



ACHC: Associative Classifier Based on Hierarchical Clustering

Jamolbek Mattiev^{1,2} and Branko Kavšek^{2,3}(✉)

¹ Urgench State University, Khamid Olimjan 14, 220100 Urgench, Uzbekistan
mattiev.jamolbek@urdu.uz, jamolbek.mattiev@famnit.upr.si

² University of Primorska, Glagoljaška 8, 6000 Koper, Slovenia
branko.kavsek@upr.si

³ Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia

Abstract. The size of collected data is increasing and the number of rules generated on those datasets is getting bigger. Producing compact and accurate models is being the most important task of data mining.

In this research work, we develop a new associative classifier – ACHC, that utilizes agglomerative hierarchical clustering as a post-processing step to reduce the number of rules and a new method is proposed in the rule-selection step to increase classification accuracy.

Experimental evaluations show that the ACHC method achieves significantly better results than classical rule learning algorithms in terms of rules on bigger datasets while maintaining classification accuracy on those datasets. More precisely, ACHC achieved the highest (43) result on the average number of rules and the third-highest (84.8%) result in terms of average classification accuracy among 10 classification algorithms.

Keywords: Frequent itemsets · Class association rules · Associative classification · Agglomerative hierarchical clustering · Cluster center

1 Introduction

Since information technologies are developing very rapidly and the amount of collected data is growing, analyzing such big data is an important task of data mining.

The number of rules generated from “real-life” datasets can easily grow very large, which may cause a combinatorial explosion. Therefore, mining association rules from these data and reducing their number to produce compact models for end-users is becoming a crucial data mining task [17].

To overcome this problem and achieve more compact as well as understandable models, rules have to be pruned and/or clustered.

Association [1] and classification rule mining [2, 7, 26] are two important fields of data mining. Classification rule mining aims at building accurate models to forecast the class value of a future object by selecting small subsets of rules, while association rule mining algorithms find all existing rules in a dataset based on some user-specified constraints by exploring the entire search space.

Associative Classification (AC) [5, 12, 14, 18, 21, 22] is another data mining technique that combines classification and association rule mining. The main goal of AC methods is to produce compact, accurate and descriptive models based on association rules. Thus, the performance of AC methods can sometimes be better than some of the traditional classification methods on accuracy, in spite of worse efficiency, because AC methods are sensitive to user-defined parameters such as minimum support and confidence.

Another area in data mining is clustering [13, 25, 29]. Clustering methods can be usually grouped in two groups: partitional clustering [24, 30] that aims at grouping similar objects together by using partitioning techniques, and hierarchical clustering [28], which is a nested sequence of partitions.

In this research work, we present a novel cluster-based associative classification method. Firstly, we describe a new normalized combined distance metric to find the similarity of two class association rules (CARs). Secondly, we cluster the CARs by using a bottom-up approach of hierarchical agglomerative clustering. In this step, we automatically identify the optimal number of clusters. Thirdly, after CARs are clustered, we present a novel method of extracting the “representative” CAR for each cluster. Finally, we develop a compact and accurate associative classification model by including just the representative CARs from each cluster.

The performance of our method (ACHC) is evaluated on 12 selected datasets taken from the UCI ML Repository [6] and compared with 9 popular (associative) classification approaches, such as Decision Table and Naïve Bayes (DTNB) [10], Decision Table (DT) [15], PART (PT) [7], C4.5 [26], CBA [18], Simple Associative Classifier (SA) [20], FURIA (FR) [11], Ripple Down rules (RDR) [27], and J&B [19].

2 Related Work

The first related approach [3] is clustering of association rules based on the k-means clustering algorithm. Similarly to our approach, this approach utilizes the “APRIORI” algorithm [1] to generate the association rules. The key differences between ACHC and this method are (1) it uses a different algorithm in the clustering step, that is, it clusters the rules based on interestingness measures by using the k-means clustering method and (2) a novel distance metric is used to cluster the rules in our method.

The fuzzy clustering algorithm on association rules (FCAR) [4] is proposed based on rule simulation. In this method, researchers aimed to develop a fuzzy clustering method which partitions n objects into k subsets. FCAR uses the “APRIORI” algorithm in the rule-generation part as in our method, but they use a different technique (partitional) in the clustering phase. Another key difference is that they cluster the association rules while we intend to cluster class association rules.

