

Article

Coverage-Based Classification Using Association Rule Mining [†]

Jamolbek Mattiev ^{1,2}  and Branko Kavsek ^{1,3,*} 

¹ Department of Information Sciences and Technologies, University of Primorska, 6000 Koper, Slovenia; jamolbek.mattiev@famnit.upr.si or jamolbek_1992@mail.ru

² Information Technologies Department, Urgench State University, 220100 Urgench, Uzbekistan

³ AI Laboratory, Jožef Stefan Institute, 1000 Ljubljana, Slovenia

* Correspondence: branko.kavsek@upr.si; Tel.: +386-5-6117-653

[†] This work paper is an extended version of paper published in the international conference: Statistical Methods and Artificial Intelligence (IWSMAI), Warsaw, Poland, 6–9 April 2020.

Received: 15 September 2020; Accepted: 23 September 2020; Published: 9 October 2020



Abstract: Building accurate and compact classifiers in real-world applications is one of the crucial tasks in data mining nowadays. In this paper, we propose a new method that can reduce the number of class association rules produced by classical class association rule classifiers, while maintaining an accurate classification model that is comparable to the ones generated by state-of-the-art classification algorithms. More precisely, we propose a new associative classifier that selects “strong” class association rules based on overall coverage of the learning set. The advantage of the proposed classifier is that it generates significantly smaller rules on bigger datasets compared to traditional classifiers while maintaining the classification accuracy. We also discuss how the overall coverage of such classifiers affects their classification accuracy. Performed experiments measuring classification accuracy, number of classification rules and other relevance measures such as precision, recall and f-measure on 12 real-life datasets from the UCI ML repository (Dua, D.; Graff, C. UCI Machine Learning Repository. Irvine, CA: University of California, 2019) show that our method was comparable to 8 other well-known rule-based classification algorithms. It achieved the second-highest average accuracy (84.9%) and the best result in terms of average number of rules among all classification methods. Although not achieving the best results in terms of classification accuracy, our method proved to be producing compact and understandable classifiers by exhaustively searching the entire example space.

Keywords: frequent itemset; minimum support; minimum confidence; class association rules; associative classification; overall coverage; statistical significance

1. Introduction

Association rule mining [1,2] is one of the important data mining tasks that is nowadays applied to solve different kinds of problems in weak formalization fields. The main goal of association rule mining is to find all rules in datasets that satisfy some basic requirements, such as minimum support and minimum confidence. It was initially proposed by Agrawal [3] to solve the market basket problem in transactional datasets and now it has developed to solve many other problems, such as classification, clustering, etc.

Classification and prediction are two important tasks of data mining that can be used to extract the hidden regularities from the given datasets to form accurate models (classifiers). Classification approaches aim to build classifiers (models) to predict the class label of a future data object. Such analysis can help to provide us with a better understanding of the data comprehensively. Classification methods

have been widely used in many real-world applications, such as customer relationship management [4], medical diagnosis [5] and industrial design [6].

Associative Classification (AC) is a combination of these two important data mining techniques, namely, classification and association rule mining. AC aims to build accurate and efficient classifiers based on association rules. Researchers have proposed several classification algorithms based on association rules called associative classification methods, such as CBA: Classification-based Association [7], CMAR: Classification based on Multiple Association Rules [8], MCAR: Mining Class Association Rules [9], CPAR: Classification based on Predicted Association Rule [10], MMSCBA: Associative classifier with multiple minimum support [11], CBC: Associative classifier with a small number of rules [12], MMAC: Multi-class, multi-label associative classification [13], ARCID: Association Rule-based Classification for Imbalanced Datasets [14], and others.

Experimental evaluations [7,10–15] show that associative classification approaches could achieve higher accuracy than some traditional classification algorithms produced by decision trees [16–21], rule induction [22–26] and non-rule-based approaches [27–30]. Most of the associative classification methods [8,9,31–37] tend to build accurate and compact classifiers that include a limited number of effective rules by using rule selection or pruning techniques. In comparison with some traditional rule-based classification approaches, associative classification has two main characteristics. Firstly, it generates a large number of association classification rules. Secondly, support and confidence thresholds are applied to evaluate the significance of classification association rules. However, associative classification has some drawbacks. Firstly, it often generates a very large number of classification association rules in association rule mining that is computationally expensive, especially when the training dataset is large. It takes great effort to select a set of high-quality classification rules from among them. Secondly, the accuracy of associative classification depends on the setting of the minimum support and the minimum confidence constraints, that is, unbalanced datasets may heavily affect the accuracy of the classifiers. Although associative classification has some drawbacks, it can achieve higher accuracy than the rule- and tree-based classification algorithms on certain real-life datasets.

Our expected contributions in this research work are:

- To build a compact, accurate and understandable associative classifier (model) based on dataset coverage.
- To solve some of the real-life problems by applying our proposed approach (model).

To achieve these goals, we propose a relatively simple and accurate method to produce a compact and understandable classifier by exhaustively searching the entire example space. More precisely, we select the strong class association rules (CARs) according to their contribution to improving the overall coverage of the learning set. Our proposed classifier has an advantage in that it generates a reasonable fewer number of rules on bigger datasets compared to traditional classifiers, because our method has a stopping criterion in the rule-selection process based on the training dataset's coverage; that is, our algorithm is not sensitive to the size of the dataset. Once we build the classifier, we compute the relevance measures such as "Precision", "Recall" and "F-measure", as well as an overall coverage of the classifier to show how overall coverage affects the accuracy of the classifier [38]. We then perform statistical significance testing by using the paired T-test method between our method and other classification methods.

We have performed experiments on 12 datasets from the UCI Machine Learning Database Repository [39] and compared the experimental result with 8 well-known rule-based classification algorithms (Decision Table (DT) [40], Simple Associative Classifier (SA) [41], PART (PT) [24], C4.5 [18], FURIA (FR) [42], Decision Table and Naïve Bayes (DTNB) [43], Ripple Down Rules (RDR) [44], CBA).

The rest of the paper is organized as follows: Section 2 highlights the related work to our research work. The problem statement is given in detail in Section 3. Section 4 presents how to generate class association rules. In Section 5, our proposed algorithm is widely described. Experimental evaluation

(obtained results) is discussed in Section 6. Discussion about this research work is presented in Section 7. Conclusions and future plans are given in Section 8. The Acknowledgement and References close the paper.

2. Related Works

In computer science literature, there are a lot of rule-based, tree-based and probabilistic classification models proposed by prominent researchers. Since we propose associative classification methods based on database coverage, we discuss the performed research works in associative classification fields that were related to our proposed approaches in this section.

CMAR, an associative classification method, also uses multiple association rules for classification [8]. This method first extends FP-growth (an efficient frequent pattern mining) to mine large datasets [45]. Next, CMAR employs a novel data structure to improve the overall accuracy and efficiency, called CR-tree. The main goal of using CR-tree is to store and retrieve a large number of rules compactly and efficiently. In the rule-selection process, CMAR extracts the highly confident, highly related rules based on database coverage and analyzes the correlation between those rules. More precisely, for each rule R , all the examples are found covered by rule R and if R correctly classifies at least one example, then R will be included to a final classifier and cover count (cover count threshold C is applied at the beginning) of those examples covered by R is increased by 1. If the cover count of an example is exceeded by C , then that example is removed. This process continues while both the training dataset and ruleset are not empty.

The key differences are that our approach uses the APRIORI algorithm [3] to generate frequent itemsets, instead of FP-tree and rules being pruned based on minimum support and confidence constraints.

