal K-value is determined directly by counting the number leaf nodes in the decision tree classifier. That is K-value is set equal to the number leaf nodes of the decision tree. The most important advantage of the decision tree classifier is that without determining K-value separately it is possible to find class label of the test tuple easily by searching and finding the correct leaf node in the decision tree and then setting the class label of the leaf node as the class label of the test tuple. In case if K-value is required then use proposed heuristic procedure, number of leaf nodes of the decision tree is set as K-value. Also, experimentally verified that the optimal K-values determined through K-nearest neighbor classification algorithm are almost equal to the K-value determined through the proposed decision tree classifier technique. Decision tree classifiers are powerful, well established, bench marking standard, highly accurate, scalable, and predominantly using techniques in both machine learning and data mining.

Index Terms: **Optimal K-value, K-value, decision tree classifier, multiple search trees, K-nearest neighbour values, KNN, K-nearest neighbor classification, machine learning, data mining.**

I. INTRODUCTION

Automatic data classification is the fundamental task in many applications such as pattern classification, image identification, medical data analysis, marketing, physics, biology, zoology, military, agriculture, retail, business, and banking and so on. Despite the success of K-nearest neighbor classification algorithm it must cleverly handle determination of K, searching of K-nearest neighbour values, and selecting correct classification rules. Finding a K-value is difficult and there are three ways to find K-value. In the first method same K-value is set for entire training dataset, in the second method different K-values are set for different subgroups of training datasets and in the third way different K-values are set for different test tuples.

In machine learning K-nearest neighbor classification is simple, efficient, effective and popular data classification algorithm. The K-nearest neighbor classification algorithm is included in the top-10 data mining algorithms list. Its main applications are – business, banking, text classification, research, pattern recognition, image processing, retail, target marketing, and task allocation and so on. In traditional K-nearest neighbor classification always fixed K-value will be set for all test tuples. Others assign different K-values for different test tuples using cross validation technique but it is a time-consuming process. K-nearest neighbor classification is a model free data classification technique. That is, KNN is non-parametric or lazy classification method. The fundamental requirement of the K-nearest neighbour classification algorithm is that the search complexity must be reduced to the maximum extent. Learning error increases as the K-value increases consequently test accuracy decreases. A low K-value causes overfitting and the high K-value causes high errors.

For a given test tuple, a set of nearest training tuples are searched and then majority class label of the nearest neighbors is assigned as the correct class label of the test tuple. There are two steps in K-nearest neighbor classification; in the first phase a suitable K-value is determined and in the second phase a set of K-nearest neighbors is selected. It is very difficult to find suitable K-value in K-nearest neighbor classification. In K-nearest neighbor classification one good idea is assign different K-values for different subspaces or assign different K-values for different test tuples. KNN search is really a challenging task in finding nearest neighbors because it has to search

entire training dataset. Sometimes it is good to find approximate nearest neighbors. KNN classification algorithm accuracy decreases as the unbalanced classes increases in the training dataset.

Setting the K-value and then searching for the K-nearest neighbours is the usual trend but the latest technique is combining the two steps into a single step. Cross validation, holdout cross validation and k-fold cross validation are important techniques for finding optimal K-value in KNN classification. There is no actual training step in KNN but in the testing step entire training data must be searched for finding nearest neighbors. In the traditional K-nearest neighbor classification first K-value is determined by using a suitable technique and then K-nearest neighbors are determined by using a suitable distance metric and then majority class label of the nearest neighbors is assigned to the test tuple. Sometimes weighted K-nearest neighbor classification rule is used in some applications. Weight of the nearest tuple increases as the distance between test tuple and the nearest neighbour decreases.

II. RELATED WORK

The K-nearest neighbor classification KNN algorithm is one of the most important data classification algorithms in data analytics, machine learning, artificial intelligence, pattern analysis, and data mining. It has been popularly used in many real-world data classification applications. Though it is being implemented successfully in many real time applications still it has got its own disadvantages including optimal K-value determination, over fitting, selecting suitable nearest neighbors, selecting and then applying nearest neighbor search, and utilization of classification rules. Zhang [1] proposed algorithms for handling many of the drawbacks of the K-nearest neighbor classification algorithm including imbalance training dataset sizes. Whenever there is a problem in creating a data model for a given training dataset it is advisable to apply K-nearest neighbor classification algorithm on that data for obtaining an appropriate solution. K-nearest neighbor classification algorithm is one of the top-10 data mining algorithms; as a result of this a special focus has been given to KNN in artificial intelligence. The K-nearest neighbor classification algorithm is very useful for finding useful patterns in big data analytics.