Another related approach [8] is distance-based clustering of association rules. In this method, an “indirect” distance metric (based on CARs support and coverage) is used to group the association rules, while we develop a novel “combined” distance metric (based on direct and indirect measures) and an agglomerative clustering method to cluster the rules.

In [16], authors mine the clusters with association rules. The APRIORI algorithm is used to find strong CARs in the first step and then, it clusters these CARs by using a

hierarchical clustering method as in our method. However, they used a different distance metric based on an “indirect” distance metric (based on coverage probabilities) in the clustering step.

The algorithm, described in this paper extends our previous work from [21] and [22]. In [21], the CMAC algorithm used a direct distance metric in the clustering phase of CARs and the cluster centroid approach to select the representative CAR for each cluster, while in [22] CMAC is compared to two similar algorithms, one using the direct distance metric and covering approach in the representative CAR selection phase and the other using combined (direct and indirect) distance metric with the same covering approach to select the representative CAR. This paper presents the only remaining combination, i.e., using a combined distance metric in the CAR clustering phase and the cluster centroid approach to select the representative CAR for each cluster.

3 Our Proposed Method – ACHC

It is assumed that we have given a relational table consisting of S examples (transactions). Each example is defined by A different attributes and classified into one of the C known classes.

Our main goals and contributions within this research are listed below:

- Developing a novel distance metric to measure the similarity of class association rules;
- Clustering of class association rules by utilizing the hierarchical clustering algorithm and finding the optimal number of clusters for each class value;
- Defining a new method of selecting a representative class association rule for each cluster to represent the compact and accurate classifier;
- Performing an experimental evaluation to show the advantages and disadvantages of the developed algorithm.

3.1 Class Association Rules

This section describes how to produce the strong class association rules. We first need to discover the frequent itemsets and we then generate the class association rules from these frequent itemsets. To generate the frequent itemsets, we apply the minimum support threshold, that is, frequent itemsets satisfied by minimum support constraint are generated in the first step. After generating the frequent itemsets, it is a straightforward approach to discover the class association rules from frequent itemsets. In this step, we apply the minimum confidence threshold to produce strong CARs.

We utilize the APRIORI algorithm to generate the frequent itemsets, because APRIORI is a well-known and frequently used algorithm for association rule mining.

Once we generate the frequent itemsets, our next goal is to produce the strong class association rules which satisfy the minimum confidence constraint. We check the confidence of each rule, if it satisfies the required threshold, then we generate that rule. Confidence of the *rule*: $A \rightarrow B$ is computed as follows:

$$\text{confidence}(\text{rule}) = \frac{\text{support_count}(A \cup B)}{\text{support_count}(A)} \quad (1)$$

Equation (1) describes the confidence of a rule based on the support count of a frequent itemset, where A represents the antecedent (left-hand side of the rule) and B the consequent (class value in the case of CARs) of a rule. Moreover, $support_count(A \cup B)$ is the number of examples in the dataset that match the itemset $A \cup B$, and $support_count(A)$ is the number of examples that match the itemset A . We generate the strong class association rules that satisfy the minimum confidence constraint based on Eq. (1), as follows:

1. All nonempty subsets S of frequent itemset L belonging to class C are generated;
2. For every nonempty subset S of L , output the strong rule R in the form of “ $S \rightarrow C$ ” if, $confidence(R) \geq min_conf$, where min_conf is the minimum confidence threshold.

3.2 The New “combined” Distance Metric

In our previous research [21], we proposed a novel “direct” distance metric and defined its advantages and disadvantages. In this research, we use the previously developed “direct” distance metric and also focus on the “indirect” distance metric based on the Conditional Market-basket Probability (CMBP) [8]. Using a probability estimate for distance computation has many advantages. Probabilities are well understood, are intuitive, and a good measure for further processing, and it is appropriate for rules only with the same consequent. The distance d^{CMBP} between two rules $rule1$ and $rule2$ is the (estimated) probability that one rule does not hold for a basket, given at least one rule holds for the same baskets. This distance is defined as follows:

$$d_{rule1,rule2}^{CMBP} = 1 - \frac{|m(BS_{rule1} \cup BS_{rule2})|}{|m(BS_{rule1})| + |m(BS_{rule2})| - |m(BS_{rule1} \cup BS_{rule2})|}, \quad (2)$$

where BS is both sides of the rule, that is, the itemset for the association rule. $m(BS)$ and $m(BS_{rule1} \cup BS_{rule2})$ denotes the set of transactions (baskets) matched by BS and $(BS_{rule1} \cup BS_{rule2})$ respectively. $lm(BS)$ and $|m(BS_{rule1} \cup BS_{rule2})|$ is the number of such transactions.

