Lazy Learning Associative Classification with WkNN and DWkNN Algorithm

Preeti Tamrakar[1], and S. P. Syed Ibrahim[1]*

1School of Computing Science and Engineering, VIT Chennai Campus, Tamil Nadu, 600127 India

Abstract. One of the algorithms, which prudently denote better outcomes than the traditional associative classification systems, is the Lazy learning associative classification (LLAC), where the processing of training data is delayed until a test instance is received, whereas in eager learning, before receiving queries, the system begins to process training data. Traditional method assumes that all items within a transaction is same, which is not always true. This paper recommends a new framework called lazy learning associative classification with WkNN (LLAC_WkNN) which uses weighted kNN method with LLAC, that gives a subset of rules when LLAC is applied to the dataset. In order to predict the class label of the unseen test case, the weighted kNN (WkNN) algorithm is then applied to this generated subset. This creates the enhanced accuracy of the classifier. The WkNN also gives an outlier more weight. By applying Dual Distance Weight to LLAC named as LLAC_DWkNN, this limitation of WkNN is resolved. LLAC_DWkNN gives less weightage to outliers, which improve the accuracy of the classifier, further. This algorithm has been applied to different datasets and the experiment results demonstrate that the proposed method is efficient as compared to the traditional and other existing systems**.**

1 Introduction

Associative classification (AC) [1] has been popularly explored in the last decades and taken full advantage of it in many successful real time applications. It integrates classification and frequent pattern mining into one system. AC can be of two types; Eager associative classification (EAC) and Lazy learning associative classification (LLAC). Two steps are involved in the construction of EAC. Training dataset is used to construct the model in the first step and test dataset is in the second step. The model is validated by using test dataset. EAC generates a large set of rules. During classification, many rules are not useful for classification and some important rules may not be generated. To overcome this issue, LLAC (non- eager) is proposed by Adriano Veloso et al. [2] where focus is given to the attributes of the given test cases. By this way, the chance of generating useful rules is increased. The processing of data is delayed by LLAC until the test instance requires classification. There are two phases in LLAC: 1. Subset Generation and 2. Subset evaluation. Test dataset is used to generate the subsets. In the second phase, subsets generated are assessed. Probability, support and confidence measures are used for the evaluation. For the provided test instance, the highest probability class label is allocated.

Eager associative classification generates a large set of rules. If one instance matches several rules, which classes are different, that is called rule conflict problem. On the other hand, kNN classifies test instance based upon the nearest training samples in the dataset. But it also suffers from two deficiencies: a. Its efficiency is less because it has to compute the

similarity/ distance between test and every single training instance and b. Accuracy is totally dependent how to determine the optimum value for k.

To overcome above said issues, Associative classification with kNN (AC-kNN) is proposed by Zaixiang Huang et al. [3]. By integrating these two concepts, it can resolve the rule conflict and low efficiency problem. Rules are developed in AC-kNN method based on the AC system such as CARS discovery, rule ranking, and pruning. kNN applies for the purpose of classification. Haida et al. [4] suggested HAC that would combine the AC algorithm with the NB algorithm to boost classification accuracy and to minimize the memory space and execution time.

Eager Associative Classification with kNN (EAC-kNN) resolves the issue of rule conflict and low efficiency, but Eager associative classification generates many rules. Some of them are not useful and some important rules are not generated. To address this issue, LLAC with Fixed kNN (LLAC_kNN) [5] is proposed in our previous work. Lazy learning associative classification (LLAC) delays the data processing until classification is required for a query. It generates rules specific to each test instance. In LLAC_kNN method, LLAC is applied to the dataset to generate the subset in the first step. In the second step, kNN is applied for the classification. By this way, better accuracy is achieved, but still suffers from the sensitivity problem of choosing k for KNN. To address the sensitivity problem of kNN, Dudani [6] proposed Weighted k Nearest Neighbor (WkNN), in that nearest neighbor gets the heavy weight and farthest neighbor gets the light weight according to their distance from the test instance. This WkNN method is used in the proposed work.