In general, K-nearest neighbor classification algorithm is simple and good in performance but there are two problems with it. The first problem is that it requires a distance measure to find distances from the test tuple to training tuples and in the second problem searching time complexity is very high due to vast searching in the entire training dataset. To overcome these two problems T. Liao et al. proposed a new K-nearest convolutional neural networks technique for learning a suitable distance metric and it avoids noisy training tuples in the training dataset[2].The K-Nearest Neighbor Classification is a lazy data classification technique in that searching starts only when the test tuple is given. S. Zhang [3] proposed a one-step computation procedure to replace the lazy portion of the actual K-Nearest Neighbor Classification algorithm. Lazy part task is transformed into a matrix computation. Determining K-value and searching nearest neighbors is converted into a unified function and a new classification rule is proposed for improving the one step KNN classification results.Simplicity and significant performance are the attractive features of K-nearest neighbour classification. S. Zhang and others [4] proposed a kTree technique to learn different K-values for different test tuples. Optimal K-values are learned first and then kTree decision tree is constructed. Also proposed kTree technique is similar in terms of running cost but classification accuracy is very high. Also proposed a variant of kTree called k*Tree for further speeding up the testing procedure.

S. Zhang et al. proposed a new technique called correlation matrix technique for learning different K-values for different test tuples [5]. Zhang was revised K-nearest neighbour classification algorithm using sparse reconstruction framework that can automatically generates different K-values for different test samples[6]. In K-nearest neighbour data classification, determining K-value is difficult and the K-value is not optimal in all the cases. S. Liu and S. Qin proposed a new technique using weighted values for finding K-value particularly for imbalanced datasets. According to tuple distribution local optimal K-value is determined for each test tuple. The experimental results have shown that proposed technique is better in performance accuracy.K-nearest neighbour is a non-parametric data classification technique. Gou et al. proposed local mean representation-based K-nearest neighbour data classification technique and experimentally proved that the performance of the proposed technique is far better than the existing methods[7]. For each class K-nearest neighbors are determined and a local mean vector is constructed for K-neighbors of each class. By taking the linear representation, a relationship matrix is obtained and this matrix is used to construct a new distance function.

Simple K-value and simple majority voting techniques in K-nearest neighbour classification are not up to the mark. Pan et al. proposed a new adaptive K-nearest neighbour classification algorithm using first majority and second majority classes in K-neighbor-hood of the query[8].K-nearest neighbour classification algorithm is predominantly used in many cloud-based services such as on line shopping, face recognition, recommender systems, retail, database searches, transaction management, distributed data management and so on. Chen et al. proposed protocols for security improvement using top-K selection form the given n instances[9]. Experimentally they have shown that faster response time is achieved.Effective and efficient management of traffic data is very important and it is particularly important in managing temporal data. L. Zheng et al. proposed a tensor-based data structure approach for spatial data management using K-nearest neighbour classification[10]. Tensor-based KNN is very useful in traffic prediction under many data missing contexts.

Data classification is one of the most studied technique in machine learning. Imbalanced data learning is difficult in machine learning. Minority class in the imbalanced dataset is very important and plays an important role in data classification. Mahin et al. proposed a

technique for separating minority class based on the distance of the logical neighbourhood using dataset specific distance function[11].K-nearest neighbour classification algorithm is the most important algorithm in pattern recognition and its performance is dependent on the selected distance metric and the selected intelligent data structure for searching. Jiao and Pan proposed a new KNN classifier called BPkNN based on pairwise distance metrics and belief function theory [12]. In the proposed technique learning of global distance metric is replaced with pairwise separate distance metrics. Pairwise KNN sub-classifiers are adaptively designed for separating the classes. Experimental results have shown that proposed method performance better than the existing methods.