With this metric, rules having no common market baskets are at a distance of 1, and rules valid for an identical set of baskets are at a distance of 0.

In this research, we combine the “direct” and “indirect” distance metric to produce a new Weighted and Combined Distance Metric (WCDM). WCDM combines direct measure (rule items) and indirect measure (rule coverage). The weighted distance d^{WCDM} between two rules, $rule1$ and $rule2$ is defined in Eq. (3).

$$d_{rule1,rule2}^{WCDM} = \alpha \times d_{rule1,rule2}^{direct} + (1 - \alpha) \times d_{rule1,rule2}^{indirect}, \quad (3)$$

where α is a weigh balancing parametr in (3). We set $\alpha = 0.5$ parameter in the distance metric developing part, that is, the contribution of direct and indirect is considered equal to make a weighted and balanced distance metric. The resulting distance metric is defined in Eq. (4).

$$d_{rule1,rule2}^{WCB} = 0.5 \times d_{rule1,rule2}^{direct} + 0.5 \times d_{rule1,rule2}^{indirect}. \quad (4)$$

3.3 Clustering

Clustering algorithms are usually split into two groups: partitional and agglomerative hierarchical clustering. In our research, we apply the complete linkage method of agglomerative hierarchical clustering (in the bottom-up fashion) because it is frequently used and more consistent than other algorithms.

In the algorithms based on the bottom-up fashion, every example is considered as a unique cluster at the beginning and the two the nearest clusters are merged in each iteration until all clusters have been merged into a unique cluster.

In the complete linkage (farthest neighbor) method of agglomerative hierarchical clustering, the similarity between two clusters is computed based on the similarity of their most dissimilar examples, that is, the farthest groups are taken as an intra-cluster distance.

Algorithm 1: Computing the natural number of clusters

Input: Cluster heights.

Output: Optimal number of clusters.

```

1: Set Max to 0
2: Set Opt_number_of_cls to 1
3: Set N to length(cls_heights)
4: FOR each index in N DO
5:   IF cls_heights[index]-cls_heights[index-1] > Max THEN
6:     Set Max to cls_heights[index]-cls_heights[index-1]
7:     Set Opt_number_of_cls to N-index
8:   END IF
9: END FOR

```

To cluster the class association rules, we first identify the optimal number of clusters. In this step, we utilize the most-common technique (described in Algorithm 1), where the dendrogram is cut from the point which represents the maximum difference of two consecutive cluster heights.

Algorithm 1 gets cluster heights (distance between two clusters) which are computed during the dendrogram construction process as an input, and it outputs the optimal number of clusters.

N stores the total number of clusters in line 3. For each two consecutive cluster heights (lines 4–9), we compute the differences and store the maximum difference to Opt_number_of_cls parameter.

3.4 Selecting the “Representative” CAR

After clustering the CARs, we define an algorithm (presented in Algorithm 2) to select a “representative” CAR for each cluster. Only representative CARs are included in the final classifier.

Algorithm 2: Selecting the representative class association rule

Input: Class association rules in the cluster;

1. **Computation:** For each CAR in the cluster, we sum up all distances from the selected CAR to all other CARs in cluster;
 2. **Update:** check if the distance of the selected CAR is lower than the representative CAR's distance, then update the selected CAR as a representative CAR.
 3. **Output:** after checking all the CARs in the cluster, return the representative CAR which has the minimum distance.
-

Algorithm 2 describes the method of extracting the representative CAR. Firstly, the distances between the selected CAR and all other CARs are calculated for each CAR. Secondly, we find the CAR which obtains the minimum distance and returns that class association rules as a representative.

Algorithm 3 summarizes all the above-mentioned steps and describes the final ACHC classifier. It gets the training dataset, minimum support as well as confidence thresholds, as input parameters and outputs the compact and accurate associative classification model – ACHC.