In CBA and MCAR approaches that use the so-called vertical mining approach, the “APRIORI” algorithm is executed first to generate CARs. Then, both algorithms apply greedy pruning based on database coverage to produce predictive rules. They have different rule-selection methods compared to ours—the rule that can classify at least one training example, that is, if the candidate rule’s body and class labels match the training example’s, the body and class labels are selected for the classifier. This step is different in our method since we are trying to increase the overall coverage and accuracy as well as decreasing the size of the classifier: we select the rule as a potential candidate, only if the rule contributes to improving the overall coverage (if the rule’s body matches the training example’s body, we ignore the class label).

The CPAR algorithm proposed by Yin and Han [10] combines the advantages of associative classification and traditional rule-based classification. CPAR utilizes a greedy algorithm to generate rules directly from the training dataset, instead of generating large candidate rules from frequent itemsets as in other associative classification methods. To avoid overfitting, CPAR uses expected accuracy to evaluate each rule and the classification process is also different than traditional classification approaches: First, CPAR selects all rules whose bodies match the testing example, second, it extracts the best k rules for each class among the rules selected in step 1, and finally, CPAR compares the average expected accuracy of the best k rules for each class found in step 2 and predicts the class label which achieves the highest expected accuracy.

The AC method proposed in Reference [46] is based on database coverage associated with test data. During the classification process of the test dataset, this approach considers the class label of rules that are similar to the test data and rule’s position in the classifier. The score of each class label is computed based on the above conditions and the test data is predicted by the class label that has the highest score.

Another AC [37] method aims to build a classifier based on rule-pruning. This method applies the vertical mining (TidList) to generate class association rules. Researchers proposed a new procedure called Active Rule Pruning (APR) to improve predictive accuracy and to reduce rule redundancy. Specifically, in the rule-selecting process, when a rule is being evaluated to be a potential candidate

for a classifier, ranking of the remaining untested rules is also updated constantly after removing the training examples covered by that rule. The most important difference between our method and all other similar AC methods is that our approach has the stopping criteria in the rule-evaluation step based on user-defined coverage (for the learning dataset).

In the associative classification method [47], researchers proposed a new measure, called “condenseness” to evaluate the infrequent rule items. Authors argued that infrequent rule items which are not satisfied with the minimum support threshold could produce strong association rules for classification. “Condenseness” of infrequent rule items is the average of *lift* of all association rules that can be generated from those rule items. A rule item with a high condenseness means that its items are closely related and can produce potential rules for AC, although it does not have enough support. A new associative classifier, CARC—Condensed Association Rules for Classification, is presented based on the concept of “condenseness”. Association rules are generated by using the modified “APRIORI” algorithm and develop the new strategy of rule inference with the condenseness.

A recently proposed associative classifier called ARCID (Association Rule-based Classification for Imbalanced Datasets) aims to extract hidden regularities and knowledge from imbalanced datasets by emphasizing information extracted from minor classes. The key difference between our method and ARCID is in the rule-selection process. ARCID selects the best k rules by using the following interestingness measures: first, rules that have higher accuracy (rules that have higher supports can have higher predictive accuracy) and belong to major classes; second, rare rules that have lower support and belong to minor classes, while our algorithm selects the best rules according to their overall coverage (training dataset coverage) and the number of rules generated by our model depends on the *intended_overall_coverage* threshold.

Another recent approach [48] aims to produce an optimized associative classifier based on non-trivial data insertion for the incremental dataset. To solve the unattended data insertion and optimization problem, a new technique called “C-NTDI” is used to build a classifier, when there is an insertion of data that takes place in a non-trivial fashion in the initial data that are used for updating the classification rules. In the rule-generation part, PPCE and BAT [49] techniques are applied: PPCE is used to generate the rules, and further Proportion of Frequency Occurrence check with BAT Algorithm (PFOCBA) is applied for pruning the rules that are generated.

The FURIA (Fuzzy Unordered Rule Induction Algorithm) is a rule-based classification method [42] which is a modified and extended version of the RIPPER algorithm [23]. FURIA learns fuzzy rules instead of conventional rules and unordered rulesets (namely a set of rules for each class in a one-vs-rest scheme) instead of rule lists. Moreover, to deal with uncovered examples, it makes use of an efficient rule stretching method. The idea is to generalize the existing rules until they cover the example.

3. Problem Definition and Goals

Associative classification and association rule discovery are two important tasks of association rule mining. There are some differences between them. First, associative classification is supervised learning where the class value is known, while association rule discovery is unsupervised learning. The second and main difference between association rule discovery and AC is that the rule’s consequence in AC can have only the class attribute, however, multiple attribute values are allowed in the rule’s consequence to discover association rules. Overfitting prevention is an important task in AC, but it is really not essential in association rule discovery as in AC. Overfitting problems usually occur with imbalanced datasets, noise datasets or very small learning datasets. We can define the building of an accurate and compact classifier by using AC in the following steps.

Figure 1 shows the general steps used in the AC approach. The first step is computationally expensive because it is similar to the association rule discovery which exhaustively searches the entire example space to discover the frequent itemsets (Step 1). In the second step, we generate strong class association rules from frequent itemsets that satisfy a user-defined minimum confidence threshold. A single rule is represented in AC by implication $X \rightarrow C$, where C is the consequence (class label) and X

is antecedent (Itemset) of the rule, that is, the highest frequency of class associated with its antecedent X in the learning dataset. Once the strong class association rules are generated, building an accurate and compact classifier is a straightforward approach, because no support counting or no scanning of the training dataset is required. The most important task is to select the most effective ordered subset of rules to form an intended classifier. In the last step, we measure the quality of the classifier on some independent test datasets.

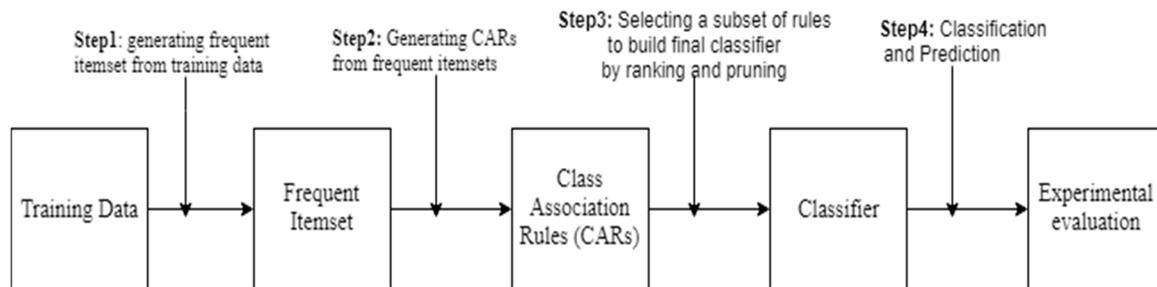


Figure 1. Associative Classification working procedure.

We assume that we have given a normal relational table which consists of N examples described by L distinct attributes and all examples are classified into M known classes. Attribute type can be categorical (nominal) or continuous (numeric). Our proposed research supports only categorical (nominal) attributes, therefore, we treat all the attributes uniformly. We map the categorical attribute's values to consecutive positive integers. Numeric attributes are discretized into intervals (bins), and the intervals are also mapped to consecutive positive integers. We use an equal height (frequency) discretization method to discretize the numeric attributes. Discretization methods are not discussed in this work since enough discretization methods are proposed in machine learning literature. Our goals are stated below:

- Our first goal is to generate the complete set of strong class association rules that satisfy the user-specified minimum support and minimum confidence constraints.
- The second and main goal is to build a simple and accurate classifier by straightforward pruning; more precisely, we select the strong rules that have the contribution to the improvement of overall coverage.
- The third goal is to discuss how overall coverage affects the classification accuracy of our proposed classifier.