Zhang [13] proposed two efficient sensitive K-nearest neighbour data classification techniques called direct and distance K-nearest neighbour methods. These two techniques are further enriched with additional features such as ensemble, smoothing, minimum cost K-value selection and so on. Experimental results have shown that proposed techniques significantly reduce misclassification costs.M. Ahmed et al. proposed an optimized K-nearest neighbour data classification algorithm using search optimization technique in wireless sensor networks[14]. The proposed method improves the network security and reduces the networks' energy consumption.Zhu et al. proposed adaptive process monitoring technique for controlling the imbalanced datasets, insufficient training data in the case of ill-posed datasets, nonlinearity and time varying behaviours[15]. A distance-based search technique is applied for reducing the computational load of online implementations.Authors [16] proposed compression algorithm for K-nearest neighbour classification with performance guarantee limits by using performance bounds.In K-nearest neighbour classification quality of the neighbourhood plays a significant role. Authors [17] proposed a deep learning architecture for learning optimal similarity function. This similarity function is not predetermined but it is dynamically determined on the fly and it handles high dimensional data very easily.

Yan et al.proposed a new K-nearest neighbour classification search by combining multiple search trees[18]. Computation complexity is reduced to the maximum extent with the proposed technique.Manocha et al. designed a probabilistic method for computing the optimal-K-values of nearest neighbors [19]. Sun et al. cleverly designed an adaptive and an intelligent new algorithm for finding the optimal K-values in the K-nearest neighbour classification for a test data [20].Li et al. proposed a technique that assigns a separate set of nearest neighbors for each class in the training dataset. Large K-value will be assigned for large class and small K-value will be assigned for small class. This type K-nearest neighbour classification is particularly suitable for text data classification [21].Mahin et al. set the computed geometric mean value as K-value for selecting and processing K-nearest neighbors and this new technique is applied on the unbalanced datasets [22].Yu et al. proposed a new distance metric called iDistance, which is particularly suitable for finding K-nearest neighbors in high dimensional data space and it uses a special index data structure for search speedup [23].

Salvador et al. proposed a compressed technique for finding K-nearest neighbours and make it suitable for big data analytics[24]. First it compresses into different groups and then it tries to find nearest neighbors through clever search.Le et al. proposed a deep similarity metric for finding K-nearest neighbors and it uses a special function to map given data to high dimensional space which increases the performance accuracy of the K-nearest neighbour classification [25].Gou et al. proposed K-nearest neighbour classification algorithm using local mean vector representation [26]. Singh et al. proposed K-nearest neighbour classification for image identification [27]. They used the KNN algorithm to find the nearest average distance instead of finding maximum number of nearest neighbors. Meha et al. proposed K-nearest neighbour classification algorithm through harmonic mean distance [28]. For each class it computes K-nearest centroid neighbors of the test data. It also computes local centroid mean vector. It uses the harmonic mean distance between the test data and local centroid mean to predict the class label of the test tuple. Syaliman et al. proposed a new K-nearest neighbour classification algorithm using local mean and distance weight parameters. It computes local mean vector for each class and wights based on distance [29].

III. PROCEDURE FOR DETERMINING OPTIMAL K-VALUE IN K-NEAREST NEIGHBOR CLASSIFICATION USING DECISION TREE CLASSIFIER

Decision tree model is very simple and convenient for finding class label of the test tuple without actually determining the K-value. If K-value is really needed then set K-value equal to the number of leaf nodes of the decision tree.Different techniques have been developed for finding optimal K-value in K-nearest neighbor classification including general methods, heuristic methods, scientific methods, and cross validation techniques. In this paper a new heuristic technique is proposedfor finding K-value in K-nearest neighbor(KNN) classification using decision tree model. Decision tree is a simple and efficient model for easy determination of actual and optimal K-value. Time complexity of the test tuple using decision tree classifier technique is only O(log n), which is very negligible one for any real datasets.

In this paper proposed heuristic technique is explained here. Before using K-nearest neighborclassification first construct decision tree classifier for the given training dataset and then directly count the number of leaf nodes in the decision tree and then set this leaf node count as the optimal K-value in the K-nearest neighbor (KNN) classification. Now execute K-nearest neighbor classification algorithm for finding the class label of the given test tuple. Decision tree will be created only once for entire training dataset and it can be used for finding accuracy of all the test tuples.

