Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 165 (2019) 370–376

Procedia Computer Science

www.elsevier.com/locate/procedia

INTERNATIONAL CONFERENCE ON RECENT TRENDS IN ADVANCED COMPUTING 2019, ICRTAC 2019

Comparative Study of different Lazy Learning Associative Classification Methods

Preeti Tamrakar, Syed Ibrahim SP*

Vellore Institute of Technology, Chennai,Tamil Nadu, India

Abstract

Lazy Learning Associative Classification (LLAC) is a promising approach in the field of data mining. It is one of the associative classification methods in which it delays the processing of training datasets until it receives the test instance for the class prediction. Lazy learning associative classification can be constructed in two phases. Subset generation is the first phase and the subset evaluation is the second phase. In the past decades, many lazy learning associative classification methods have been proposed. These algorithms utilize several different methods for subset generation and subset evaluation. This paper focuses on comparative study of different lazy learning associative classification methods.

Peer-review under responsibility of the scientific committee of the INTERNATIONAL CONFERENCE ON RECENT TRENDS IN ADVANCED COMPUTING 2019.

Keywords: Associative classification ; Lazy learning ; Subset generation ; Subset evaluation.

1. Introduction

We are living in the digital age where everything is processed digitally and we are generating a numerous amount of data. Storing all the generated data, handling the data and extracting the useful information are challenging task.

Knowledge discovery also known as Data mining, deals with extracting information from the pool of data by using some algorithms or techniques. Associative classification [1] employs with two important data mining task,

* Corresponding author. Tel.: +9-900-319-3532
E-mail address: syedibrahim.sp@vit.ac.in

1877-0509
Peer-review under responsibility of the scientific committee of the INTERNATIONAL CONFERENCE ON RECENT TRENDS IN ADVANCED COMPUTING 2019.
10.1016/j.procs.2020.01.036

those are association rule mining (ARM) and classification. ARM utilizes unsupervised learning and extracts the highly correlated features. Classification utilizes supervised learning, in that classifier learns from the training dataset and generates a set of classification rules. Those generated rules are utilized for the classification of the test dataset to the predefined class label.

Associative classification (AC) can be characterized into two types; Eager associative classification (EAC) and Lazy learning associative classification (LLAC). Eager associative classification is constructed by using two steps. By using the training dataset, model is constructed in the first step and by using test datasets the model is validated in the second step.

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, Lazy learning associative classification (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.

LLAC [3] postpones the processing of data until the test query needs classification. Test dataset is used to generate the subsets. Generated subsets are evaluated by using support, confidence or probability with the help of training datasets. Based on that, class label is predicted for the test query.

These lazy associative classification methods provide higher accuracy of the classifier, but leads to high computation cost. Various information gain based attribute selection methods are proposed to reduce the computation cost.

The aim of this paper is to compare and study the different ultra-modern Lazy learning Associative Classification techniques.

The paper is arranged as follows: The Lazy learning Associative Classification problem and its main steps are given in Section 2. Different methods used to generate and evaluate the subsets are surveyed in Sections 3. Final section presents the conclusions of this study.

2. Lazy Learning Associative Classification

Eager associative classification consumes more time on generating the high quality rules, ranking and pruning. To avoid these problems, lazy learning associative classification method came into existence. Lazy learning associative classification postpones building the classification model until a query is given. Lazy learning classification systems can concurrently handle numerous problems and solve successfully with modification in the problem area. It best suits for large datasets with few attributes. It focuses on the attributes of the given test tuple. Lazy learning associative systems are applicable where training datasets are frequently updated. Fig. 1 shows the general steps used in the lazy learning associative classification method. Some lazy learning associative classifications are explained here:

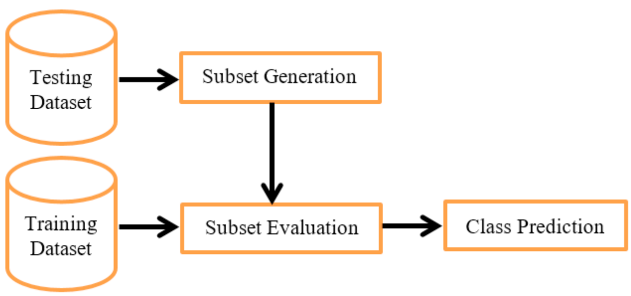

Fig. 1: Lazy Learning Associative Classification Steps

2.1 Attribute method based LLAC

Attribute or feature is an important aspect of dataset. Dataset consists of many attributes. All the attributes does not contribute to classify the test instance. Some attributes are irrelevant and redundant. So, we need attribute selection methods to select some important and relevant attribute that directly impart for the classification.

Information gain, correlation attribute and gain ratio are some attribute selection methods that have been used for the research.