4. Discovery of Class Association Rules

The step of finding frequent itemsets in associative classification is the most important and computationally expensive step [10,11]. Several different approaches to discover frequent rule items from a dataset have been adopted from association rule discovery. For example, some AC methods [15,50] employ the "APRIORI" candidate generation method. Other AC methods [8,51,52] mainly apply the FP-growth approach [45] to generate association rules. The CPAR algorithm utilizes a greedy strategy produced in FOIL [53]. "Tid-lists" methods of vertical data layout [54,55] are used in References [9,13]. This section focuses on discovering class association rules from frequent itemsets.

Association rule generation is usually split up into two main steps:

1. In the first step, the minimum support threshold is applied to find all frequent itemsets from the training dataset.
2. A minimum confidence constraint is applied to generate strong class association rules from the frequent itemsets.

In the first step, we use the “APRIORI” algorithm to find frequent itemsets. “APRIORI” utilizes the ‘downward-closure’ property to speed up the searching process by reducing the number of candidate itemsets at any level. The main feature of ‘downward-closure’ is that all subsets of a frequent itemset must be frequent. Infrequent itemsets found at any level are removed, because, it is not possible to “make” a frequent itemset from an infrequent one. “APRIORI” uses this process to prune infrequent itemsets before computing their support at any level. This reduces the time complexity of computing the support for all combinations of items in a dataset.

Once all frequent itemsets are found from a learning dataset in the first step, it is a straightforward approach to generate strong class association rules that satisfy both minimum support and minimum confidence constraints from these frequent itemsets in the second step. This can be done using the following equation for confidence:

$$\text{confidence}(X \rightarrow C) = \frac{\text{support_count}(X \cup C)}{\text{support_count}(X)} \quad (1)$$

The Equation (1) is expressed in terms of itemsets support count, where X is antecedent (itemsets that is, left-hand side of the rule), C is a consequence (a class label that is, right-hand side of the rule), $\text{support_count}(X \cup C)$ is the number of transactions containing the itemsets $X \cup C$, and $\text{support_count}(X)$ is the number of transactions containing the itemsets X . Based on this equation, CARs can be generated as follows:

- For each frequent itemsets L and class label C , generate all nonempty subsets of L .
- For every nonempty subset S of L , output the strong rule R in the form of “ $S \rightarrow C$ ” if, $\frac{\text{support_count}(R)}{\text{support_count}(S)} \geq \text{min_conf}$, where min_conf is the minimum confidence threshold.

5. Our Associative Classification Approach—J&B

Our proposed approach—J&B, can be defined in three steps:

1. A complete set of strong class association rules is generated from the training dataset discussed in Section 4.
2. We then select strong class association rules that highly contribute to increasing the coverage of the classifier until we meet the stopping criterion to build an accurate, compact and understandable classifier.
3. In the last step, we find the overall coverage and accuracy of the intended classifier to show how the overall coverage of the classifier affects its classification accuracy.

5.1. Our Intended Classifier

Once we generated the strong class association rules in Section 4, we propose our classifier by using them. Our proposed method is outlined in Algorithm 1.

Lines 1–2 find all frequent itemsets in the training dataset by using the “APRIORI” algorithm described in Section 4. The third line generates strong class association rules that satisfy the minimum support and confidence constraints from frequent itemsets. Class association rules (generated in line 3) are sorted by confidence and support descending order in the fourth line by the following criteria: Given two rules R_1 and R_2 , R_1 is said to have a higher rank than R_2 , denoted as $R_1 > R_2$,

- If and only if, $\text{conf}(R_1) > \text{conf}(R_2)$; or
- If $\text{conf}(R_1) = \text{conf}(R_2)$ but, $\text{supp}(R_1) > \text{supp}(R_2)$; or
- If $\text{conf}(R_1) = \text{conf}(R_2)$ and $\text{supp}(R_1) = \text{supp}(R_2)$, R_1 has fewer attribute values in its left-hand side than R_2 does;
- If all the parameters of the rules are equal, we can choose any of them.

Initial values of *classified_traindata* array are filled with *false* value in line 5. This array is used later to compute the overall coverage of the training data. Basically, the classifier is built in lines 6–17. We scan the training dataset for each rule (lines 6–7): if the rule classifies a new unclassified example 5 (that is, the rule’s body matches the new example’s body), then we increase the contribution (that is, the contribution of the rule to increase the overall coverage) in lines 8–11. If *contribution* is higher than 0, that is, the rule classified new example(s), then we increase the overall coverage by *contribution*, and we add that rule to our final classifier in lines 12–14. If the rule can classify training examples, but cannot contribute to the betterment of the overall coverage, then we do not evaluate that rule as a potential rule for our classifier. Lines 15–16 stop the procedure if the classifier achieves the intended overall coverage that is higher than the user-defined *intended_coverage* threshold, and line 17 returns the proposed classifier.

Algorithm 1: Building a compact and accurate classifier—J&B

Input: A training dataset *D*, minimum support and confidence constraints, intended coverage

Output: A subset of strong class association rules for classification

```

1: D = taining_dataset;
2: F = frequent_itemsets(D);
3: Rule = genCARs(F);
4: Rule = sort(R, minconf, minsup);
5: Fill(classified_traindata, false);
6: for i: = 1 to Rule.length do begin
7:   for j: = 1 to D.length do begin
8:     if classified_traindata[j] = false then
9:       if Rule[i].premise classifies D[j].premise then
10:        classified_traindata[j] = true;
11:        contribution = contribution + 1;
12:   if contribution > 0 then
13:     overall_coverage = overall_coverage + contribution;
14:     Classifier = Classifier.add(rule[i]);
15:   if (overall_coverage/D.length) ≥ intended_coverage then
16:     break;
17: return Classifier

```

The classification process of our method is shown in Algorithm 2.

Algorithm 2: Classification process of J&B

Input: A Classifier and test dataset

Output: Predicted class

```

1: for each rule y ∈ Classifier do begin
2:   if y classify test_example then
3:     class_count[y.class]++;
4:   if max(class_count) = 0 then
5:     predicted_class = majority_classifier;
6:   else
7:     predicted_class = max_index(class_count);
8: return predicted_class

```

Algorithm 2 depicts the procedure of using the J&B classifier to find the class label of a (new) test example. For each rule in the J&B classifier (line 1): if the rule can classify the example correctly, then we increase the corresponding class count by one and store it (lines 2–3). If none of the rules can classify the new example correctly, then the algorithm returns the majority class value (lines 4–5); otherwise, it returns the majority class value of correctly classified rules (lines 6–7).

5.2. Overall Coverage and Accuracy

After our classifier is built (as in Section 5.1), it is straightforward to compute the overall coverage and accuracy of the classifier (defined in Algorithm 3). To compute the overall coverage, we count the examples that are covered by the classifier and divide it by the total number of examples in the dataset. For accuracy, we count all the examples that are classified by the classifier and divide it by the total number of examples in the dataset.

The first line finds the length of the dataset. In the second and third lines, we fill all initial values of *classified_testdata* (to compute the overall coverage) and *classified_testdata_withclass* (to compute the overall accuracy) arrays as *false*. Lines 4–11 generally switch the status of the test example, without a class label to compute the overall coverage in lines 6–8 and with a class label to calculate the accuracy in lines 9–11. The number of correctly classified examples by classifier without class labels and with class labels are counted in lines 12–16 to calculate the overall coverage of the classifier, and overall accuracy is computed in lines 17–18. The last line returns the obtained results.