*corresponding author: syedibrahim.sp@vit.ac.in

For the further improvement in performance of LLAC_FkNN, This paper proposes LLAC with Weighed kNN (LLAC_WkNN). In this, subset is generated based on LLAC method. Then the distance is calculated between the generated subset and the test instance. Based on the distance, weight is calculated and assigned to all the k neighbors. In WkNN method, outliers are also getting high weights. To solve this issue Jianping Gou et al. [7] proposed Dual Weighted kNN (DWkNN) method. In that dual weight is assigned in place of the single weight. By applying dual weight, in between outliers get the lesser weight that leads to an accurate classifier. Thus, this paper proposes LLAC_DWkNN method to reduce the weight of outliers. The proposed work has improved the accuracy of the existing system and traditional systems.

At first, Liu B et al. [1] integrated classification and association rule mining (ARM) and referred to as associative classification (AC). AC constructs Class Association Rules (CARS), in that left-hand side is rule and right-hand side is restricted to a class label. AC has been applied fruitfully in several tasks of classification such as detecting fraud, spam filtering, diagnosing cancer, and so on. There are two forms of associative classification, EAC and LLAC.

There are two stages to the Eager Associative Classification (EAC). EAC uses association rule mining in the first step to build the rules on class association. In phase two, the classifier is constructed, based on the CARs created from phase one. The EAC offers enhanced accuracy, but some drawbacks exist.

2 Introduction

Associative classification (AC) [1] has been popularly explored in the last decades and taken full advantage of it in many successful real time applications. It integrates classification and frequent pattern mining into one system. AC can be of two types; Eager associative classification (EAC) and Lazy learning associative classification (LLAC). Two steps are involved in the construction of EAC. Training dataset is used to construct the model in the first step and test dataset is in the second step. The model is validated by using test dataset. EAC generates a large set of rules. During classification, many rules are not useful for classification and some important rules may not be generated. To overcome this issue, LLAC (non- eager) is proposed by Adriano Veloso et al. [2] where focus is given to the attributes of the given test cases. By this way, the chance of generating useful rules is increased. The processing of data is delayed by LLAC until the test instance requires classification. There are two phases in LLAC: 1. Subset Generation and 2. Subset evaluation. Test dataset is used to generate the subsets. In the second phase, subsets generated are assessed. Probability, support and confidence measures are used for the evaluation. For the provided test instance, the highest probability class label is allocated.

Ranking, pruning and producing a huge number of rules are very challenging process.

Lazy learning associative classification is implemented to address these problems. It postpones data processing until the test query requires classification. Merschmann L and Plastino [8] and [9] implemented lazy learning system using the Highest Subset Probability (HiSP) algorithm. Adriano Veloso et al. have proposed numerous lazy classifiers which have enhanced the accuracy of the classification. The attribute, which has scored highest information gain is selected for the construction of rules in LACI as proposed by Syed Ibrahim et al. [10] and Zhang X [11]. Preeti et al. [12] have proposed various ranking learning AC attribute rankings methods and improved classification accuracy. In [13], Variance; a measure of the spread of the data items of number N, from the mean is used. Variance ranking is used to handle the imbalance problem. Rank Order Similarity (ROS) technique is used for similarity measurement. Decision Tree is used in [14] for the early prediction of the Diabetes. The compact subset generation method is used in ChiSC-AC method [15]. Multiple algorithms and methods are presented by multiple authors, but still there is the possibility to improve the accuracy of the classifier.

This paper is divided into five parts, in which the second section addresses a literature survey of the classification and kNN system. Section 3 deals with proposed method and pseudocode. Discussion of the experimental finding is presented in section 4, and the final section addresses the conclusion.