2.1.1 Attribute Selection based LLAC

In this paper, three attribute selection methods have been used to construct lazy learning associative classifier. Correlation attribute measures how strongly one attribute implies the other, based on the available data. A feature can be considered as good if correlation is high with the class and low with other features. The information gain is used to identify the more informattive attribute from data set. The gain ratio is the ratio of the information gain to the split information 1. Correlation attribute based LLAC (CA-LLAC), 2. Information gain based LLAC (IG-LLAC) and 3. Gain ratio based LLAC (GR-LLAC). Lazy method starts processing when it gets test instance. Attribute selection method is used to find the best attribute among all the attribute of the test instance. Based on that best attribute, subset is generated. For subset evaluation, the probability is calculated for each subset. Maximum probability class is assigned to the given test instance.

2.1.2 Attribute Ranking based LLAC

In attribute ranking based approach [4], three methods are there. 1. Correlation attribute rank based LLAC (CA_R-LLAC), 2. Information gain rank based LLAC (IG_R-LLAC) and 3. Gain ratio rank based LLAC (GR_R-LLAC). Previous attribute selection based LLAC methods can be further improved by using the ranking mechanism. In place of choosing only the best attribute, in this method, rank is calculated for all the attribute. Subset is generated based on the rank of all the attributes. Each generated subsets are evaluated based on the probability values.

2.2 LLAC with Hybrid Feature Selection Method

Forward Selection and Backward Elimination are two of the feature selection approaches. Both of these methods are iterative in nature. Forward selection begins with having zero feature in the model and in every step, the feature which improves the model are included; Whereas backward elimination considers all the features and removes the least important feature at every step basis on the improvement in performance and stops the process when no improvement is noticed.

In this method, the forward selection and backward elimination methods are integrated to achieve the advantages of both the methods and called hybrid feature selection method [5]. In this, two variables have been initialized as 'a'=1 and 'b'= m, where m is the total number of attribute. Two feature subsets are created, namely 'S1' and 'S2'. One subset for adding the feature like forward selection and the other subset is for removing the feature like backward elimination. After each iteration 'a' value is incremented and 'b' value is decremented until the number of features in both the subsets 'S1' and 'S2' are equal. Downward closure property is applied to remove the infrequent features. Based on the subsets generated in 'S1' and 'S2', test data is classified by one of the given classes.

2.3 LLAC with k nearest neighbors algorithms

k nearest neighbor is one of the best classification algorithm in statistics and data mining due to easy implementation and noteworthy performance. Algorithm stores the training dataset and class label is predicted for the new test instance based on some similarity measures or distance measures. It calculates the distance between test instance and each training instance and based on the distance, finds the nearest neighbors. Majority class label of the nearest neighbors is assigned to the give test instance.

Different integrating methods have been proposed in past decades like AC-kNN [6], in that associative classification is integrated with k nearest neighbor algorithm and in [7] Associative classification is integrated with Naïve Bayes for the system performance improvement. Rank Order Similarity (ROS) technique is used for similarity measurement in [8]. Decision Tree is used for the early prediction of the Diabetes [9].

In LLAC-kNN method, kNN is applied with LLAC. Subset is generated based on LLAC method. Evaluation of subset and prediction of the class label for test instance is calculated based on kNN algorithm.

kNN algorithm is of two types. 1. Fixed kNN: in this, k value is predefined or fixed by an expert for all the different test instances. 2. Varied kNN: in this, based on different methods like correlation matrix or cross validation method, different k values are generated for the different test instances. Based on these two types of kNN, following methods are presented below:

2.3.1 LLAC with Fixed kNN

In this LLAC_FkNN method, dataset is splited into two parts; training dataset and test dataset. Subset is generated based on the LLAC method. The similarity between the test instance ‘t’ and each generated subset ‘s’ is calculated based on the Eq. (I). If number of training instances is n, then the k value is calculated as $\sqrt{n}$. k nearest neighbors are chosen based on the similarity value. Majority class label is assigned to the given test instance.

$$Similarity(t,s) = \sum_{i=1}^{m} \partial(t_i, s_i) / m \tag{I}$$

Where,

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

And m = number of attribute

2.3.2 LLAC with Varied kNN

For the different test instances, one fixed k value is not suitable in real life. Various methods have been presented, where different k values are generated for different test instances [10], [11]. In this LLAC_VkNN method, different and optimal k values are generated by using the similarity formula given in equation 1 and train to train reconstruction matrix is created. Suppose there are 5 instances in the training dataset and R is the reconstruction matrix as shown in Fig. 2. The correlation of first training instance with all the 5 training instances is given in the first column. Diagonal values are 1, because of the self correlation.

$$R = \begin{bmatrix} 1 & .75 & .75 & .5 & .25 \\ .75 & 1 & .5 & .25 & 0 \\ .75 & .5 & 1 & .5 & .25 \\ .5 & .25 & .5 & 1 & .75 \\ .25 & 0 & .25 & .75 & 1 \end{bmatrix}$$

Fig. 2: Reconstruction Matrix

Suppose threshold is 0.75, then the values which are equal to or greater that 0.75 is counted. So, the k-value for 1st training instance is 3, because 3 values are satisfying the threshold value. The k-values are stored in the k-hash table as shown in Fig. 3. For the 5 training instances, k-values are stored respectively.