Algorithm 3: Overall coverage and accuracy of the classifier

Input: Dataset *D* and classifier *C*

Output: Overall coverage and accuracy of the classifier

```

1: n = D.length();
2: Fill (classified_testdata, false);
3: Fill (classified_testdata_withclass, false);
4: for i: = 1 to C.length do
5:   for j: = 1 to n do
6:     if C[i].premise classifies D[j].premise then
7:       if classified_testdata[j] = false then
8:         classified_testdata[j] = true;
9:       if C[i] classifies D[j] then
10:        if classified_testdata_withclass[j] = false then
11:          classified_testdata_withclass[j] = true;
12: for i: = 1 to n do
13:   if classified_testdata[i] then
14:     testcoverage++;
15:   if classified_testdata_withclass[i] then
16:     accuracy++;
17: Overallcoverage_testdata = testcoverage/n;
18: Overallaccuracy_testdata = accuracy/n;
19: return Overallaccuracy_testdata, Overallcoverage_testdata

```

Example 1. Let us assume we have given the following training dataset that is illustrated in Table 1 and we apply here minimum coverage threshold as 80%, that is, when our intended classifier covers at least 80% of the training examples, then, we stop.

We have the following class association rules (shown in Table 2) (which satisfied the user-specified minimum support and confidence thresholds) that are generated based on minimum support and confidence thresholds which are discussed in the fourth section.

In the second step, we sort the class association rules by confidence and support descending order. The result is shown in Table 3.

Table 1. Training dataset.

#	a1	a2	a3	a4	a5	Class
1	3	4	2	1	3	1
2	5	3	1	4	2	2
3	1	5	2	4	2	1
4	3	2	3	1	2	3
5	?	2	?	3	5	1
6	1	3	5	4	3	2
7	2	2	1	2	2	2
8	1	4	4	2	2	1
9	1	?	2	4	3	2
10	1	5	1	1	5	3
11	1	2	1	1	5	2
12	1	4	4	4	2	1
13	1	3	?	?	?	2
14	1	4	5	4	4	3
15	3	2	2	1	2	10

Table 2. Generated CARs from the training dataset

CARs	Support (%)	Confidence (%)
{a1 = 3, a3 = 2, a4 = 1, a5 = 3} → {1}	4	72
{a1 = 1, a2 = 5} → {3}	5	100
{a1 = 3, a4 = 1} → {1}	8	100
{a1 = 1, a3 = 5, a4 = 4} → {3}	5	100
{a1 = 2, a2 = 2, a3 = 2} → {2}	5	63
{a2 = 3, a3 = 1, a4 = 4, a5 = 2} → {1}	3	80
{a1 = 2, a2 = 3, a3 = 4} → {2}	11	85
{a3 = 1, a5 = 2} → {1}	8	90
{a1 = 1, a2 = 5, a5 = 5} → {3}	6	65
{a1 = 1, a3 = 4, a5 = 2} → {2}	10	85
{a1 = 1, a2 = 2, a3 = 1, a4 = 3, a5 = 1} → {3}	4	64
{a2 = 2, a5 = 5} → {2}	6	100

Table 3. Sorted class association rules.

CARs	Support (%)	Confidence (%)
{a1 = 3, a4 = 1} → {1}	8	100
{a2 = 2, a5 = 5} → {2}	6	100
{a1 = 1, a2 = 5} → {3}	5	100
{a1 = 1, a3 = 5, a4 = 4} → {3}	5	100
{a3 = 1, a5 = 2} → {1}	8	90
{a1 = 2, a2 = 3, a3 = 4} → {2}	11	85
{a1 = 1, a3 = 4, a5 = 2} → {2}	10	85
{a2 = 3, a3 = 1, a4 = 4, a5 = 2} → {1}	3	80
{a1 = 3, a3 = 2, a4 = 1, a5 = 3} → {1}	4	72
{a1 = 1, a2 = 5, a5 = 5} → {3}	6	65
{a1 = 1, a2 = 2, a3 = 1, a4 = 3, a5 = 1} → {3}	4	64
{a1 = 2, a2 = 2, a3 = 2} → {2}	5	63

In the next step, we form our classifier by selecting the strong rules. We select strong rules which contribute to improving the overall coverage, and we continue until achieving the intended training dataset coverage. Table 4 illustrates our final classifier.

Table 4. Final classifier.

N	Classifier Rules	# Classified Examples
1	{a1 = 3, a4 = 1} → {1}	3
2	{a2 = 2, a5 = 5} → {2}	1
3	{a1 = 1, a2 = 5} → {3}	2
4	{a1 = 1, a3 = 5, a4 = 4} → {3}	2
5	{a3 = 1, a5 = 2} → {1}	2
6	{a1 = 1, a3 = 4, a5 = 2} → {2}	2

Our classifier includes 6 rules. In this example, intended coverage is equal to 80% and 6 classification rules in the classifier cover 80% of the learning set. Since our training dataset has some examples with missing values, our classifier covered the whole training dataset (examples without missing values). Other rules also may cover unclassified examples, but we cannot exceed the user-defined training dataset coverage threshold. This is our stopping criterion and we cannot include other rules into our classifier. We cannot also include classification rules which cover only classified examples (this means it does not contribute to the improvement of overall coverage). Now, we classify the following unseen example:

$$\{a1 = 1, a2 = 5, a3 = 5, a4 = 4, a5 = 5\} \rightarrow ?$$

So, this example is classified by the third and fourth classification rules. The class values of the rules which correctly classified the new example are 3 and 3. So, our classifier predicts that the class value of the new example is 3 (because majority class value is 3).

6. Experimental Evaluation

We tested our model on 12 real-life datasets taken from the UCI Machine Learning Repository. We evaluated our classifier (J&B) by comparing it with 8 well-known classification methods on the accuracy, relevance measures and the number of classification rules. All differences were tested for statistical significance by performing a paired t-test (with a 95% significance threshold).

J&B was run with default parameters *minimum support* = 1% and *minimum confidence* = 60% (on some datasets, however, *minimum support* was lowered to 0.5% or even 0.1% to ensure enough CARs to be generated for each class value). For all other 8 rule learners, we used their WEKA workbench [56] implementation with default parameters. We applied 90% as a required coverage (training dataset) threshold that is the stopping point of selecting rules to our classifier described in Section 5.1. The description of the datasets and input parameters are shown in Table 5.

Table 5. Description of datasets and AC algorithm parameters.

Dataset	# of Attributes	# of Classes	# of Records	Min Support	Min Confidence
Breast Can	10	2	286	1%	60%
Balance	5	3	625	1%	50%
Car.Evn	7	4	1728	1%	50%
Vote	17	2	435	1%	60%
Tic-Tac-Toe	10	2	958	1%	60%
Nursery	9	5	12,960	0.5%	50%
Hayes-root	6	3	160	0.1%	50%
Lymp	19	4	148	1%	60%
Spect.H	23	2	267	0.5%	50%
Adult	15	2	45,221	0.5%	60%
Chess	37	2	3196	0.5%	60%
Connect4	43	3	67,557	1%	60%

Furthermore, all experimental results were produced by using a 10-fold cross-validation evaluation protocol.

Experimental results on classification accuracies (average values over the 10-fold cross-validation with standard deviations) are shown in Table 6 (OC: Overall Coverage).

Table 6. The comparison between our method and some classification algorithms on accuracy.