Eager associative classification generates a large set of rules. If one instance matches several rules, which classes are different, that is called rule conflict problem. On the other hand, kNN classifies test instance based upon the nearest training samples in the dataset. But it also suffers from two deficiencies: a. Its efficiency is less because it has to compute the similarity/ distance between test and every single training instance and b. Accuracy is totally dependent how to determine the optimum value for k.

To overcome above said issues, Associative classification with kNN (AC-kNN) is proposed by Zaixiang Huang et al. [3]. By integrating these two concepts, it can resolve the rule conflict and low efficiency problem. Rules are developed in AC-kNN method based on the AC system such as CARS discovery, rule ranking, and pruning. kNN applies for the purpose of classification. Haida et al. [4] suggested HAC that would combine the AC algorithm with the NB algorithm to boost classification accuracy and to minimize the memory space and execution time.

Eager Associative Classification with kNN (EAC-kNN) resolves the issue of rule conflict and low efficiency, but Eager associative classification generates many rules. Some of them are not useful and some important rules are not generated. To address this issue, LLAC with Fixed kNN (LLAC_kNN) [5] is proposed in our previous work. Lazy learning associative classification (LLAC) delays the data processing until classification is required for a query. It

generates rules specific to each test instance. In LLAC_kNN method, LLAC is applied to the dataset to generate the subset in the first step. In the second step, kNN is applied for the classification. By this way, better accuracy is achieved, but still suffers from the sensitivity problem of choosing k for KNN. To address the sensitivity problem of kNN, Dudani [6] proposed Weighted k Nearest Neighbor (WkNN), in that nearest neighbor gets the heavy weight and farthest neighbor gets the light weight according to their distance from the test instance. This WkNN method is used in the proposed work.

For the further improvement in performance of LLAC_FkNN, This paper proposes LLAC with Weighed kNN (LLAC_WkNN). In this, subset is generated based on LLAC method. Then the distance is calculated between the generated subset and the test instance. Based on the distance, weight is calculated and assigned to all the k neighbors. In WkNN method, outliers are also getting high weights. To solve this issue Jianping Gou et al. [7] proposed Dual Weighted kNN (DWkNN) method. In that dual weight is assigned in place of the single weight. By applying dual weight, in between outliers get the lesser weight that leads to an accurate classifier. Thus, this paper proposes LLAC_DWkNN method to reduce the weight of outliers. The proposed work has improved the accuracy of the existing system and traditional systems.

At first, Liu B et al. [1] integrated classification and association rule mining (ARM) and referred to as associative classification (AC). AC constructs Class Association Rules (CARS), in that left-hand side is rule and right-hand side is restricted to a class label. AC has been applied fruitfully in several tasks of classification such as detecting fraud, spam filtering, diagnosing cancer, and so on. There are two forms of associative classification, EAC and LLAC.

There are two stages to the Eager Associative Classification (EAC). EAC uses association rule mining in the first step to build the rules on class association. In phase two, the classifier is constructed, based on the CARs created from phase one. The EAC offers enhanced accuracy, but some drawbacks exist. Ranking, pruning and producing a huge number of rules are very challenging process.

Lazy learning associative classification is implemented to address these problems. It postpones data processing until the test query requires classification. Merschmann L and Plastino [8] and [9] implemented lazy learning system using the Highest Subset Probability (HiSP) algorithm. Adriano Veloso et al. have proposed numerous lazy classifiers which have enhanced the accuracy of the classification. The attribute, which has scored highest information gain is selected for the construction of rules in LACI as proposed by Syed Ibrahim et al. [10] and Zhang X [11]. Preeti et al. [12] have proposed various ranking learning AC attribute rankings methods and improved classification accuracy. In [13], Variance; a measure of the spread of the data items of number N, from the mean is used. Variance ranking is used to handle the imbalance problem. Rank Order Similarity (ROS)

technique is used for similarity measurement. Decision Tree is used in [14] for the early prediction of the Diabetes. The compact subset generation method is used in ChiSC-AC method [15]. Multiple algorithms and methods are presented by multiple authors, but still there is the possibility to improve the accuracy of the classifier.