Algorithm 3: ACHC associative classifier

Input: A training dataset D , *minimum support*, *minimum confidence*.

Output: ACHC associative classifier.

- 1: $Data := \{\text{all examples in the dataset}\}$, $s := \text{minimum support}$, $c := \text{minimum confidence}$;
 - 2: $F := \{\text{find the frequent itemsets from } Data \text{ which support } \geq s\}$
 - 3: $C := \{\text{Generate the CARs from frequent itemsets } F \text{ which confidence } \geq c \text{ and sort them in confidence and support descending order}\}$;
 - 4: $G := \{\text{Group the CARs } C \text{ based on class label}\}$
 - 5: **FOR** each group in G **DO**
 - 6: Build *distance_matrix* of CARs in the same group;
 - 7: Compute *cluster_heights* by using *hierarchical_clustering*;
 - 8: Compute *optimal_number_of_clusters* N by using *cluster_heights*;
 - 9: Cluster CARs into N clusters by using *hierarchical_clustering*;
 - 10: **FOR** each cluster in N **DO**
 - 11: Extract *representative_CAR*;
 - 12: Add *representative_CAR* to *Associative_Classifier*;
 - 13: **END FOR**
 - 14: **END FOR**
 - 15: **RETURN** *Associative_Classifier*
-

The first three lines generate the strong CARs defined in Sect. 3.1 and sort them by confidence and support descending order due to the following criterion:

If $Rule_1$ and $Rule_2$ represent two CARs, $Rule_1$ is said to have higher rank than $Rule_2$ ($Rule_1 > Rule_2$):

- if and only if, $\text{conf}(Rule_1) > \text{conf}(Rule_2)$ or
- if $\text{conf}(Rule_1) = \text{conf}(Rule_2)$, but $\text{supp}(Rule_1) > \text{supp}(Rule_2)$ or
- if $\text{conf}(Rule_1) = \text{conf}(Rule_2)$ and $\text{supp}(Rule_1) = \text{supp}(Rule_2)$, but $Rule_1$ has fewer items in its left-hand side than $Rule_2$.

Once the CARs are sorted, we group them according to class value (line 4) and then we build the distance matrix (line 6) for each group of CARs (defined in Sect. 3.2) to apply the hierarchical clustering algorithm with complete linkage (For example: if we have 3 class value, we build the model for three group of CARS and merge them at the end). Line 7 finds the cluster heights (distances between clusters) and the natural number of clusters is identified by using the cluster heights (line 8). After determining the optimal number of clusters, the hierarchical clustering algorithm is again utilized to find the cluster of CARs (*Cluster* array stores the list of clustered CARs) in line 9. In the last step (lines 10–12), we select the representative class association rule for every cluster (described in Subsect. 3.4) to produce the final compact and accurate model.

4 Experimental Setting and Results

Our developed algorithm is compared against 9 rule learners in terms of classification accuracy and rules. All associative classifiers were run with default parameters set up by WEKA [9] software. Some parameters (minimum support and confidence) were modified on “imbalanced” datasets to achieve the intended number of rules (at least 10 rules for each class value) for AC methods.

Statistical significance testing is performed based on the paired t-test (significance difference threshold was set to 95%) method. The description of the datasets and input parameters are shown in Table 1.

Moreover, a 10-fold cross-validation assessment technique was employed to represent all experimental results. Table 2 shows the experimental results for classification accuracies (with standard deviations).

Table 2 shows that our proposed associative classifier (ACHC) achieved comparable average accuracies (84.8%) to other classification models on selected datasets. More precisely, ACHC gained the third-highest average accuracy among 10 classification and association rule mining algorithms.

Our developed classifier obtained the best accuracy on “Balance” (except DTNB), “Breast.Can”, “Spect.H”, “Hayes.R” and “Connect4” datasets among all algorithms, while on “Car.Evn” and “Nursery” datasets, ACHC is beaten by all “classical” classification models.

Unexpectedly, standard deviation of all rule-learners was high (this situation happens mainly with imbalanced datasets which affect the rule-generation part of “APRIORI” algorithm) on the “Hayes.R”, “Lypm” and “Connect4” datasets.

It can be seen from Fig. 1 that all classification algorithms achieved almost similar result on average accuracy except DTNB, DT and SA.