Dataset	DTNB	DT	C4.5	PT	FR	RDR	CBA	SA	J&B	OC
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	88.7
Balance	81.4 ± 8.1	66.7 ± 5.0	64.4 ± 4.3	76.2 ± 5.6	77.5 ± 7.6	68.5 ± 4.3	73.2 ± 3.8	74.0 ± 4.1	74.1 ± 2.6	86.3
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	94.2
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	92.8
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	100.0
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	85.6
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
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	90.1
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	94.8
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	91.7
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	100.0
Connect4	78.8 ± 5.9	76.7 ± 7.7	80.0 ± 6.8	81.1 ± 7.9	80.6 ± 7.1	80.0 ± 6.4	80.9 ± 8.1	78.7 ± 6.0	81.2 ± 5.2	90.6
Avg (%):	81.5 ± 4.4	79.3 ± 4.5	83.6 ± 5.2	85.6 ± 4.9	84.5 ± 4.4	83.6 ± 4.1	84.5 ± 4.9	82.3 ± 3.4	84.9 ± 3.3	91.3

Our observations (Table 6) on selected datasets show that our proposed classifier (J&B) achieved better performance than Decision Table (DT), C4.5, Decision Table and Naïve Bayes (DTNB), Ripple Down Rules (RDR) and Simple Associative Classifier (SA) (84.9% and 79.3%, 83.6%, 81.5%, 83.6%, 82.3%) on average accuracy. Our proposed method achieved the best accuracy on “Breast Cancer”, “Hayes.R” and “Connect4” datasets.

Standard deviations ranged around 4.0 in all classification methods and were higher for all methods on “Breast cancer”, “Hayes.R”, “Lymp” and “Connect4” datasets (above 4); that is, the differences between accuracies fluctuated and were reasonably high in 10-fold cross-validation experiments. When the overall coverage is above 90%, our proposed method tends to get reasonably high accuracy on all datasets. On the other hand, overall coverage of “J&B” was lower than its accuracy on “Vote” and “Nursery” datasets. This fact is not surprising since uncovered examples get classified by the majority classifier.

Statistically significant testing (wins/losses counts) on accuracy between J&B and other classification models is shown in Table 7. W: our approach was significantly better than compared algorithms, L: selected rule-learning algorithm significantly outperformed our algorithm, W-L: no significant difference has been detected in the comparison.

Table 7 illustrates that the performance of the J&B method on accuracy was better than DTNB, DT, RDR and SA methods. Although J&B obtained a similar result with FR and CBA (there is no statistical difference on 8 datasets out of 12), it is beaten by the PT algorithm according to win/losses counts. However, on average, the classification accuracies of J&B are not much different from those of the other 8 rule-learners.

Table 7. Statistically significant wins/losses counts of J&B method on accuracy.

	DTNB	DT	C4.5	PT	FR	RDR	CBA	SA
W	6	6	4	2	2	4	2	6
L	3	2	3	4	2	1	2	1
W-L	3	4	5	6	8	7	8	5

Results show (Table 8) that J&B generates a reasonably smaller number of compact rules which is not sensitive to the size of the training datasets, while the number of rules produced by traditional classification methods such as “DT”, “C4.5”, “PT” and “DTNB” algorithms depend on the dataset’s size. Even though not achieving the best classification accuracies on “Nursery” and “Adult” datasets, it produced the lowest number of rules on those datasets among all classification models.

Table 8. The number of classification rules generated by the classifiers.

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

Statistically significant counts (wins/losses) of J&B against other rule-based classification models on classification rules is shown in Table 9.

Table 9. Statistically significant wins/losses counts of J&B method on rules.

	DTNB	DT	C4.5	PT	FR	RDR	CBA	SA
W	10	7	6	6	8	5	7	10
L	2	5	4	6	4	5	3	2
W-L	0	0	2	0	0	2	2	0

Table 9 proves that J&B produced a statistically smaller classifier than DTNB and SA methods on 10 datasets, and DT, FR and CBA methods on 7 (or more than 7) datasets out of 12. The most importantly, J&B generated statistically smaller classifiers than all other models on bigger datasets, which was our main goal in this research. Experimental evaluations on bigger datasets (over 10,000 samples) are shown in Figure 2.

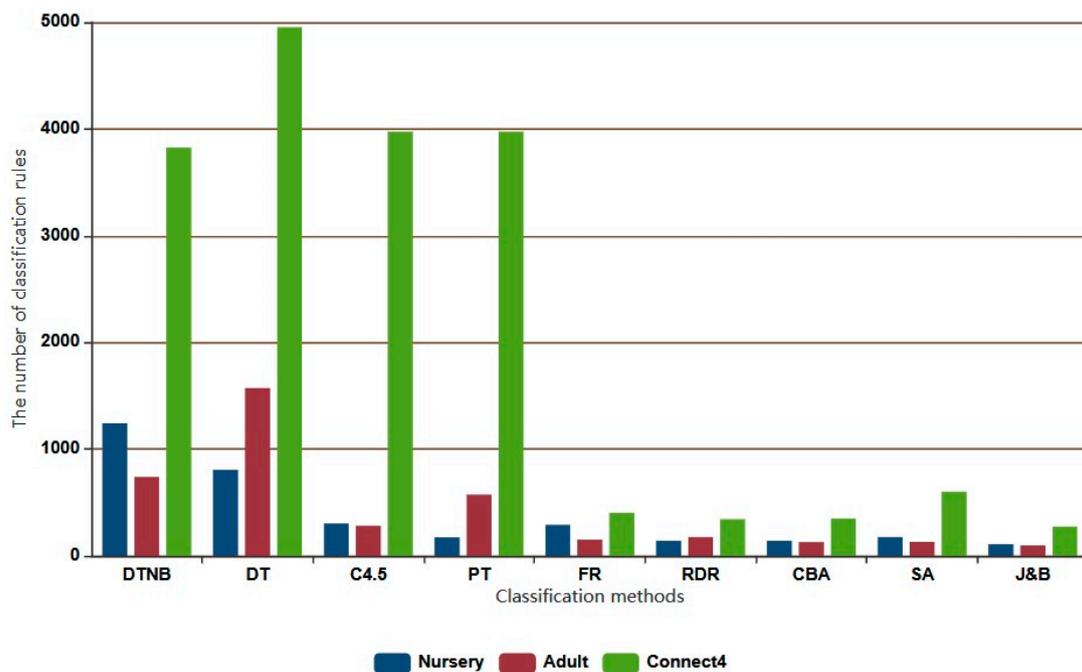


Figure 2. Comparison of rule-based classification methods on the average number of rules.

Figure 2 proves the advantage of the J&B method: it produced the smallest classifier among all rule-based classification models on selected datasets.

Our experiment on relevance measures (average results) such as “Precision”, “Recall” and “F-measure”, is highlighted in Figure 3. The detailed result for each dataset can be found in Appendix A.

