Expert Systems With Applications 79 (2017) 1–7

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Predictability-based collective class association rule mining

Kiburm Song, Kichun Lee*

Department of Industrial Engineering, Hanyang University, Seoul, Korea

ARTICLE INFO

Article history:
Received 5 July 2016
Revised 12 February 2017
Accepted 13 February 2017
Available online 16 February 2017

Keywords:
Associative classification
Class association rules
Rule ranking
Rule pruning
Data mining

ABSTRACT

Associative classification is rule-based involving candidate rules as criteria of classification that provide both highly accurate and easily interpretable results to decision makers. The important phase of associative classification is rule evaluation consisting of rule ranking and pruning in which bad rules are removed to improve performance. Existing association rule mining algorithms relied on frequency-based rule evaluation methods such as support and confidence, failing to provide sound statistical or computational measures for rule evaluation, and often suffer from many redundant rules. In this research we propose predictability-based collective class association rule mining based on cross-validation with a new rule evaluation step. We measure the prediction accuracy of each candidate rule in *inner cross-validation* steps. We split a training dataset into inner training sets and inner test sets and then evaluate candidate rules' predictive performance. From several experiments, we show that the proposed algorithm outperforms some existing algorithms while maintaining a large number of useful rules in the classifier. Furthermore, by applying the proposed algorithm to a real-life healthcare dataset, we demonstrate that it is practical and has potential to reveal important patterns in the dataset.

1. Introduction

The process of classification involves models describing an important relationship between input data and their class labels from given datasets. In other words, classification is the task of identifying a mapping function that assigns a new observation to its predicted class using a training data set with known class information. In general, classification has been used in many application domains such as spam mail filtering, document classification, and image classification. Recently, its application has expanded to sentiment analysis in text mining and disease prediction in healthcare systems. From a methodological viewpoint, various classification methods such as decision tree, naïve Bayes rule, and support vector machine have been proposed and used in the fields of data mining and machine learning. In addition, a rule-based classification (rule-induction) method (Cohen, 1995; Quinlan & Cameron-Jones, 1993) has been frequently used. One advantage of rule-based classification is high-level interpretability while maintaining good performance and managing missing observations. The core procedure of such an approach lies in formation of formal and effective rules from a training data set, which can be regarded as an extension of a decision tree (Quinlan, 1993). Specifically, associative classification, a rule-based classification method, has been the focus of many studies since the introduction of associative rule mining by Liu, Hsu, and Ma (1998).

Associative classification (AC) incorporates classification and association rule mining to model construction, i.e. a classifier, in the task of classification. Briefly, association rule mining identifies an antecedent (or condition) and a consequent (or result) that have a conditional connection, such as "if a condition event takes place, a result event is likely." As procedures in data mining, classification and association rule mining differ in that classification is aimed at the prediction of class labels, whereas association rule mining determines interesting relations between items in a transactional dataset (Thabtah, 2007). A few studies (Chen, Hsu, & Chu, 2012; Chen, Wang, Li, Wu, & Tian, 2014; Liu et al., 1998; Thabtah, Cowling, & Peng, 2004) have indicated that AC is superior to several traditional classification approaches such as decision tree and rule induction with respect to prediction accuracy. Conceptually, AC consists of two parts compared with several traditional rule-based classification algorithms. First, AC generates a number of candidate rules that associate observations with class labels (class association rules; CARs). Second, support and confidence are used as measures to evaluate the significance of CARs (Zhou, 2014).

Specifically, the first part of AC involves rule generation, a process of discovering frequent rule items (CARs) using association rule mining algorithms such as Apriori (Agrawal & Srikant, 1994), Eclat (Zaki, 2000), and FP-growth (Han, Pei, Yin, & Mao, 2004). The second part of AC is a rule evaluation step in which redundant and less useful CARs are removed, involving rule ranking and

* Corresponding author.
E-mail addresses: ksong@hanyang.ac.kr (K. Song), skylee@hanyang.ac.kr (K. Lee).

http://dx.doi.org/10.1016/j.eswa.2017.02.024
0957-4174/

pruning. Candidate CARs are sorted according to specific criteria to establish a priority that decides which rule will be included first in a classifier. The rule-ranking process to evaluate the significance of CARs is important since class prediction for a new observation naturally follows by aggregating the results of several top-ranked CARs. Thus, we hypothesized that the performance of AC will improve if the evaluation of CARs improves, and the effective CARs with high predictability will be selected. Consequently, we present a survey of related previous studies and propose an approach in association rule mining by constructing effective CARs.