This paper is divided into five parts, in which the second section addresses a literature survey of the classification and kNN system. Section 3 deals with proposed method and pseudocode. Discussion of the experimental finding is presented in section 4, and the final section addresses the conclusion.

3 k Nearest Neighbor Algorithm

In data mining, the Nearest Neighbor rule (NN) proposed by Cover [16], is one of the earliest and easiest classification method. In NN, one test instance is given as input along with the training dataset. It has to identify a point from the training dataset, which is closest to the given test instance and assigns its class label to it.

In kNN [17], As input, a training dataset and a test instance are given. The value of k is determined. The algorithm finds the k nearest neighbors of the test instance from the training dataset by calculating the Euclidean distance as given in Eq. (1) between test and all the training instances. The label of the majority class is assigned to the test query given.

Euclidean distance:
$$dist(t,q) = \sqrt{(q_1 - t_1)^2 + (q_2 - t_2)^2 + \cdots + (q_n - t_n)^2} \quad (1)$$

In kNN algorithm, selecting k is difficult task. Classification result directly depends on the selection of k. If k is very small, accuracy may decrease because the prediction will be done based on very few instances. If k is very large, outliers and noise may interrupt the classification result. To overcome this issue, weight is assigned to the neighbors.

Consider the following scenario in Figure 1. Tringle shape is the test instance. There are two classes; circle and star. Task is to assign a class label for the test instance. If k nearest neighbor is applied, then test instance will be assigned as circle class because of the majority voting. But if weight is calculated and assigned, then majority weighted voting is star class. Weight is also an important factor for classification.
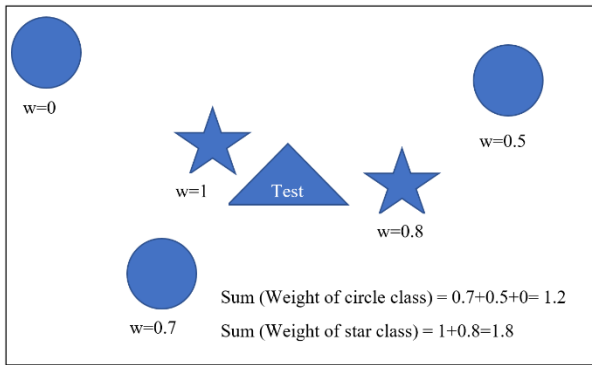
Fig. 1. Illustration of weighted kNN.

Weighted k nearest neighbor (WkNN) is introduced by Dudani, with the concept of assigning more weight to the nearest neighbor and the less weight to the farthest one. After calculating the distance (d1, d2,...,dk) between test instance and training instances, weight (w1, w2,…,wk) is calculated for each neighbor by using the weight formula given in Eq. (2).

$$w_i = \begin{cases} \frac{d_k - d_i}{d_k - d_1}, & if\, d_k \neq d_i \\ 1, & if\, d_k = d_i \end{cases}.$$

(2)

After calculating the weight for each neighbor, class label is assigned to the given test instance based on the majority weighted voting. If the outliers are present in the neighbors, they are also getting good weights and then the performance of the classifier decreases. To overcome this issue, J. Gou et al. proposed dual distance weighted k nearest neighbor algorithm (DWkNN). In this, dual weight is appointed by multiplying a new weight to the original weight as shown in equation II. By applying dual weight, each neighbor gets lesser weight, so the outliers also get lesser weight and hence the classification performance improves. Dual distance weight formula is given in Eq. (3).

$$\mathbf{w_i} = \begin{cases} \frac{d_k - d_i}{d_k - d_1} \times \frac{d_k + d_i}{d_k + d_1}, & \textbf{if } \mathbf{d_k \neq d_i} \\ \mathbf{1}, & \textbf{if } \mathbf{d_k = d_i} \end{cases}$$

(3)

4 Proposed Work

There are two methods proposed in this paper, first is Lazy Learning Associative Classification with Weighted kNN (LLAC_WkNN) and second is Lazy Learning Associative Classification with Dual Weighted kNN (LLAC_DWkNN).