Table 3 represents the statistical significance testing (wins/losses counts) on accuracy between ACHC and other classification models. **W**: winning count (our approach was significantly better than the compared algorithm); **L**: losing count (our approach was significantly worse than the selected algorithm); **N**: no statistically significant difference has been detected in the comparison.

Table.1 Description of datasets and AC algorithm parameters

Dataset	# of attributes	# of classes	# of records	<i>Min support</i>	<i>Min confidence</i>	# of analyzed rules
Breast.Can	10	2	286	1%	60%	1000
Balance	5	3	625	1%	50%	218
Car.Evn	7	4	1728	1%	50%	1000
Vote	17	2	435	1%	60%	500
Tic-Tac	10	2	958	1%	60%	3000
Nursery	9	5	12960	0.5%	50%	3000
Hayes.R	6	3	160	0.1%	50%	1000
Lymp	19	4	148	1%	60%	1500
Spect.H	23	2	267	0.5%	50%	3000
Adult	15	2	45221	0.5%	60%	5000
Chess	37	2	3196	0.5%	60%	3000
Connect4	43	3	67557	1%	60%	5000

Table.2 Overall accuracies with standard deviations:

Dataset	DTNB	DT	C4.5	PT	FR	RDR	CBA	SA	J&B	ACHC
Breast.Can	70.4 ±4.1	69.2 ±6.7	75.0 ±6.9	74.0 ±4.0	75.1 ±5.3	71.8 ±5.7	71.9 ±9.8	79.3 ±4.4	80.5 ±4.7	80.6 ±5.1
Balance	81.4 ±8.1	66.7 ±5.0	64.4±4.3	76.2±5.6	76.3±7.6	68.5±4.3	73.2±3.8	74.0±4.1	74.1±2.6	76.5 ± 2.1
Car.Evn	95.4±0.8	91.3±1.7	92.1±1.7	94.3±1.0	91.8±1.1	91.0±1.8	91.2±3.9	86.2±2.1	89.4±1.4	86.9 ± 1.9
Vote	94.7±3.4	94.9±3.7	94.7±4.4	94.8±4.2	94.4±2.8	95.6±4.1	94.4±2.6	94.7±2.3	94.1±1.8	91.8 ± 1.9
Tic-Tac	69.9±2.7	74.4±4.4	85.2±2.7	94.3±3.3	94.1±3.1	94.3±2.9	100.0±0.0	91.7±1.5	95.8±2.0	91.0±1.4
Nursery	94.0±1.5	93.6±1.2	95.4±1.4	96.7±1.7	91.0±1.4	92.5±1.5	92.1±2.4	91.6±1.2	89.6±1.1	89.7 ± 0.7
Hayes.R	75.0±7.2	53.4±8.3	78.7±8.4	73.1±9.7	77.7±8.7	74.3±7.1	75.6±10.9	73.1±6.0	79.3±5.9	80.7 ± 6.0
Lymp	72.9±9.0	72.2±8.3	76.2±8.7	81.7±9.0	80.0±8.2	78.3±7.3	79.0±9.7	73.7±5.1	80.6±5.7	81.5 ± 7.1
Spect.H	79.3±2.7	79.3±1.6	80.0±9.0	80.4±5.6	80.4±2.2	80.4±2.2	79.0±1.6	79.1±2.1	79.7±3.1	80.6 ± 1.1
Adult	73.0±4.1	82.0±2.3	82.4±4.7	82.1±4.7	75.2±3.2	80.8±2.7	81.8±3.4	80.8±2.6	80.8±2.9	81.3 ± 2.4
Chess	93.7±3.0	97.3±3.1	98.9±3.6	98.9±3.1	96.4±2.1	95.8±3.3	95.4±2.9	92.2±3.8	94.6±2.7	97.0 ± 1.6
Connect4	78.8±5.9	76.7±7.7	80.0±6.8	81.0±7.9	80.6±7.1	80.0±6.4	80.9±8.1	78.7±6.0	81.0±5.2	81.0 ± 6.9
Avg(%):	81.5 ± 4.4	79.3 ± 4.5	83.6±5.2	85.6±4.9	84.4±4.4	83.6 ± 4.1	84.5±4.9	82.3±3.4	84.9±3.3	84.8 ± 3.2