Decision tree model is a well-established and a well-known classification and regression model in machine learning. To compute K-valueinK-nearest neighbor classification, initially a pruned decision tree classifier is constructed for the given dataset and from the resulted decision tree,K-value is directly set as the number of leaf nodes of the newly constructed decision tree classifier. This heuristic technique is very easy and very simple to apply. Finding K-valueusing decision tree classifier construction with time complexity O(logn) is computationally efficient and usage of decision tree classifier is set as a bench marking technique before applying any data analytics task on the selected dataset. Once K-value is determined experimentally then normal K-nearest neighbor classification algorithm can be used for data classification in various applications. Finding K-value is considered to be the preprocessing step of theK-nearest neighbor classification algorithm. Different experimental and heuristic procedures fordeterminingK-value in K-nearest neighbor classification are shown in Table-1.

S.No	Techniques used for K-valuedetermination
1	K = cross validation techniques
2	K = Sqrt(n)
3	K = Sqrt(Sqrt(n))
4	K = Sqrt(n)/2
5	K = Number of distinct class labels in the training dataset
6	**K = Number of leaf nodes in the decision tree**
7	K = Number of non-leaf nodes in the decision tree
8	K = Number of leaf nodes satisfying threshold probability

Table-1 Different ways useful for finding K-value in K-nearest neighbor classification

Experiments are conducted by taking many UCI machine learning training datasets and then experimental results containing various parameters are shown in the respective tables. Pruned decision trees are used for reducing the error value as much as possible.

Iris dataset	Training data size = 150	Test data size 150	
S.No	Number of K-nearest neighbors	Test accuracy	Correctly Classified tuples
1	5	96.66	145
2	**10**	**98.0**	**147**
3	15	98.0	147
4	20	98.0	147
5	25	96.66	145
6	30	96.66	145
7	35	96.66	145
8	40	96.66	145
9	45	96.0	144
10	50	94.66	142

Table-2 Iris dataset test accuracies with K-nearest neighbor classification

As shown in the Table-2 Iris dataset test accuracies are determined by using K-nearest neighbor classification algorithm for different K-values. Optimal K-values are identified at highest test accuracies. From the Table-2 possible optimal K-values are 10 or 20 or 30. Now decision tree classifier is executed on Iris dataset for different pre-settled pre-pruned values and the test classification accuracies are noted down in the Table-3.From thedecision tree classifier accuracies results on Iris dataset, number of leaf nodes is selected and it is set as optimal K-value and in this case optimal K-value is set as K = 10. Bold K-values indicate possible optimal K-values. Sometimes average of these optimal K-values is taken as the final optimal K-value. When two or more accuracies are same then take the K-value as the average of the respective counts of leaf nodes or the highest leaf node count value in the tabular results as the real and correct K-value.

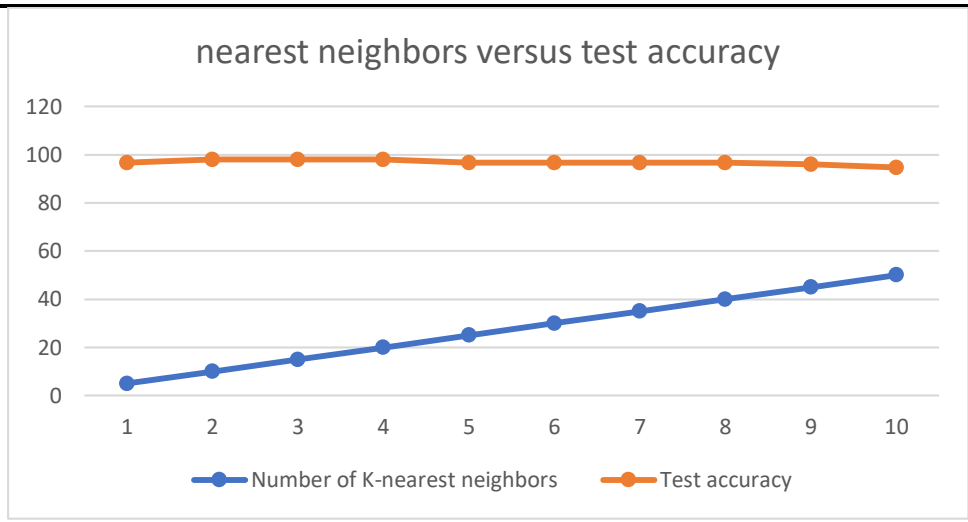

Figure-1 Relationship between nearest neighbors and test accuracies.