4.1 LLAC_WKNN

The LLAC has been employed by many researchers in various areas and has achieved considerable success when is combined with other algorithms. In machine learning area, this combination or hybridization is to

invent new algorithms as a solution necessary in overcoming certain shortcomings observed during the use of classical algorithms. These new algorithms could have new promising features for solving some classification problems. LLAC generates a large number of subsets of rules that degrades the performance of the classifier and kNN is very sensitive to the selection of k. To address these issues, this paper proposes LLAC_WkNN algorithm. In the proposed algorithm, subset is generated based on the LLAC method and the weighted kNN is applied for the classification. The proposed framework is given in Figure 2.

In Algorithm 1, First, subset of test query 't' from the training instances is generated by using information gain attribute. In general, kNN uses Euclidean distance as the distance metric as data is numerical in the dataset, but in this research, data is categorical in the dataset, so similarity function is used to find the similar transaction and arranged in descending order, so similar transaction will be on the top. Similarity is calculated between test query 't' and generated subset 'S' and stored in 'sim'. All the instances of subsets are arranged in descending order according to the similarity value. The more similarity value will be, the more nearest neighbor will be for the test query. Lall and Sharama mentioned that setting a suitable k should satisfy k = √n for training datasets with sample size larger than 100 [18]. Top k nearest neighbors are selected with the class labels and stored in 'x'. Weight 'W= (w1,…,wk)' is calculated for all the k instances. And based on the majority weighted voting, for the given test instance class label is predicted.
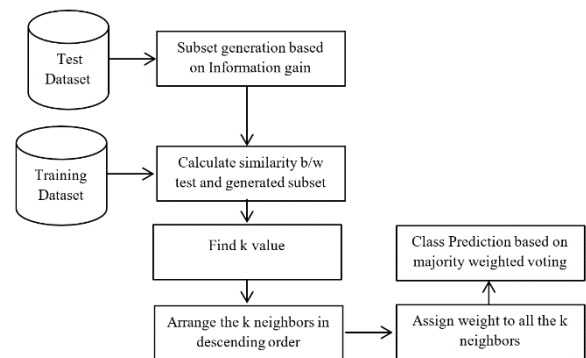

Fig. 2. Proposed Framework for LLAC_WkNN.

Algorithm 1 The proposed LLAC_WkNN algorithm

Input: X = Training Dataset (n X m), Y = Class labels of training dataset, t = test query
Output: c ∈ Y; Predicted class label of the test query.
Step1: Generate subsets for the test query 't' by using information gain attribute.
For i=1 to n do

    If (attribute of test query = = attribute of training instance i)

Subset S = training instance i

End if

End for

Step 2: Calculate the similarity between test query 't' and generated subset 'S'

$$Sim(t,s) = \sum_{i=1}^{m} \partial(t,s_i) \Big/ m.$$

Where, $\partial(t,s_i) = \begin{cases} = 1 \,, if(t = s_i) \\ = 0 \,, otherwise \end{cases}.$

And, m= number of attributes

Step 3: Sort the similarity 'sim' in descending order

Step 4: Find k and select top k nearset neighbors of the test query 't'.

For i = 1 to k do

xi = si,ci

end for

Step 5: Calculate the weight for all the nearest neighbors. $W = (w_1, \dots w_k)$

For i = 1 to k do

If $sim_k = sim_1$

$w_i = 1$

else

$$w_i = \frac{sim_k - sim_i}{sim_k - sim_1}$$

end if

end for

Step 6: Assign a majority weighted voting class label 'c' to the test query 't'.