Regarding the ranking of CARs, several criteria have been developed. For instance, classification based on associations, CBA (Liu et al., 1998) exploits a heuristic approach to rank candidate rule items with confidence-support-cardinality (CSC), where cardinality indicates the length of the rule antecedent. In the CSC method, confidence is designated as the first priority criterion, support as the second criterion, and cardinality as the last in rank of candidate rules. The authors first compared confidence; then, if two rules had the same confidence value, they compared the support values. Cardinality was analyzed in the same manner. If two or more rules have the same confidence, support, and cardinality, the one created first is chosen. Multi-class classification based on association rules, (MCAR; Thabtah, Cowling, & Peng, 2005) added class distribution to the previous CSC criteria to specify the rule ranking procedure in detail. MCAR considered the frequency of rules through class distribution, which describes the number of occurrences of each class. The rule ranking method was applied to a multi-class associative classification algorithm, (MAC; Abdelhamid, Ayesh, Thabtah, Ahmadi, & Hadi, 2012).

Hernández-León, Carrasco-Ochoa, Martínez-Trinidad, and Hernández-Palancar (2012) proposed a new measure, called *Netconf*, for rule sorting. The values of the proposed measure range between −1 and 1, in which positive values stand for positive relationships, negative values negative relationships, and zero no relationship. In a follow-up study, Hernández-León, Hernández-Palancar, Carrasco-Ochoa, and Martínez-Trinidad (2014) also proposed a hybrid rule ranking combining Netconf with rules of a long length. They revealed that the new strategy with specific rules outperformed the conventional CSC method in the experiments.

Several rule pruning approaches have been proposed to evaluate CARs. A pruning method based on pessimistic error estimation in the C4.5 algorithm (Quinlan, 1993) uses a training set to estimate error rates. Since an estimate of error rates obtained from the training dataset is biased, adjustment of the estimate involves some amount of assumed pessimistic error. Studies by Liu et al. (1998) and Wang, Zhou, and He (2000) adopted this pruning method to reduce the number of CARs. Database coverage (DBC) is a pruning approach proposed in the CBA algorithm (Liu et al., 1998). The approach identifies a set of CARs so that each rule in the set covers at least one instance of training dataset and is used in many AC studies including CBA (2) (Liu, Ma, & Wong, 2000), CAAR (Xu, Han, & Min, 2004), MCAR (Thabtah et al., 2005), MAC (Abdelhamid et al., 2012), and MCAC (Abdelhamid, Ayesh, & Hadi, 2014). A statistical approach to pruning rules has also been applied. The Chi-square test (χ^2) is a statistical test to determine whether two categorical variables are correlated. Classification based on multiple association rules (CMAR; Li, Han, & Pei, 2001) has used a Chi-square test to determine whether the antecedent and the consequent of a rule are correlated and whether the rule will be removed. If positive correlation exists, the CMAR algorithm stores the rule in a classifier, or CMAR deletes the rule from consideration.

The lazy pruning technique (Baralis & Garza, 2002) reduces the number of pruning steps by removing 'harmful rules' that classify each training case into different class labels. Live and let live (L3; Baralis & Garza, 2002) splits the entire set of rules into three types; used rules, spare rules, and harmful rules. The used rules have already been correctly used for classifying the training dataset. The spare rules have not yet been used as rules for classification of training instances. First, the used rules are applied to the original training data, and then the spare rules are applied to the remaining data to widen the rule coverage range for the training instances. Lastly, the harmful rules that misclassified training instances are removed.