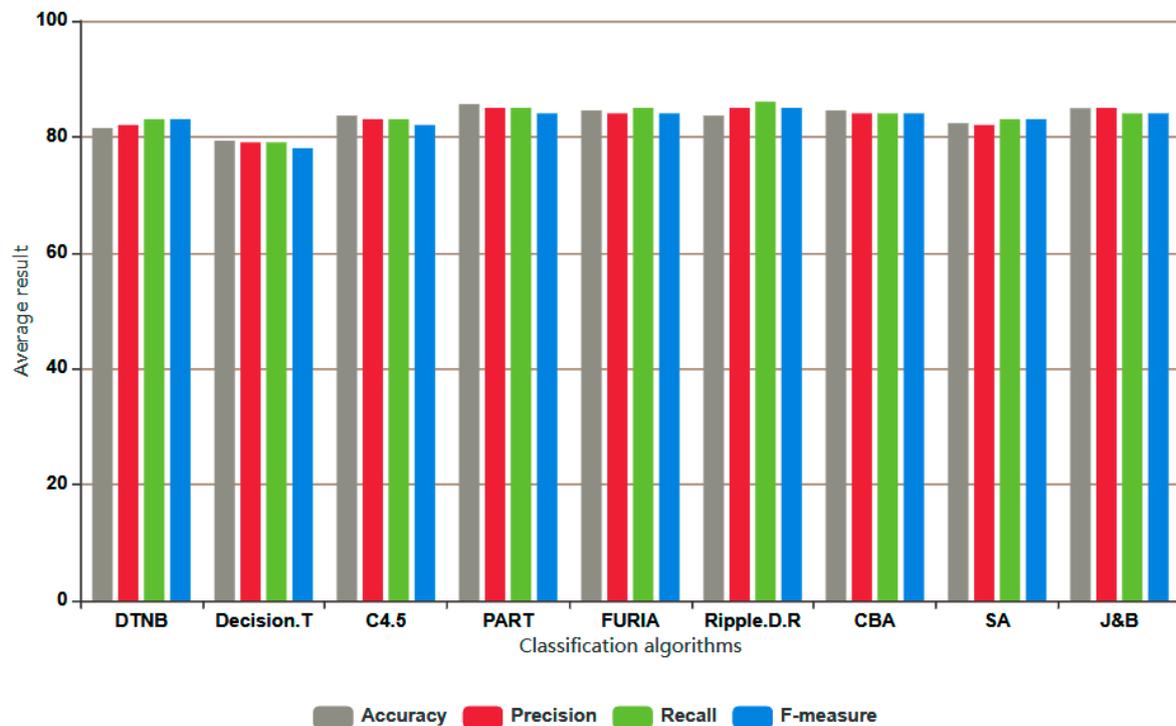


Figure 3. Comparison of the J&B classifier on Accuracy, Precision, Recall and F-measure.

7. Discussion

J&B produces classifiers that have on average far fewer rules than those produced by the “classical” rule-learning methods and slightly smaller number of rules comparing to associative classification models included in the comparison. J&B generated a reasonably smaller number of rules on bigger datasets compared to all other classification methods. J&B achieved the best accuracy on “Breast cancer”, “Hayes-root” and “Connect4” datasets among all methods (80.5%, 79.3% and 81.2% respectively), while it got a slightly worse result on “Nursery” and “Car.Evn” datasets. On other datasets, J&B achieved almost similar results with all rule-based classification methods.

We used the “WEKA” software to generate class association rules and to explore other classification algorithms because “WEKA” is open-source software that includes all the necessary classification approaches and it is relatively easy to use and program. Our proposed algorithm has some limitations:

- To get “enough” class association rules (“enough” means at least 5–10 rules for each class value—this situation mainly happens with imbalanced datasets) for each class value and achieve a reasonable overall coverage, we need to apply appropriate minimum support and minimum confidence thresholds. That is, we should take into consideration the class distribution of each dataset.
- In this research work, one of the novel approaches is to apply the intended coverage constraint which becomes the stopping criteria of selecting rules for our classifier. Here, we aimed to show that it is possible to achieve reasonably high accuracy without covering the whole dataset (especially when the dataset has some missing or noisy examples). This criterion becomes an advantage of our method that produces significantly smaller classification rules on bigger datasets.
- The J&B produced an accurate and compact classifier on a dataset which has a large number of attributes and records (especially, with balanced datasets on class-distribution), while the

performance of J&B will be slightly worse with the datasets having a small number of attributes and a large number of records (as proven on “Nursery” and “Car.Evn” datasets).

- It is unnecessary to remove the missing values from the dataset because all “classical” classification algorithms and our method can handle missing values directly.
- J&B generates a smaller number of classification rules compared to all other traditional and associative classifiers on bigger datasets.
- We tried to solve the outer-class overlapping problem (that means some samples from different classes have very similar characteristics) by grouping the CARs according to their class value, and the inter-class overlapping problem (several rules that belong to the same class may cover the same samples) by selecting the rules based on database coverage.
- Our method may get slightly worse results in terms of classification accuracy than other traditional classification methods on imbalanced datasets, because unbalanced class distribution may affect the rule-generation part of our algorithm.
- The time complexity of the APRIORI algorithm is potentially exponential but, on the other hand, we gain in more exhaustively exploring the entire search space and in part, this issue can be solved by trying to use more efficient data representation and parallelization of the algorithm, which is what we plan to do in future work.

8. Conclusions and Future Work

Our experiments on accuracy and number of rules show that our method is compact, accurate and comparable with 8 other well-known classification methods. Although it did not achieve the best average classification accuracy, it produced significantly smaller rules on bigger datasets compared to other classification algorithms. Our proposed classifier achieved reasonably high average coverage with 91.3%.

Statistical significance testing shows that our method was statistically better than or equal to other classification methods on some datasets, while it obtained worse results than those methods on some other datasets. The most important achievement in this research was that J&B got significantly better results in terms of an average number of classification rules than all other classification methods, while it had comparable results to those methods on accuracy.

This research was the first and main step for our future goal, where we plan to cluster class association rules by their similarity and thus further reduce their number and increase the accuracy and understandability of the classifier.

Author Contributions: Conceptualization, B.K.; Software, J.M.; Writing—original draft preparation, J.M.; Writing—review and editing, J.M. and B.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors gratefully acknowledge the European Commission for funding the InnoRenewCoE project (Grant Agreement #739574) under the Horizon2020 Widespread-Teaming program and the Republic of Slovenia (Investment funding of the Republic of Slovenia and the European Union of the European Regional Development Fund). 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.

Conflicts of Interest: The authors declare no conflict of interest

Appendix A

We have performed the experiments on relevance measures such as Precision (P), Recall (R) and F-measure (F) of the classifiers by constructing the confusion matrix. Weighted average results are reported in Table A1.

It can be seen from the table that our proposed approach (85, 84, 84) had almost similar average results on Precision, Recall and F-measure with C4.5, PART, DTNB, FURIA, RDR, SA and CBA on all selected datasets (the same as in average classification accuracy).

Table A1. Weighted relevance measures.

Dataset	Relevance Measures (%): Precision (P), Recall (R), F-measure (F)																										
	DT			C4.5			PART			DTNB			FR			RDR			CBA			SA			J&B		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Breast.C	65	69	65	73	75	74	71	74	71	69	70	70	72	75	74	71	74	71	71	71	78	81	79	84	79	77	
Balance	66	66	66	64	64	64	79	77	74	81	80	80	77	78	77	68	69	68	72	73	72	75	74	73	74	74	
Car.Evn	91	91	91	92	92	92	95	95	95	95	95	95	93	92	92	91	91	91	90	91	90	86	87	86	89	89	
Vote	95	94	94	95	95	95	95	95	95	94	95	94	95	94	94	96	95	95	94	94	94	94	94	94	94	94	
Tic-Tac	73	74	73	85	85	85	94	94	94	69	70	70	94	95	94	93	95	94	100	100	100	91	92	91	95	96	
Nursery	93	93	93	94	95	95	96	96	96	96	95	95	91	91	91	90	93	91	92	92	92	91	91	91	89	89	
Hayes	58	50	51	80	78	78	75	73	73	75	76	75	76	78	77	74	75	74	76	75	75	73	74	73	79		
Lymph	70	72	68	77	76	76	81	81	81	74	72	73	80	81	80	76	79	78	79	79	79	74	75	74	80		
Spect.H	79	79	79	79	80	79	79	80	80	79	80	79	78	79	79	80	80	80	79	79	79	79	79	79	79	79	
Adult	82	82	82	82	82	82	82	82	82	82	82	82	75	75	75	81	80	80	81	82	81	80	80	80	80	80	
Chess	97	97	97	98	98	98	98	98	98	93	94	94	95	96	95	93	97	95	95	95	95	92	92	92	94	95	
Connect	76	76	76	80	80	80	80	81	80	78	79	78	80	81	80	80	80	80	81	81	80	78	78	78	80	81	
Avg (%)	79	79	78	83	83	82	85	85	84	82	83	83	84	85	84	85	86	85	84	84	84	82	83	83	85	84	