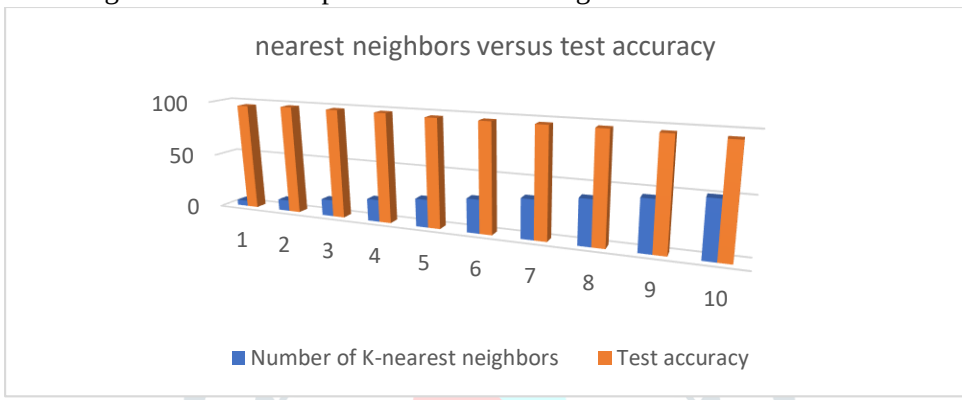

Figure-2 For Iris dataset nearest neighbors versus test accuracies.

Iris	Decision tree method	Training size = 150	Test size 150	
S.No	Pruned threshold	Number of leaf nodes	Test accuracy	Correctly Classified tuples
1	2	11	97.33	146
2	**3**	**10**	**98.0**	**147**
3	6	8	98.0	147
4	10	6	96.0	144
5	20	5	96.0	144
6	30	5	96.0	144
7	40	5	96.0	144

Table-3 Optimal K-value (leaf nodes) through Decision tree classifier using Iris dataset

Optimal K-value through decision tree classifier is selected from the Table-3 and it is set as optimal K = 10 against test accuracy value 98.0. from the both tables test accuracies are equal and at the same time K-value is equal to the number of leaf nodes of the newly constructed decision tree classifier. The heuristic method proposed in this paper is that count of leaf nodes of the decision tree is set equal to the optimal K-value. This point is experimentally verified and proved on Iris dataset.

Glass	Training size = 214	Test size 214	
S.No	Number of K-nearest neighbors	Test accuracy	Correctly Classified tuples
1	**5**	**99.53**	**213**
2	10	99.06	212
3	15	99.06	212
4	20	98.13	210
5	25	96.26	206
6	30	94.39	202
7	35	92.99	199
8	40	90.65	194
9	45	88.31	189
10	50	85.98	184

Table-4UCI machine learning Glass dataset test accuracies

K-nearest neighbor classification test accuracies for the Glass training dataset are shown in the Table-4, and possible optimal K-valuesof the K-nearest neighbor classifier areeither 5 or 10 or 15 depending upon the application. Based on the KNN highest test accuracies (99.53) obtained through experiments, one of the best possible optimal K-value can be taken and set as optimal K = 5. Sometimes average of the K-values against all the highest test accuracies is taken as the final optimal K-value.

Experimental results of decision tree classifier on glass training dataset are shown in Table-5 and the number of leaf nodes against highest test accuracy (100.0) is 6. So, possible optimal K-value is selected and then it is set as K = 6. Optimal K-value determined using decision tree classifier is K = 6 is almost equal to the optimal K-value determined through K-nearest neighbor classification. From this it is clear that number of leaf nodes in the decision tree with highest test accuracy can be set directly as optimal K-value without using K-nearest neighbor classification algorithm. Experimental results have shown that class label of the test tuple is determined directly with the class label of the leaf node in the decision tree.