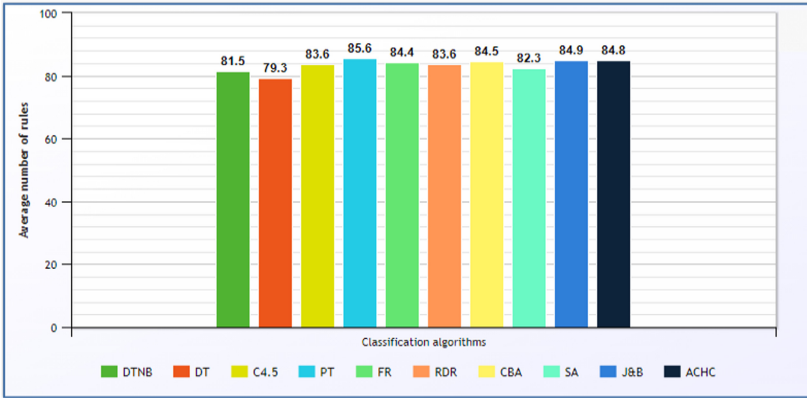


Fig. 1. Comparison between our method and other methods on average accuracy

Table.3 Statistically significant wins/loss counts of ACHC method on accuracy

	DTNB	DT	C4.5	PT	FR	RDR	CBA	SA	J&B
W	7	6	5	2	3	4	4	5	2
L	4	3	4	5	2	4	4	1	3
N	1	3	3	5	7	4	4	6	7

Table 3 illustrates that the performance of the ACHC method on accuracy is better than DTNB, DT and SA methods while this performance is similar or the same to C4.5, FR, J&B, RDR and CBA according to win/losses counts. Table 4 illustrates the size of all classification methods.

Experimental evaluations on the number of rules (Table 4) show that ACHC produced the best result in terms of the average number of rules among 10 rule-learners. Our approach produced a statistically smaller classifier on “Car.Evn” and “Nursery” datasets, although not achieving the best classification accuracies on those datasets.

On “Hayes-root” and “Balance” datasets, ACHC obtained an unexpectedly larger number of rules (due to imbalanced datasets) but it produced accurate classifiers for those datasets.

Figure 2 illustrates that associative classifiers achieved better result than “classical” classification models on the average number of rules. The main reason is that “classical” classification models are sensitive to the size of the dataset.

Table 5 shows that ACHC produced smaller classifiers than DTNB, DT, C4.5, PT, FR, SA, and J&B algorithms on more than 8 datasets out of 12 (by win/losses count).

5 Conclusion and Future Work

Overall, our developed method achieved the highest result on the average number of rules by exhaustively searching the entire example space using constraints and clustering while maintaining a classification accuracy that was comparable to state-of-the-art rule-learning classification algorithms. Experimental evaluations showed that ACHC

Table.4 Number of CARs

Dataset	DTNB	DT	C4.5	PT	FR	RDR	CBA	SA	J&B	ACHC
Breast.Can	122	22	10	20	13	13	63	20	47	9
Balance	31	35	35	27	44	22	77	45	79	79
Car.Evn	144	432	123	62	100	119	72	160	41	32
Vote	270	24	11	8	17	7	22	30	13	6
Tic-Tac	258	121	88	37	21	13	23	60	14	17
Nursery	1240	804	301	172	288	141	141	175	109	80
Hayes.R	5	8	22	14	11	10	34	45	34	80
Lymp	129	19	20	10	17	11	23	60	29	7
Spect.H	145	2	9	13	17	12	4	50	11	5
Adult	737	1571	279	571	150	175	126	130	97	88
Chess	507	101	31	29	29	30	12	120	24	17
Connect4	3826	4952	3973	3973	403	341	349	600	273	102
Avg(%):	618	674	409	411	93	75	79	125	64	43