Chen, Hsu, and Hsu (2012) used an adjusted scoring based on the best rule pruning method. Their method assigned higher scoring on highest-ranked rules for two class-label problems in consideration of class imbalance. The method is limited in that it basically works with only two class labels and that the ranking is not statistically motivated. Recent rule mining incorporates positive and negative rules (Li & Zaiane, 2015). For instance, a negative CAR is in one of the following forms: $X \rightarrow \neg C$, $\neg X \rightarrow C$, and $\neg X \rightarrow \neg C$, in which X and ¬X represent the presence and absence of X, respectively. Then, each negative CAR, $X \rightarrow \neg C_k$ (C_k corresponding to the kth class label), is pruned if it incorrectly classifies at least one training instance. This approach shares the same principle with the traditional DBC approach, fundamentally.

Most previous rule evaluation approaches only adjusted the number of candidate rules to expand the rule coverage of training cases or to increase the speed of the process and did not focus on the ability to classify an unknown observation from a statistical viewpoint. To the best of our knowledge, studies that determine the predictive power of candidate rules are limited. In this manuscript, we proposed a new approach in CAR mining to increase its overall predictive performance by appropriately applying cross-validation and aggregating the resulting rules.

The remainder of this study is organized as follows. The preliminary concepts in CARs are described in Section 2. Our approach to formation of a collective rule set based on cross-validation with the aim of enhancing prediction performance is proposed in Section 3, and several related works on AC are introduced. In Section 4, we describe the experimental results. In Section 5, the proposed algorithm is applied to a real-life healthcare dataset. In Section 6, we present the conclusion and future work.

2. Preliminaries for class association rule mining

We briefly introduce the definition of the associative classification problem in data mining (Thabtah, Hadi, Abdelhamid, & Issa, 2011). A CAR is a special case of association rule mining in which only the class attribute appears in the rule's consequent (Liu et al., 1998); for example, in a rule such as $X \rightarrow Y$, Y must be a class attribute. We define the AC problem using a training dataset T with m distinct attributes $A_1, A_2, \cdots, A_m$, and C is a list of classes. The number of rows (cases) in T is denoted by $|T|$.

Definition 1. A training case (for the jth row) in T can be described as a combination of observed values a_{ij}, corresponding to attributes A_i $(i = 1, \ldots, m)$ and a class label denoted by C_j, $j = 1, \cdots, |T|$.

Definition 2. An *AttributeValueSet* can be described as a term A_i and value a_i, denoted as $\langle (A_i, a_i) \rangle$. Similarly, it can be described as a set of disjoint attribute values contained in a training case, denoted by $\langle (A_{i1}, a_{i1}), \cdots, (A_{ik}, a_{ik}) \rangle$.

Definition 3. A *rule item r* is of the form $\langle AttributeValueSet, c \rangle$, where $c \subset C$ is the class.

Definition 4. The actual occurrence ($actoccr_T$) of a rule item r in T is the number of rows (cases) in T that match the *AttributeValueSet* of r.

Definition 5. The support count ($suppcount_T$) of a rule item r is the number of rows in T that match r's *AttributeValueSet* and belong to class c of r.

Definition 6. A rule item r passes the *minsupp* threshold if $(\frac{suppcount_T(r)}{|T|}) \geq minsupp$.

Definition 7. A rule item r passes the *minconf* threshold if $(\frac{suppcount_T(r)}{actoccr_T(r)}) \geq minconf$.

Definition 8. Any rule item r passing the *minsupp* threshold is considered a *frequent ruleitem*.

Definition 9. A CAR is represented in the form:

$$(A_{i1}, a_{1i}) \wedge \cdots \wedge (A_{ik}, a_{1k}) \rightarrow c,$$

where the antecedent (rule body/the left hand side) of the rule is an *AttributeValueSet*, and the consequent (the right hand side) is a class.