Glass	Decision tree method	Training size = 214	Test size 214	
S.No	Pruned threshold	Number of leaf nodes	Test accuracy	Correctly Classified tuples
1	**2**	**6**	**100.0**	**214**
2	5	6	100.0	214
3	10	6	100.0	214
4	20	6	100.0	214
5	30	6	100.0	214
6	40	4	89.71	192
7	50	4	89.71	192

Table-5 Optimal K-value (leaf nodes) through Decision tree classifier using Glass dataset

Optimal K-value selected form the Table-5 is K = 6 and it is approximately equal to the K-nearest neighbor classifier results of the glass dataset. So, K = 5 or 6 is the correct K-value.

Breast cancer	Training size = 699	Test size = 699	
S.No	Number of K-nearest neighbors	Test accuracy	Correctly Classified tuples
1	**5**	**74.82**	**523**
2	10	68.24	477
3	15	69.95	489
4	20	69.81	488
5	25	68.24	477
6	30	68.38	478
7	35	68.09	476
8	40	67.66	473
9	45	66.95	468
10	50	67.09	469

Table-6UCI machine learning Breast cancer dataset test accuracies

Breast Cancer	Decision tree method	Training size = 699	Test size=699	
S.No	Pruned threshold	Number of leaf nodes	Test accuracy	Correctly Classified tuples
1	2	19	96.85	677
2	5	14	96.70	676
3	10	13	96.56	675
4	15	11	96.56	675
5	20	10	95.56	668
6	30	7	94.56	661
7	**40**	**7**	**94.56**	**661**

Table-7 Optimal K-value (leaf nodes) through Decision tree classifier using breast cancer dataset

Iono sphere	Training size = 351	Test size 351	
S.No	Number of K-nearest neighbors	Test accuracy	Correctly Classified tuples
1	**3**	**91.16**	**320**
2	4	85.47	300
3	5	87.46	307
4	10	83.47	293

5	15	85.18	299
6	20	84.33	296
7	25	84.04	295
8	30	82.33	289
9	35	82.05	288
10	40	81.19	285
11	45	80.91	284
12	50	78.34	275

Table-8UCI machine learningIonosphere dataset test accuracies

Iono sphere	Decision tree method	Training size = 351	Test size 351	
S.No	Pruned threshold	Number of leaf nodes	Test accuracy	Correctly Classified tuples
1	**2**	**4**	**92.30**	**324**
2	5	4	92.30	324
3	10	4	92.30	324
4	20	4	92.30	324
5	30	4	92.30	324
6	40	4	92.30	324

Table-9 Optimal K-value (leaf nodes) through Decision tree classifier using ionosphere dataset

From the Table-8 and Table-9 the optimal K-value is selected either 3 or 4 and then it is set as optimal K-value in the experimentation. In this case, finally, optimal K-value obtained from the decision tree classifier accuracy results is selected and then it is set as optimal K-value = 4. Always both the K-value results need not be equal and slightly they have some small differences. That is small differences between two different K-value computation procedures does not matters.

Mamographic masses	Training size = 961	Test size=961	
S.No	Number of K-nearest neighbors	Test accuracy	Correctly Classified tuples
1	5	83.66	804
2	**10**	**80.85**	**777**
3	15	79.39	763
4	20	79.08	760
5	25	78.56	755
6	30	78.14	751
7	35	78.45	754
8	40	78.56	755
9	45	78.25	752
10	50	77.93	749

Table-10 UCI machine learningMamographicmasses dataset test accuracies with K-nearest neighbor classification

Mamographic Masses	Decision tree method	Training size = 961	Test size 961	
S.No	Pruned threshold	Number of leaf nodes	Test accuracy	Correctly Classified tuples
1	2	13	81.37	782
2	5	13	81.37	782
3	10	13	81.37	782
4	20	13	81.37	782
5	**30**	**11**	**81.37**	**782**
6	40	11	81.37	782
7	50	11	81.37	782
8	60	10	81.37	782
9	70	10	81.37	782
10	80	10	81.37	782
11	200	7	78.77	757
12	500	3	80.22	771

Table11 Optimal K-value (leaf nodes) through Decision tree classifier using Mamographic masses

First, K-nearest neighbor classification algorithm is executed on UCI machine learning mammographic dataset for different K-values and the experimented results are noted down in the Table-10. Secondly, decision tree classifier model is constructed for the UCI machine learning mammographic dataset and the experimental results containing various parameters are noted down in the Table-11. Carefully

observe the Table-11, here test data accuracy is not changing though there is a change in the pre pruning parameters with selected threshold. Hence, any count value of leaf nodes in the range can be set as K-value. In this case K-value may be either 10 or 11 or 12 or average of these values.