References

1. Das, A.; Ahmed, M.M.; Ghasemzadeh, A. Using trajectory-level SHRP2 naturalistic driving data for investigating driver lane-keeping ability in fog: An association rules mining approach. *Accid. Anal. Prev.* **2019**, *129*, 250–262. [[CrossRef](#)] [[PubMed](#)]
2. Jeong, H.; Gan, G.; Valdez, E.A. Association rules for understanding policyholder lapses. *Risks* **2018**, *6*, 69. [[CrossRef](#)]
3. Agrawal, R.; Srikant, R. Fast algorithms for mining association rules. In Proceedings of the 20th International Conference on Very Large Data Bases, VLDB '94, Santiago de Chile, Chile, 12–15 September 1994; Bocca, J.B., Jarke, M., Zaniolo, C., Eds.; Morgan Kaufmann: San Francisco, CA, USA, 1994; pp. 487–499.
4. Eric, N.; Xiu, L.; Chau, D. Application of data mining techniques in customer relationship management. A literature review and classification. *Expert Syst. Appl.* **2009**, *36*, 2592–2602.
5. Jyoti, S.; Ujma, A.; Dipesh, S.; Sunita, S. Predictive data mining for medical diagnosis: An overview of heart disease prediction. *Int. J. Comput. Appl.* **2011**, *17*, 43–48.
6. Yoon, Y.; Lee, G. Two scalable algorithms for associative text classification. *Inf. Process Manag.* **2013**, *49*, 484–496. [[CrossRef](#)]
7. Liu, B.; Hsu, W.; Ma, Y. Integrating classification and association rule mining. In Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 27–31 August 1988; Agrawal, R., Stolorz, P., Eds.; AAAI Press: New York, NY, USA, 1998; pp. 80–86.
8. Li, W.; Han, J.; Pei, J. CMAR: Accurate and efficient classification based on multiple class-association rules. In Proceedings of the 1st IEEE International Conference on Data Mining (ICDM'01), San Jose, CA, USA, 29 November–2 December 2001; Cercone, N., Lin, T.Y., Wu, X., Eds.; IEEE Computer Society: Washington, DC, USA, 2001; pp. 369–376.
9. Thabtah, F.A.; Cowling, P.; Peng, Y. MCAR: Multi-class classification based on association rule. In Proceedings of the 3rd ACS/IEEE International Conference on Computer Systems and Applications, Cairo, Egypt, 3–6 January 2005; Institute of Electronical and Electronics Engineers, 2005; pp. 127–133.
10. Yin, X.; Han, J. CPAR: Classification based on Predictive Association Rules. In Proceedings of the SIAM International Conference on Data Mining, San Francisco, CA, USA, 1–3 May 2003; Barbara, D., Kamath, C., Eds.; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA; pp. 331–335.
11. Hu, L.Y.; Hu, Y.H.; Tsai, C.F.; Wang, J.S.; Huang, M.W. Building an associative classifier with multiple minimum supports. *SpringerPlus* **2016**, *5*, 528. [[CrossRef](#)]
12. Deng, H.; Runger, G.; Tuv, E.; Bannister, W. CBC: An associative classifier with a small number of rules. *Decis. Support Syst.* **2014**, *50*, 163–170. [[CrossRef](#)]
13. Thabtah, F.A.; Cowling, P.; Peng, Y. MMAC: A new multi-class, multi-label associative classification approach. In Proceedings of the Fourth IEEE International Conference on Data Mining, Brighton, UK, 1–4 November 2004; IEEE Computer Society: Washington, DC, USA; pp. 217–224.

14. Abdellatif, S.; Ben Hassine, M.A.; Ben Yahia, S.; Bouzeghoub, A. ARCID: A New Approach to Deal with Imbalanced Datasets Classification. In Proceedings of the SOFSEM 2018: 44th International Conference on Current Trends in Theory and Practice of Computer Science, Krems, Austria, 29 January–2 February 2018; Tjoa, A., Bellatreche, L., Biffl, S., van Leeuwen, J., Wiedermann, J., Eds.; Lecture Notes in Computer Science. Volume 10706, pp. 569–580.
15. Chen, G.; Liu, H.; Yu, L.; Wei, Q.; Zhang, X. A new approach to classification based on association rule mining. *Decis. Support Syst.* **2006**, *42*, 674–689. [[CrossRef](#)]
16. Bayardo, R.J. Brute-force mining of high-confidence classification rules. In Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, Newport Beach, CA, USA, 14–17 August 1997; Heckerman, D., Mannila, H., Eds.; AIII Press: Newport Beach, CA, USA, 1997; pp. 123–126.
17. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
18. Quinlan, J. C4.5: Programs for Machine Learning. *Mach. Learn.* **1993**, *16*, 235–240.
19. Shafer, J.C.; Agrawal, R.; Mehta, M. SPRINT: A scalable parallel classifier for data mining. In Proceedings of the 22nd International Conference in Very Large databases, Mumbai, India, 3–6 September 1996; Vijayaraman, T.M., Buchmann, A.P., Mohan, C., Sarda, N.L., Eds.; Morgan Kaufmann: San Francisco, CA, USA, 1996; pp. 544–555.
20. Friedman, N.; Getoor, L.; Koller, D.; Pfeffer, A. Learning probabilistic relational models. In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99), Stockholm, Sweden, 31 July–6 August 1999; Dean, T., Ed.; Morgan Kaufmann: San Francisco, CA, USA, 1999; pp. 307–335.
21. Ahmed, A.M.; Rizaner, A.; Ulusoy, A.H. A novel decision tree classification based on post-pruning with Bayes minimum risk. *PLoS ONE* **2018**, *13*, e0194168. [[CrossRef](#)]
22. Clark, P.; Niblett, T. The CN2 induction algorithm. *Mach. Learn.* **1989**, *3*, 261–283. [[CrossRef](#)]
23. Cohen, W.W. Fast Effective Rule Induction. In Proceedings of the Twelfth International Conference on Machine Learning, ICML'95, Tahoe City, CA, USA, 9–12 July 1995; Frieditis, A., Russel, S.J., Eds.; Morgan Kaufmann: San Francisco, CA, USA, 1995; pp. 115–123.
24. Frank, E.; Witten, I. Generating Accurate Rule Sets Without Global Optimization. In Proceedings of the Fifteenth International Conference on Machine Learning, Madison, WI, USA, 24–27 July 1998; Shavlik, J.W., Ed.; Morgan Kaufmann: San Francisco, CA, USA, 1998; pp. 144–151.
25. Qabajeh, I.; Thabtah, F.; Chiclana, F.A. Dynamic rule-induction method for classification in data mining. *J. Manag. Anal.* **2015**, *2*, 233–253. [[CrossRef](#)]
26. Pham, D.; Bigot, S.; Dimov, S. RULES-5: A rule induction algorithm for classification problems involving continuous attributes. *Proc. Inst. Mech. Eng. C J. Mech. Eng. Sci.* **2003**, *217*, 1273–1286. [[CrossRef](#)]
27. Baralis, E.; Cagliero, L.; Garza, P. A novel pattern-based Bayesian classifier. *IEEE Trans. Knowl. Data Eng.* **2013**, *25*, 2780–2795. [[CrossRef](#)]
28. Vapnik, V.N. An overview of statistical learning theory. *IEEE Trans. Neural Netw.* **1999**, *10*, 988–999. [[CrossRef](#)] [[PubMed](#)]
29. Zhou, Z.; Liu, X. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Trans. Knowl. Data Eng.* **2006**, *18*, 63–77. [[CrossRef](#)]
30. Donato, M.; Michelangelo, C.; Annalisa, A. A relational approach to probabilistic classification in a transductive setting. *Eng. Appl. Artif. Intell.* **2009**, *22*, 109–116.
31. Thabtah, F.A.; Cowling, P.I. A greedy classification algorithm based on association rule. *Appl. Soft Comput.* **2007**, *7*, 1102–1111. [[CrossRef](#)]
32. Liu, Y.; Jiang, Y.; Liu, X.; Yang, S.-L. CSMC: A combination strategy for multi-class classification based on multiple association rules. *Knowl. Based Syst.* **2008**, *21*, 786–793. [[CrossRef](#)]
33. Dua, S.; Singh, H.; Thompson, H.W. Associative classification of mammograms using weighted rules. *Expert Syst. Appl.* **2009**, *36*, 9250–9259. [[CrossRef](#)] [[PubMed](#)]
34. Song, J.; Ma, Z.; Xu, Y. DRAC: A direct rule mining approach for associative classification. In Proceedings of the International Conference on Artificial Intelligence and Computational Intelligence (AICI '10), Sanya, China, 23–24 October 2010; Wang, F.L., Deng, H., Gao, Y., Lei, J., Eds.; Springer-Verlag: Berlin, Germany, 2010; pp. 150–154.