1	→	3
2	→	2
3	→	2
4	→	2
5	→	2

Fig. 3: k-hash Table

Different and optimal k values are stored in the hash table. Now there is a test instance, which class label is to be predicted. Similarity between the test and each training instance is calculated. Training instance, which has the highest similarity, is chosen. From the k hash table, k value is taken for that instance. The majority vote of the k nearest neighbors’s class label is assigned to the test instance.

2.4 LLAC with weighted nearest neighbors algorithms

2.4.1 LLAC with WkNN

WkNN [12] is proposed to improve the performance of kNN by introducing distance weighted function. The weights are distributed linearly, where weight 1 is given to nearest and 0 to furthest. In this method, ‘k’ is determined and the distance between query and training instances ‘d’ is calculated. Training dataset is arranged in ascending order of distance. Weight is calculated using the distance weighted function that is given in Eq (II) where w_i is the weight of the i^{th} nearest neighbor of the given test instance:

$$w_i = \begin{cases} \dfrac{d_k - d_i}{d_k - d_1}, & if \quad d_k \neq d_1 \\ 1, & if \;\; d_k = d_1 \end{cases} \tag{II}$$

After calculating the weight, query is classified by the majority weighted voting.

In LLAC_WkNN method, the dataset is partitioned into training and testing dataset. Subset is generated based on the LLAC method. The similarity between the test instance ‘t’ and each generated subset ‘s’ is calculated based on the equation 1. K-hash table outputs the optimal k value for the given test instance. K nearest neighbors are chosen based on the similarity value. Weight is assigned to all the nearest neighbors using the weight function as given in Eq. (II). The majority weighted class label is assigned to the given test instance.

2.4.2 LLAC with DWkNN

DWkNN [13] is the extension of WkNN by calculating the dual weight for the nearest neighbors. Although WkNN is not much sensitive to the selection of k than kNN, then also robustness to the change of neighborhood size k is still unsatisfactory. WkNN performs one step ahead than other weighted method for kNN through their experimental comparisons in different cases. However, the performance of WkNN degrades due to outliers, especially for small data sample. The performance of kNN and WkNN are also affected due to irregular class distribution.

In DWkNN, the dual distance-weight is used in place of the corresponding weight for the neighbor. The dual weight is calculated from eq (III), where the original weight is multiplied by another new weight. The DWkNN reduces the weight of intermediate neighbors without effecting nearest and furthest neighbors, which improves the classification performance.

$$w_i = \begin{cases} \dfrac{d_k - d_i}{d_k - d_1} * \dfrac{d_k + d_1}{d_k + d_i}, & if \quad d_k \neq d_1 \\ 1, & if \;\; d_k = d_1 \end{cases} \tag{III}$$

In this, LLAC method and DWkNN method are combined, where subsets are generated using LLAC method and classification is done by DWkNN method. The combined method yields better accuracy.

3. Result and discussion

All the different lazy learning associative systems are tested and validated with the help of the standard datasets taken from the California University (UCI Repository) [14]. All the experiments are performed on Intel (R) Core (TM) i3-2120 processor, a clock speed 3.3 GHz and RAM 4 GB. Table 1 presents the short illustration of the dataset as well as an abbreviation of dataset names. The dataset names are Balance Scale, Breast Cancer, Breast Wisconsin, Credit Approval, Diabetes, Solar Flare, Glass dataset, Ionosphere and Iris Plant.

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 calculated from the given Eq. (IV). Accuracy comparison is shown in Table 2.

$$Accuracy = \frac{Number\ of\ correctly\ predicted\ test\ data}{Total\ no\ of\ test\ data} \qquad \text{(IV)}$$