Clearly some relationships are there among the parameters. These relationships are reliable and useful in real time applications scenarios. From the tables Table-10 and Table-11 it is clear that approximate optimal K-value is set as K = 10. Both K-nearest neighbor classifier and decision tree classifier methods are giving the same accuracy and same K-value. It shows that decision tree classifier method is a potentially suitable method for the proposed method and it is very easy and convenient for finding optimal K-value for any training dataset because the decision tree is dominant, standard, reliable, bench marked, scalable, rule based, and ever green data structure and effective data classification model.

CONCLUSIONS

K-nearest neighbor classification algorithm is very simple, non-parametric, and popular classification algorithm. In K-nearest neighbor classification how to set different K-values for different test datasets is still an open problem and, in many cases, approximate K-value is used. For improving the KNN search, an appropriate, efficient and correct tree structure index like data structure must be identified. In this paper particularly a new direct technique, called K-nearest neighbor decision tree (KNNDT) classifier is proposed for optimal K-value determination using normal decision tree classifier. The attractive feature of the decision tree is that the class label of the test tuple can be determined directly form the suitable class label of the leaf node in O(log n) time without actually going for K-nearest neighbour classification algorithm. In the future special care will be taken and investigated for enhancing efficiency of the K-nearest neighbor search in K-nearest neighbor classification.

For simplicity and easy calculation purpose same training dataset is taken as the test dataset but in the future, this would be taken distinctly. That is, training dataset and test dataset would be created distinctly without any overlapping.

REFERENCES

[1] S. Zhang, “Challenges in KNN Classification”, IEEE Transactions on Knowledge and Data Engineering, Oct.2022, pp.4663-4675, vol.34

[2] T. Liao, Z. Lei, T. Zhu, S. Zheng, Y. Li, and C. Yuan, “Deep Metric Learning for K-Nearest Neighbor Classification”, Published in: IEEE Transactions on Knowledge and Data Engineering (Volume: 35, Issue: 1, 01 January 2023)

[3] S. Zhang, “KNN Classification with One-Step Computation”, Published in: IEEE Transactions on Knowledge and Data Engineering, Manuscript received 21 Dec. 2020; revised 23 Aug. 2021; accepted 6 Oct. 2021.

[4] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, “Efficient kNN classification with different numbers of nearest neighbors,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 5, pp. 1774–1785, May 2018.

[5] S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng, “Learning k for kNN classification,” ACM Trans. Intell. Syst. Technol., vol. 8, no. 3, pp. 1–19, 2017.

[6] S. Liu, P. Zhu, and S. Qin, “An improved weighted KNN algorithm for imbalanced data classification,” in Proc. IEEE 4th Int. Conf. Comput. Commun., 2018, pp. 1814–1819.

[7] J. Gou, W. Qiu, Z. Yi, Y. Xu, Q. Mao, and Y. Zhan, “A local mean representation-based k-nearest neighbor classifier,” ACM Trans. Intell. Syst. Technol., vol. 10, no. 3, pp. 1–25, 2019.

[8] Z. Pan, Y. Wang, and Y. Pan, “A new locally adaptive k-nearest neighbor algorithm based on discrimination class,” Knowl.-Based Syst., vol. 204, 2020, Art. no. 106185.

[9] H. Chen, I. Chillotti, Y. Dong, O. Poburinnaya, I. Razenshteyn, and M. S. Riazi, “SANNS: Scaling up secure approximate k-nearest neighbors search,” in Proc. 29th USENIX Secur. Symp., 2020, pp. 2111–2128.

[10] L. Zheng, H. Huang, C. Zhu, and K. Zhang, “A tensor-based k-nearest neighbors method for traffic speed prediction under data missing,” Transport metrica B, Transp. Dyn., vol. 8, no. 1, pp. 182–199, 2020.

[11] M. Mahin, M. J. Islam, B. C. Debnath, and A. Khatun, “Tuning distance metrics and k to find sub-categories of minority class from imbalance data using k nearest neighbours,” in Proc. Int. Conf. Electr. Comput. Commun. Eng., 2019, pp. 1–6.