35. Lin, M.; Li, T.; Hsueh, S.-C. Improving classification accuracy of associative classifiers by using k-conflict-rule preservation. In Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication (ICUIMC '13), Kota Kinabalu, Malaysia, 17–19 January 2013; Association for Computing Machinery: New York, NY, USA, 2013; pp. 1–6.
36. Wang, W.; Zhou, Z. A review of associative classification approaches. *Trans. IoT Cloud Comput.* **2014**, *2*, 31–42.
37. Khairan, D.R. New Associative Classification Method Based on Rule Pruning for Classification of Datasets. *IEEE Access* **2019**, *7*, 157783–157795.
38. Mattiev, J.; Kavšek, B. How overall coverage of class association rules affect affects the accuracy of the classifier? In Proceedings of the 22th International Multi-conference on Data Mining and Data Warehouse—“SiKDD”, Ljubljana, Slovenia, 11 October 2019; pp. 49–52.
39. Dua, D.; Graff, C. *UCI Machine Learning Repository*; University of California: Irvine, CA, USA, 2019.
40. Kohavi, R. The Power of Decision Tables. In Proceedings of the 8th European Conference on Machine Learning, Crete, Greece, 25–27 April 1995; Lavrač, N., Wrobel, S., Eds.; Springer-Verlag: Berlin, Germany, 1995; pp. 174–189.
41. Mattiev, J.; Kavšek, B. Simple and Accurate Classification Method Based on Class Association Rules Performs Well on Well-Known Datasets. In Proceedings of the Machine Learning, Optimization, and Data Science, LOD 2019, Siena, Italy, 10–13 September 2019; Nicosia, G., Pardalos, P., Umeton, R., Giuffrida, G., Sciacca, V., Eds.; Springer: Siena, Italy, 2019; Volume 11943, pp. 192–204.
42. Hühn, J.; Hüllermeier, E. FURIA: An algorithm for unordered fuzzy rule induction. *Data Min. Knowl. Discov.* **2019**, *19*, 293–319. [[CrossRef](#)]
43. Hall, M.; Frank, E. Combining Naive Bayes and Decision Tables. In Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference, Coconut Grove, FL, USA, 15–17 May 2008; Wilson, D.L., Chad, H., Eds.; AAAI Press: Menlo Park, CA, USA, 2008; pp. 318–319.
44. Richards, D. Ripple down rules: A technique for acquiring knowledge. In *Decision Making Support Systems: Achievements, Trends and Challenges for the New Decade*; Mora, M., Forgionne, G.A., Gupta, J.N.D., Eds.; IGI Global: Hershey, PA, USA, 2002; pp. 207–226.
45. Han, J.; Pei, J.; Yin, Y. Mining frequent patterns without candidate generation. In Proceedings of the 2000 ACM-SIGMID International Conference on Management of Data, Dallas, TX, USA, 16–18 May 2000; Association for Computing Machinery: New York, NY, USA, 16 May 2000; pp. 1–12.
46. Ayyat, S.; Lu, J.; Thabtah, F. Class strength prediction method for associative classification. In Proceedings of the Fourth International Conference on Advances in Information Mining and Management, Paris, France, 20–24 July 2014; Schmidt, A., Ed.; International Academy, Research and Industry Association: Delaware, USA, 2014; pp. 5–10.
47. Wu, C.H.-H.; Wang, J.Y. Associative classification with a new condenseness measure. *J. Chin. Inst. Eng.* **2015**, *38*, 458–468. [[CrossRef](#)]
48. Ramesh, R.; Saravanan, V.; Manikandan, R. An Optimized Associative Classifier for Incremental Data Based on Non-Trivial Data Insertion. *Int. J. Innov. Technol. Explor. Eng.* **2019**, *8*, 4721–4726.
49. Ramesh, R.; Saravanan, V. Proportion frequency occurrence count with bat algorithm (FOCBA) for rule optimization and mining of proportion equivalence Fuzzy Constraint Class Association Rules (PEFCARs). *Period. Eng. Nat. Sci.* **2018**, *6*, 305–325. [[CrossRef](#)]
50. Meretakis, D.; Wüthrich, B. Classification as Mining and Use of Labeled Itemsets. In Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 15–18 August 1999; Fayyad, U., Chaudhuri, S., Madigan, D., Eds.; Association for Computing Machinery: New York, NY, USA, 1999.
51. Baralis, E.; Garza, P. A lazy approach to pruning classification rules. In Proceedings of the IEEE International Conference on Data Mining, Maebashi City, Japan, 9–12 December 2002; IEEE Computer Society: Washington, DC, USA, 2002; pp. 35–42.
52. Baralis, E.; Chiusano, S.; Garza, P. On support thresholds in associative classification. In Proceedings of the ACM Symposium on Applied Computing, Nicosia, Cyprus, 14–17 March 2004; Association for Computing Machinery: New York, NY, USA, 2004; pp. 553–558.
53. Quinlan, J.R.; Cameron-Jones, R.M. FOIL: A midterm report. In Proceedings of the 6th European conference on Machine Learning, Vienna, Austria, 5–7 April 1993; Brazdil, P.B., Ed.; Springer-Verlag: Berlin, Germany, 1993; pp. 1–20.

54. Zaki, M.J.; Parthasarathy, S.; Ogihara, M.; Li, W. New algorithm for fast discovery of association rules. In Proceedings of the Third International conference on Knowledge discovery and Data Mining, Newport Beach, CA, USA, 14–17 August 1997; Heckerman, D., Mannila, H., Eds.; AAAI Press: New York, NY, USA, 1997; pp. 286–289.
55. Zaki, M.; Gouda, K. Fast vertical mining using diffsets. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–27 August 2003; Pedro, D., Ed.; Association for Computing Machinery: New York, NY, USA, 2003; pp. 326–335.
56. Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I.H. The WEKA Data Mining Software: An Update. *SIGKDD Explor.* **2009**, *11*, 10–18. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).