Table 2: Accuracy Comparison

Dataset Names	Lazy Learning Associative Classification Systems										
	CA-LLAC	IG-LLAC	GR-LLAC	CA_R-LLAC	IG_R-LLAC	GR_R-LLAC	LLAC_HFS	LLAC_FkNN	LLAC_VkNN	LLAC_WkNN	LLAC_DWkNN
BS	60.80	76.8	60.80	54.72	76.88	54.72	**98.41**	75.01	75.32	75.58	76.00
BC	69.76	72.43	69.76	54.72	73.44	67.23	68.96	84.33	89.31	75.86	**88.06**
BW	87.85	93.57	90.71	91.10	94.36	92.53	90.00	94.28	97.14	94.28	**98.57**
CA	85.50	85.50	85.50	89.19	89.20	89.19	65.21	88.15	89.85	85.50	**89.50**
Diabetes	75.97	75.97	75.97	75.12	75.13	75.12	77.92	77.32	75.32	75.32	**76.32**
SF	83.33	84.71	83.55	84.55	85.25	84.55	85.00	86.01	86.85	86.99	**87.23**
GD	69.76	67.44	58.13	75.55	78.55	76.55	**86.36**	69.89	81.81	82.50	82.50
Iono	94.36	95.77	91.54	91.68	89.56	87.88	88.88	93.33	93.5	94.71	**95.85**
IP	94.33	95.33	94.33	97.40	97.40	94.44	**100**	96.22	96.5	93.33	96.50
Average	*80.18*	*83.05*	*78.92*	*79.33*	*84.41*	*80.24*	*84.52*	*84.95*	*87.28*	*84.90*	*87.83*

4. Conclusion

This paper presents the comparative study of different kinds of lazy learning associative classification methods. 11 different LLAC methods have been compared and the accuracy comparison result is tabulated here. First 3 methods have used different kinds of feature selection algorithms with LLAC like correlation attribute, gain ratio and information gain. In this, Information gain based LLAC outperformed. In the next 3 methods, ranking mechanism of attribute selection methods have been used. It is observed that ranking mechanism improved the accuracy when compared with the attribute selection mechanism. The next method is LLAC with Hybrid Feature Selection. In this, forward selection and backward elimination are integrated and hybrid method has applied over LLAC for the further improvement of the classifier. In the next two methods, Fixed kNN and Varied kNN methods have been applied to LLAC method. In LLAC_FkNN method, subset is generated by LLAC method and Classification is done by kNN method. Fixed kNN is not applicable in real time scenario. So, Varied kNN is applied with LLAC. In this, different k is generated for the different test instances by using the reconstruction matrix. In the next methods, weighted and dual weighted methods have been applied with LLAC for the further improvement of the classifier. It is clearly visible from the Table 2 that LLAC_DWkNN outperforms the other lazy learning algorithms because the dual weighted mechanism is applied to give the less weight to the outliers and varied kNN is used to generate the different and optimal k value for the fair classification, so the accuracy has improved.

References

1. Liu. B. et.al, (1998) "CBA: Integrating Classification and Association Rule Mining". *Knowledge Discovery and Data Mining conference*. 80–86.
2. Merschmann. L, Plastino. (2007). "A lazy data mining approach for protein classification". *IEEE Transactions on Nanobioscience*, 6 (1):36-42.
3. Adriano Veloso et.al. (2006). "Lazy Associative Classification". *6th International Conference on Data Mining (ICDM'06),*645 – 654.
4. Preeti Tamrakar, S.P. Syed Ibrahim (2018). "Attribute ranking based lazy learning associative classification", *ARPN Journal of Engineering and Applied Sciences*, 13 (11): 3698-3705.
5. Preeti Tamrakar, S.P. Syed Ibrahim. (2019) "Lazy Learning Associative Classification with Hybrid Feature Selection", *International Journal of Recent Technology and Engineering*, 8 (1): 299-303.
6. Zaixiang Huang. (2013) "Associative Classification With Knn", *Journal of Theoretical and Applied Information Technology*, 49 (3):1013-1019.
7. Wa'el Hadia, Qasem A. Al-Radaidehb, Samer Alhawari. (2018) "Integrating associative rule-based classification with Naïve Bayes for text classification " *Applied Soft Computing*, 69 :344–356.
8. S H. Ebenuwa, Mhd Saeed Sharif. (2019) "Variance Ranking Attributes Selection Techniques for Binary Classification Problem in Imbalance Data", *IEEE Access*, 7: 24649-24666,.
9. Jensia Thomas, Anumol Joseph, Irene Johnson, Jeena Thomas. (2019) "Machine Learning Approach For Diabetes Prediction", *International Journal of Information Systems and Computer Sciences*, 8(2):55-58.
10. Shichao Zhang et al. (2018). "Efficient kNN Classification With Different Numbers of Nearest Neighbors", *IEEE Transactions on neural networks and learning systems*, 29(5):1774-1785.
11. S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng. (2017) "Learning k for kNN classification," *ACM Transactions on Intelligent Systems and Technology*. 8(3): 1–19.
12. Sahibsingh A. Dudani, (1976). "The Distance-Weighted k-Nearest-Neighbor Rule". *IEEE Transactions on Systems, Man, And Cybernetics*, 6: 325-327.
13. J. Gou et al., (2012). "A New Distance-weighted k-nearest Neighbor Classifier". *Journal of Information & Computational Science* 9 (6) :1429-1436.
14. Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.