Next, we briefly introduce the generation of candidate rules from a given transaction data set T. Following Definition 2, *AttributeValueSet*, $\langle (A_{i1}, a_{1i}), \cdots, (A_{ik}, a_{1k}) \rangle$ is generated with the transaction data using frequent itemset mining algorithms such as Apriori or Eclat. According to Definition 3, rule item r $\langle AttributeValueSet, c \rangle$ is distinguished from *AttributeValueSet*. The Apriori algorithm uses a horizontal data format as input and is the most commonly used method for frequent itemset mining. Another widely used algorithm is Eclat, which efficiently handles datasets in a vertical format. Eclat utilizes the lists of transaction identifications, denoted as tid-lists, to calculate the support of candidate ruleitems by interesting the tid-lists of two current ruleitems (Zaki, 2000). In general, tid-lists represented in the vertical display are relatively simple, and the data structure is easy to maintain. Saving I/O time is considered a faster algorithm than Apriori (Zaki, 2000; Zaki & Gouda, 2003). Therefore, we selected Eclat to generate frequent ruleitems. We applied a *minsupp* threshold for the generated ruleitems in order to meet the *minsupp* criterion.

3. Proposed approach: predictability-based collective class association rule (PCAR)

3.1. Overall concept

To form an effective rule set from the generated candidate rules, we applied cross-validation for rule evaluation consisting of rule ranking and pruning. Then, to establish a final classifier, the rules were integrated so that the collective rule set had high predictive power. Consequently, we propose predictability-based collective class association rule (PCAR) mining based on cross-validation.

From the generated frequent ruleitems, those that do not have a class were discarded in order to form CARs. Then, to choose a good rule to classify a new case, the CARs were ranked. Usually, the criteria of *minconf*, the rule-body length, and their variants are applied. The *minconf* criterion does not take into account the class distribution. The interpretation of the rule-body length is not universal and depends on the approach. For example, some prefer short-length rules to capture general relationships between a rule body and a class, while others prefer long-length rules to establish specific description between them. However, in this study, we applied a new ranking rule, prediction power. We proposed a new rule evaluation composed of ranking and pruning methods and PCAR based on cross-validation, as shown in Fig. 1.

For the first phase of PCAR, using the Eclat algorithm, an initial candidate rule set *RS* was generated from training dataset T. To measure the predictability of a rule, we apply cross-validation with k folds. We produce k loops using k equal-length folds: for example, $k=5$, in each loop IL_i $(i = 1, \ldots, 5)$, we create an inner

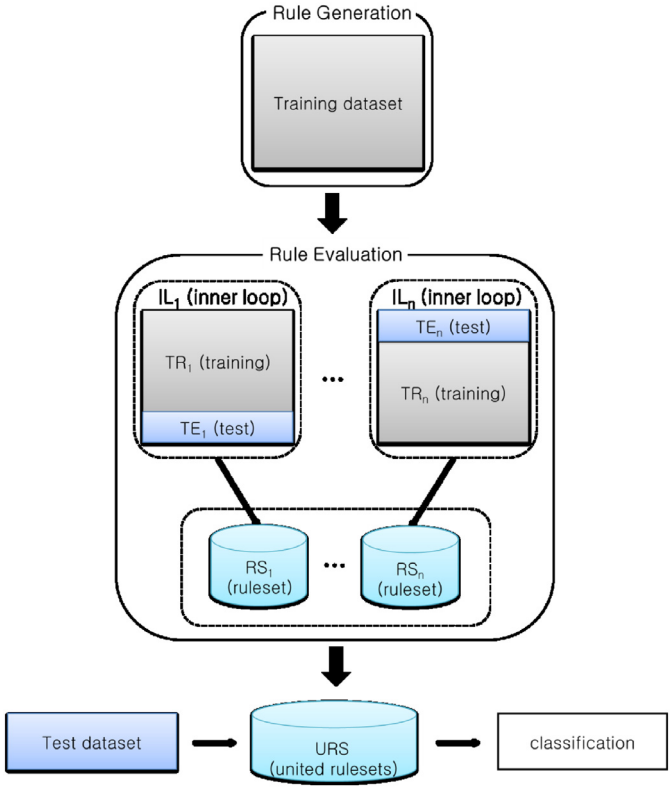

Fig. 1. Overall concept of predictability-based collective class association rule (PCAR) mining.