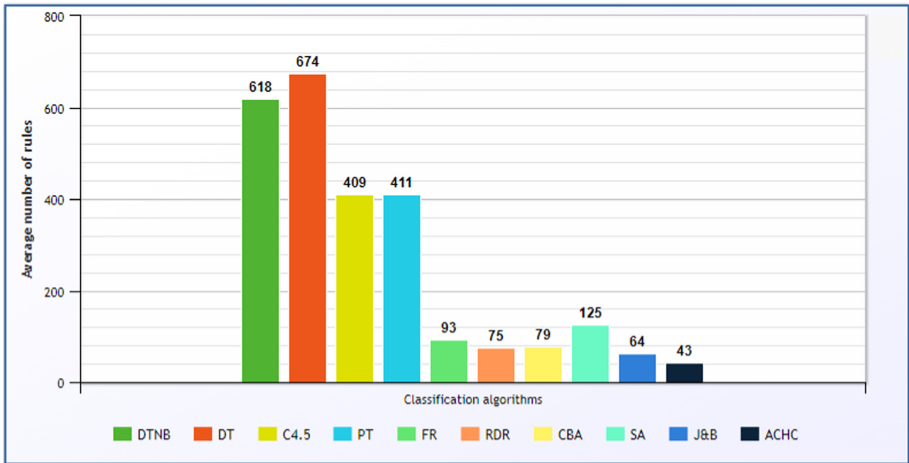


Fig. 2. Comparison between our method and other methods on average number of rules

Table.5 Statistically significant wins/loss counts of ACHC method on rules

	DTNB	DT	C4.5	PT	FR	RDR	CBA	SA	J&B
W	10	9	9	8	10	6	7	10	9
L	2	3	2	2	2	4	2	2	2
N	0	0	1	2	0	2	3	0	1

produced an accurate and compact classifier that was able to reduce the number of classification rules in the classifier by 2–4 times on average compared to the other “classical” rule-learners, while this ratio is even bigger on datasets with a higher number of examples.

The main drawback of our proposed method (ACHC) is its time efficiency. In future work we plan to optimize ACHC to bring its time complexity at least a bit closer to state-of-the-art “divide-and-conquer” rule-learning algorithms.

We plan to develop new methods by setting up different values for α parameter and perform experiments to show the advantages and disadvantages of those methods.

Acknowledgement. The authors gratefully acknowledge the European Commission for funding the InnoRenew CoE project (Grant Agreement #739574) under the Horizon2020 Widespread-Teaming programme and the Republic of Slovenia (Investment funding of the Republic of Slovenia and the European Union of the European Regional Development Fund). They also acknowledge the Slovenian Research Agency ARRS for funding the project J2-2504. Jamolbek Mattiev is also funded for his Ph.D. by the “El-Yurt-Umidi” foundation under the Cabinet of Ministers of the Republic of Uzbekistan.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Bocca, J.B., Jarke, M., Zaniolo, C. (eds.) VLDB 1994 Proceedings of the 20th International Conference on Very Large Data Bases, pp. 487–499. Chile (1994)
2. Cohen, W.W.: Fast Effective rule induction. In: Prieditis, A., Russel, S.J. (eds.) ICML 1995 Proceedings of the Twelfth International Conference on Machine Learning, pp. 115–123. California (1995)
3. Dahbi, A., Mouhir, M., Balouki, Y., Gadi, T.: Classification of association rules based on K-means algorithm. In: Mohajir, M.E., Chahhou, M., Achhab, M.A., Mohajir, B.E. (eds.) 4th IEEE International Colloquium on Information Science and Technology, pp. 300–305. Tangier, Morocco (2016)
4. Dechang, P., Xiaolin, Q.: A new fuzzy clustering algorithm on association rules for knowledge management. *Inf. Technol. J.* **7**(1), 119–124 (2008)
5. Deng, H., Runger, G., Tuv, E., Bannister, W.: CBC: an associative classifier with a small number of rules. *Decis. Support Syst.* **50**(1), 163–170 (2014)
6. Dua, D., Graff, C.: UCI Machine Learning Repository. University of California, Irvine, CA (2019)
7. Frank, E., Witten, I.: Generating accurate rule sets without global optimization. In: Shavlik, J.W. (eds) Fifteenth International Conference on Machine Learning, pp. 144–151. USA (1998)
8. Gupta, K.G., Strehl, A., Ghosh, J.: Distance based clustering of association rules. In: Proceedings of Artificial Neural Networks in Engineering Conference, pp. 759–764. USA (1999)
9. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. *SIGKDD Explor.* **11**(1), 10–18 (2009)
10. Hall, M., Frank, E.: Combining Naive Bayes and Decision Tables. In: Wilson, D.L., Chad, H. (eds.) Proceedings of Twenty-First International Florida Artificial Intelligence Research Society Conference, pp. 318–319, Florida, USA (2008)
11. Hühn, J., Hüllermeier, E.: FURIA: an algorithm for unordered fuzzy rule induction. *Data Min. Knowl. Disc.* **19**(1), 293–319 (2019). <https://doi.org/10.1007/s10618-009-0131-8>