4.2 LLAC_DWKNN

If the outlier exists in the dataset, then they are also getting weights, that leads to poor performance of the system. To overcome this issue of WkNN, Dual distance weighted k nearest neighbor (DWkNN) algorithm is introduced. This paper is also proposing LLAC_DWkNN, in that dual weight is used in place of weight. The procedure is same as shown in algorithm 1 except the weight formula. Another new weight is getting multiplied with the original weight as shown in Eq. (4).

$$w_i = \frac{sim_k - sim_i}{sim_k - sim_1} \times \frac{sim_k + sim_1}{sim_k + sim_i}$$

(4)

By assigning dual weight to the nearest neighbors, weight is reduced except the nearest one and the farthest one. It helps improving the accuracy of the classifier by assigning lesser weight to the outliers.

5 Result and Discussion

A total of 9 different datasets are used to test the proposed system. The main datasets are taken from the University of California (UCI Repository) [19]. All experiments are conducted on Intel (R) Core (TM) i3-2120 processor, a 3.3 GHz clock speed and 4 GB RAM. Tenfold cross validation method was used where each 10% of data in turn is used for testing and

the remaining 90% for training. Table 1 presents the short illustration of the dataset as well as an abbreviation of dataset names. The dataset names are Balance Scale (BS), Breast Cancer (BC), Breast Wisconsin (BW), Credit Approval (CA), Diabetes, Solar Flare (SF), Glass dataset (GD), Ionosphere (Iono) and Iris Plant (IP).

Table 1. Dataset details.

Dataset	#Transactions	#Attributes	#Classes
BS	625	5	3
BC	286	10	2
BW	699	10	2
CA	690	16	2
Diabetes	768	9	2
SF	1389	13	6
GD	214	10	6
Iono	351	35	2
IP	150	5	3

Accuracy computation: The accuracy is determined using the given equation (5). Comparison of accuracy is provided in Table 2.

$$Accuracy = \frac{Number\ of\ correctly\ predicted\ test\ data}{Total\ no\ of\ test\ data}.$$

(5)

The accuracy comparison is shown in Table 2, where dataset name is tabulated in column 1; 2nd column is the traditional method Lazy Associative Classification LAC [2], 3rd, 4th and 5th columns are existing lazy learning methods namely Lazy Associative Classification using Information gain (LACI) [9], Compact Highest Subset Confidence-based Associative Classification (CHiSC-AC) [15] and Attribute ranking based lazy learning AC [12]. The last columns are the proposed Lazy Learning Associative Classification with Weighted kNN (LLAC_WkNN) and Lazy Learning Associative classification with Dual Weighted kNN method (LLAC_DWkNN).

It can be seen in the comparison result that the proposed LLAC-WkNN system is 10.17% better than LAC, 8.23% better than LACI, 3.43% better than CHiSC and 0.40% better than ARBLazyAC respectively. Proposed LLAC-DWkNN is 13.97% better than LAC, 11.97% better than LACI, 7.00%

better than CHiSC and 3.91% better than ARBLazyAC respectively. Precision, Recall and F1- measure for LLAC_DWkNN is shown in Table 3.

The Win/Draw/Loss table is shown in Table 4. When comparing proposed method with the existing LAC method, the proposed method LLAC_WkNN has improved the classification accuracy for 8 datasets and worse for 1 dataset. When comparing LLAC_DWkNN with LACI, proposed method's accuracy is better for all the 9 datasets.

Table 2. Accuracy comparison with traditional and existing systems.

Datasets	Traditional Classifier	Lazy Learning Associative Classifier (Existing)			Proposed Methods	
	LAC	LACI	CHiSC-AC	ARB Lazy AC	LLAC_ WkNN	LLAC_ DWkNN
BS	69.29	70.32	71.27	76.88	75.58	76.00
BC	66.48	67.86	85.22	73.44	75.86	88.06
BW	93.70	88.57	90.63	94.36	94.28	98.57
CA	76.48	76.81	83.33	89.20	85.50	89.50
Diabetes	69.10	68.83	76.75	75.13	75.32	76.32
SF	81.58	84.71	84.71	85.44	86.99	87.23
GD	57.94	59.09	60.91	65.01	82.50	82.50
Iono	82.29	94.44	96.29	89.56	94.71	95.85
IP	96.67	95.33	87.22	97.40	93.33	96.50