training set TR_i with four folds and inner test set TE_i with the remaining fold from T. For a rule item r_j in *RS*, we first calculate the ratio of the support count of r_j in TE_i to the size of inner test set TE_i, denoted by $\hat{p}_{j|i}$, as follows:

$$\hat{p}_{j|i} = \frac{suppcount_{TE_i}(r_j)}{|TE_i|}. \quad (1)$$

The quantity $\hat{p}_{j|i}$ is interpreted as a local measure of the rule r_j's prediction performance in inner test set TE_i. Then we define the predictive rate of r_j, denoted by $\hat{p}_j$, as follows:

$$\hat{p}_j = \frac{1}{k} \sum_{i=1}^{k} \hat{p}_{j|i}, \quad (2)$$

which is the average of k local predictability values.

The prediction rates of each rule were calculated to identify high-performance rules in *RS* against T. To sort rules in *RS*, the prediction rate had priority over confidence in the rule-ranking procedure, as shown in Fig. 2. In contrast to previous studies that adopted general criteria discriminating between the rules (CSC; Liu et al., 1998) as mentioned in Section 1, the proposed approach used the first criterion of predictability, focusing on the ability to predict class labels in testing phases.

We view that the rules from the proposed criterion of predictability will be more reliable than those from the previous approaches for the classification task in unknown and future observations. The idea of the proposed criterion may appear comparable to using the confidence and support levels in previous approaches such as CBA and CMAR. However, the previous approaches involved no statistical or computational step to identify good rules.

In the first loop IL_1, RS_1 is likely to contain redundant rules; thus, a database coverage method was applied to remove redundant and useless rules in the rule set. Briefly, the minimum number of rules was determined by selecting rules ordered by the

Given two rules, r_a and r_b, r_a precedes r_b ($r_a > r_b$) if:

1. The predictive rate of r_a is greater than that of r_b

2. The predictive rates of r_a and r_b are the same, but the confidence of r_a is greater than that of r_a.

3. The predictive rates and confidence values of r_a and r_b are the same, but the support of r_a is greater than that of r_b.

4. The predictive rates, confidence and support values of r_a and r_b are the same, but r_a has shorter rule length than r_b.

5. The predictive rates, confidence, support, and rule length of r_a and r_b are the same, but r_a has a more frequently occurring class than r_b.

6. All above criteria are equivalent for r_a and r_b, but r_a was generated before r_b.

Fig. 2. The rule ranking procedure in predictability-based collective class association rule mining (PCAR).

```
# Predictability based evaluation and pruning algorithm, denoted by PCAR
# Input: Training data (T), k (the number of folds in CV)
# Output: United rulesets (URS)

    Apply Eclat algorithm to generate candidate rule sets, RS with T

    # Do k-fold cross-validation
    For i = 1 to k
      Split original training data T into training set TR_i and test set TE_i in the inner loop

      #Predictability
      Calculate the prediction rate of each rule r_j in RS against TE_i

      #Rule ranking
      Order each rule r_j in RS by the rule ranking procedure depicted in Figure 2

      Create initial ruleset RS_i as the null set

    #Database coverage
        For each rule r_j in RS do
          Mark all applicable instances in TR_i that match r_j's body
          If r_j correctly classifies a case in TR_i
            Insert r_j into the RS_i
            Discard all instances in TR_i covered by r_j
          End If
          If r_j covers no instances in TR_i
            Delete r_j
          End If
        End For

        If TR_i is not empty
          Generate a default rule for the largest frequency class in TR_i
        End If

    End For
    Merge RS_i (i = 1, ..., k) into united rulesets, URS
```

Fig. 3. Predictability-based evaluation and pruning algorithm.

ranking procedure until all instances in the training data set were covered by the rules and each rule covered at least an instance. The full-match strategy used in applying database coverage required a candidate rule condition to completely match the attribute values of a training case. The steps are summarized in Fig. 3.