[12] L. Jiao, X. Geng, and Q. Pan, “BpkNN: k-nearest neighbor classifier with pairwise distance metrics and belief function theory,” IEEE Access, vol. 7, pp. 48 935–48 947, 2019.

[13] S. Zhang, “Cost-sensitive KNN classification,” Neurocomputing, vol. 391, pp. 234–242, 2020

[14] M. M. Ahmed, A. Taha, A. E. Hassanien, and E. Hassanien, “An optimized k-nearest neighbor algorithm for extending wireless sensor network lifetime,” in Proc. Int. Conf. Adv. Mach. Learn. Technol. Appl., 2018, pp. 506–515

[15] W. Zhu, W. Sun, and J. Romagnoli, “Adaptive k-nearest-neighbor method for process monitoring,” Ind. Eng. Chem. Res., vol. 57, no. 7, pp. 2574–2586, 2018.

[16] L.-A. Gottlieb, A. Kontorovich, and P. Nisnevitch, “Near-optimal sample compression for nearest neighbors,” IEEE Trans. Inf. Theory, vol. 64, no. 6, pp. 4120–4128, Jun. 2018.

[17] L. Le, Y. Xie, and V. V. Raghavan, “Deep similarity-enhanced k nearest neighbors,” in Proc. IEEE Int. Conf. Big Data, 2018, pp. 2643–2650.

[18] D. Yan, Y. Wang, J. Wang, H. Wang, and Z. Li, "K-nearest neighbors search by random projection forests," IEEE Trans. Big Data, vol. 7, no. 1, pp. 147–157, Mar. 2021.

[19] S. Manocha and M. A. Girolami, "An empirical analysis of the probabilistic k-nearest neighbour classifier," Pattern Recognit. Lett., vol. 28, no. 13, pp. 1818–1824, 2007.

[20] S. Sun and R. Huang, "An adaptive k-nearest neighbor algorithm," Proc. 7th Int. Conf. Fuzzy Syst. Knowl. Discov., vol. 1, pp. 91–94, 2010.

[21] B. L. Li, S. W. Yu, and Q. Lu, "An improved k-nearest neighbor algorithm for text categorization," in Proc. Int. Conf. Comput. Process. Oriental Lang., Jan. 1, 2003, pp. 469–475.

[22] M. Mahin, M. J. Islam, B. C. Debnath, and A. Khatun, "Tuning distance metrics and k to find sub-categories of minority class from imbalance data using k nearest neighbours," in Proc. Int. Conf. Electr. Comput. Commun. Eng., 2019, pp. 1–6.

[23] C. Wang and Y. Yang, "Nearest neighbor with double neighborhoods algorithm for imbalanced classification," Int. J. Appl. Math., vol. 50, no. 1, pp. 1–13, 2020.

[24] J. Salvador-Meneses, Z. Ruiz-Chavez, and J. Garcia-Rodriguez, "Compressed KNN: K-nearest neighbors with data compression," Entropy, vol. 21, no. 3, 2019, Art. no. 234.

[25] L. Le, Y. Xie, and V. V. Raghavan, "Deep similarity-enhanced k nearest neighbors," in Proc. IEEE Int. Conf. Big Data, 2018, pp. 2643–2650.

[26] J. Gou, W. Qiu, Z. Yi, Y. Xu, Q. Mao, and Y. Zhan, "A local mean representation-based k-nearest neighbor classifier," ACM Trans. Intell. Syst. Technol., vol. 10, no. 3, pp. 1–25, 2019.

[27] S. Singh, J. Haddon, and M. Markou, "Nearest neighbour strategies for image understanding," in Proc. Workshop Adv. Concepts Intell. Vis. Syst., 1999, pp. 2–7

[28] S. Mehta, X. Shen, J. Gou, and D. Niu, "A new nearest centroid neighbor classifier based on k local means using harmonic mean distance," Inf., vol. 9, no. 9, 2018, Art. no. 234

[29] K. Syaliman, E. Nababan, and O. Sitompul, "Improving the accuracy of k-nearest neighbor using local mean based and distance weight," J. Phys., Conf. Series, vol. 978, no. 1, 2018, Art. no. 012047