Table 3. Precision, Recall and F-1 Measure

Dataset	Precision	Recall	F1-Measure
BS	0.64	0.71	0.67
BC	0.77	0.76	0.74
BW	0.97	0.94	0.93
CA	0.90	0.90	0.90
Diabetes	0.74	0.75	0.73
SF	0.90	0.90	0.90
GD	0.61	0.73	0.63
Iono	0.90	0.90	0.90
IP	0.95	0.93	0.94

Table 4. Win/Draw/Loss Table

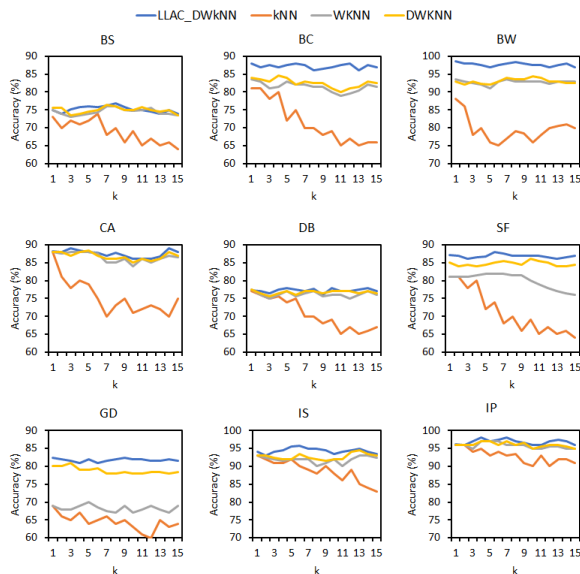| Table Head	Existing Methods			
	LAC	LACI	CHiSC-AC	ARB Lazy AC
LLAC_WkNN	8/0/1	8/0/1	6/0/3	5/0/4
LLAC_DWkNN	8/0/1	9/0/0	7/0/2	7/0/2

Figure 3 The average accuracy v/s the neighborhood size k for different datasets

Figure 3 shows the average classification result of the proposed LLAC_DWkNN and existing methods DWkNN, WkNN and kNN for the explanation of the sensitivity issue of neighborhood size k. It can be clearly seen that proposed method is performing better than the other related method, mostly when k is large. It is robust to the size of k. Accuracy is not fluctuating when the k changes and showing good performance. The Evaluation results of 9 different datasets from the UC Irvine Machine Learning Repository have proven that the proposed approach achieved higher classification accuracy when compared with competing methods.The average accuracy v/s the neighborhood size k for different datasets is shown in Figure 3. It can be seen that the value of k is not affecting the accuracy for all the 9 datasets. Accuracy is constant irrespective of the value of k. It shows that the system is robust.

Weighted kNN is applied for the classification. Weighted kNN picks only k nearest neighbors among the all generated subsets. By this way, accuracy is improved, and proposed system is robust. The value of k doesn't affect the accuracy. It has been observed that the proposed framework gives the excellent result when compared with the existing one.

6 Conclusion

In this paper, two novel approaches LLAC_WkNN and LLAC_DWkNN are proposed to improve the performance of Lazy learning associative classification. The key idea of the proposed methods is to integrate the two concepts in order to get the advantage of both the methods to improve the classification accuracy and to get rid of the sensitivity issue of neighborhood size k. First lazy learning method is applied to the dataset for subset generation and then weighted kNN is applied for the classification. Weighted kNN picks only k nearest

neighbors among the all generated subsets. By this way, accuracy is improved, and proposed system is robust. The value of k doesn't affect the accuracy. The improved outcome directs that the proposed method of uniting LLAC and WkNN is an effective algorithm.