12. Hu, L.Y., Hu, Y.H., Tsai, C.F., Wang, J.S., Huang, M.W.: Building an associative classifier with multiple minimum supports. *SpringerPlus* **5**, 528 (2016). <https://doi.org/10.1186/s40064-016-2153-1>
13. Kaufman, L., Rousseeuw, P.J.: *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, USA (1990).
14. Khairan, D.R.: New associative classification method based on rule pruning for classification of datasets. *IEEE Access* **7**, 157783–157795 (2019)
15. Kohavi, R.: The power of decision tables. In: Lavrač, N., Wrobel, S. (eds) 8th European Conference on Machine Learning, pp. 174–189. Crete, Greece (1995)
16. Kusters, W.A., Marchiori, E., Oerlemans, A.A.J.: Mining clusters with association rules. In: Hand, D.J., Kok, J.N., Berthold, M.R. (eds.) *IDA 1999*. LNCS, vol. 1642, pp. 39–50. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48412-4_4
17. Lent, B., Swami, A., Widom, J.: Clustering association rules. In: Gray, A., Larson, P. (eds.) *Proceedings of the Thirteenth International Conference on Data Engineering*, pp. 220–231. England (1997)
18. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: Agrawal, R., Stolorz, P. (eds.) *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, pp. 80–86. New York, USA (1998)
19. Mattiev, J., Kavšek, B.: A compact and understandable associative classifier based on overall coverage. In: *Procedia Computer Science*, vol. 170, pp. 1161–1167. Warsaw, Poland (2020).
20. Mattiev, J., Kavšek, B.: Simple and accurate classification method based on class association rules performs well on well-known datasets. In: Nicosia, G., Pardalos, P., Umeton, R., Giuffrida, G., Sciacca, V. (eds.) *LOD 2019*. LNCS, vol. 11943, pp. 192–204. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-37599-7_17
21. Mattiev, J., Kavšek, B.: CMAC: clustering class association rules to form a compact and meaningful associative classifier. In: Nicosia, G., et al. (eds.) *LOD 2020*. LNCS, vol. 12565, pp. 372–384. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64583-0_34
22. Mattiev, J., Kavšek, B.: Distance-based clustering of class association rules to build a compact, accurate and descriptive classifier. *Comput. Sci. Inf. Syst.* **18**(3), 791–811 (2021). <https://doi.org/10.2298/CSIS200430037M>
23. Mattiev, J., Kavšek, B.: Coverage-based classification using association rule mining. *Appl. Sci.* **10**, 7013 (2020). <https://doi.org/10.3390/app10207013>
24. Ng, T.R., Han, J.: Efficient and effective clustering methods for spatial data mining. In: Bocca, J., B., Jarke, M., Zaniolo, C. (eds.) *Proceedings of the 20th Conference on Very Large Data Bases (VLDB)*, pp. 144–155, Santiago, Chile (1994)
25. Phipps, A., Lawrence, J.H.: *An overview of combinatorial data analysis. clustering and classification*, pp. 5–63, World Scientific, New Jersey (1996)
26. Quinlan, J.: C4.5: programs for machine learning. *Mach. Learn.* **16**(3), 235–240 (1993)
27. Richards, D.: *Ripple down rules: a technique for acquiring knowledge. Decision-making support systems: achievements, trends and challenges for*, pp. 207–226. IGI Global, USA (2002)
28. Theodoridis, S., Koutroumbas, K.: *Hierarchical algorithms*. *Pattern Recogn.* **4**(13), 653–700 (2009)
29. Zait, M., Messatfa, H.: A comparative study of clustering methods. *Futur. Gener. Comput. Syst.* **13**(2–3), 149–159 (1997)
30. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: an efficient data clustering method for very large databases. In: Widom, J. (ed) *Proceedings of the 1996 ACM-SIGMOD International Conference on Management of Data*, pp. 103–114. Montreal, Canada (1996)