References

1. B. Liu, W. Hsu and M. Yiming, "Integrating Classification and Association Rule Mining," in 4th International Conference on Knowledge Discovery Data Mining (KDD), (1998).

2. A. Veloso, W. Meira Jr. and M. J. Zaki, "Lazy Associative Classification," in Sixth International Conference on Data Mining (ICDM'06), Hong Kong, China, (2006).

3. Z. Huang, Z. Zhou and T. He, "Associative Classification With Knn," Journal of Theoretical and Applied Information Technology, vol. 49, no. 3, pp. 1013-1019, (2013).

4. W. Hadi, Q. A. Al-Radaideh and S. Alhawari, "Integrating associative rule-based classification with Naïve Bayes for text classification," Applied Soft Computing, vol. 69, pp. 344-356, (2018).

5. P. Tamrakar, S. S. Roy, B. Satapathy and I. S. P. Syed, "Integration of lazy learning associative classification with kNN algorithm," in Vision Towards Emerging Trends in Communication and Networking (ViTECoN), Vellore, (2019).

6. S. A. Dudani, "The Distance-Weighted k-Nearest-Neighbor Rule," IEEE Transactions on Systems, Man, and Cybernetics, Vols. SMC-6, no. 4, pp. 325-327, (1976).

7. J. Gou, L. Du, Y. Zhang and T. Xiong, "A New Distance-weighted k-nearest Neighbor Classifier," Journal of Information & Computational Science, vol. 9, no. 6, pp. 1429-1436, (2012).

8. L. Merschmann and A. Plastino, "A Lazy Data Mining Approach for Protein Classification," IEEE Transactions On Nanobioscience, vol. 6, no. 1, pp. 36-42, (2007).

9. L. Merschmann and A. Plastino, "HiSP-GC: A Classification Method Based on Probabilistic Analysis of Patterns," Journal of Information and Data Management, vol. 1, no. 3, pp. 423-438, (2010).

10. S. P. S. Ibrahim, C. K. R. and C. J. K. Kanthasamy, "LACI: Lazy Associative Classification using Information Gain," IACSIT International Journal of Engineering and Technology, vol. 4, no. 1, pp. 1-6, (2012).

11. X. Zhang, G. Chen and Q. Wei , "Building a highly-compact and accurate associative classifier," Applied Intelligence, vol. 34, no. 1, pp. 74-86, (2011).

12. P. Tamrakar and S. P. S. Ibrahim, "Attribute ranking based lazy learning associative classification," ARPN Journal of Engineering and

Applied Sciences, vol. 13, no. 11, pp. 3698-3705, (2018).

13. S. H. Ebenuwa, M. S. Sharif, M. Alazab and A. Al-Nemrat, "Variance Ranking Attributes Selection Techniques for Binary Classification Problem in Imbalance Data," IEEE Access, vol. 7, pp. 24649-24666, (2019).

14. J. Thomas, A. Joseph, I. Johnson and J. Thomas, "Machine Learning Approach For Diabetes Prediction," International Journal of Information Systems and Computer Sciences, vol. 8, no. 2, pp. 55-58, (2019).

15. S. P. S. Ibrahim, K. R. Chandran and C. J. K. Kanthasamy, "Chisc-AC: compact highest subset confidence-based associative classification," Data Science Journal, vol. 13, pp. 127-137, (2014).

16. T. Cover and P. Hart, "Nearest neighbor pattern classification," IEEE Transactions on Information Theory,, vol. 13, no. 1, pp. 21-27, (1967).

17. E. Fix and J. L. Hodges Jr., "Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties," International Statistical Review, vol. 57, no. 3, pp. 238-247, (1989).

18. U. Lall and A. Sharma, "A nearest neighbor bootstrap for resampling hydrologic time series," Water Resources Research, vol. 32, no. 3, pp. 679-693, (1996).

19. D. Dua and C. Graff, "UC Irvine Machine Learning Repository," (2019).