Application of the rule evaluation (rule ranking and pruning) generated one rule set denoted by RS_1. The other candidate rule sets ($RS_2, \cdots, RS_k$) were obtained in the same manner. Finally, all RS_i ($i = 1, \ldots, k$) were united into one set, denoted by *URS*. When we merge all RS_i, if a rule occurs in several rule sets, we use the average of the prediction rates, which is $\hat{p}_j$ in Eq. (2). The use

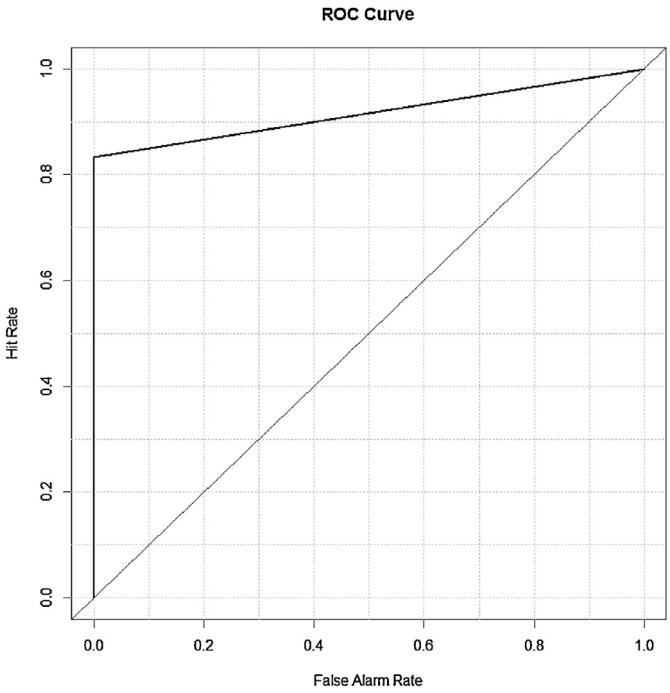

Fig. 4. ROC curve of PCAR for the Atopy dataset.

of prediction rates is reminiscent of prediction confidence which Do, Hui, & Fong (2005) measured by test data. They revealed that the combination of prediction confidence and original confidence performs better than the use of confidence only. The rule predictability, however, is in contrast with prediction confidence in that the rule predictability is an estimated prediction accuracy measured by several test cases restructured in inner loops. The proposed approach prefers maximizing predictability to maximizing confidence.

Next, a final classifier with the *URS* was constructed and used in prediction in such a way that the rule of the highest predictive power matching a new data instance was chosen, and its class label was estimated using the chosen rule. The full-match strategy in the rule-matching was applied at the prediction stage, similar to the rule pruning procedure. If the data instance found no matching rules, the majority class was assigned as the default option.

3.2. Example

Table 1 shows a model dataset with transaction id (TID), three attributes, and one multi-label class. This dataset was split into two sub-samples, the training and test sets, at a 7:3 ratio, where TIDs of test instances were 3, 4, 7, 9, 11, 12, 18, 19, and the others were training instances. Initial rulesets were generated with the training set using the Eclat algorithm, and the final 14 rules were obtained from the proposed algorithm PCAR and the previously suggested MCAR algorithm, as shown in Tables 2 and 3.

For instance, rule r_1 is "*if attribute A1 is X4 and A3 is Z3, then class is C1*." The rules are ranked in descending order according to predictability, confidence, support, cardinality, and class frequency. In accordance with Table 2, rule r_4 correctly covered an instance with TID 18, as shown in Table 1. Rule r_5 covered TID 3 but incorrectly. Rule r_6 covered instances with TID 9 correctly but with 12 incorrectly. We discovered that TIDs 9 and 12, with the same condition, were included in different classes, termed a multi-label class problem. Rule r_{10} incorrectly covered TID 7, but rule r_{13} covered TIDs 4 and 11 correctly. Lastly, in the remaining test instances, TID 19 was incorrectly covered by the default class *C*1. There-

Table 1
Model training and test data demonstrating the proposed algorithm.

TID	A1	A2	A3	Class	Training/Test
1	X2	Y1	Z2	C1	Training
2	X3	Y4	Z1	C1	Training
3	X1	Y3	Z3	C2	Test
4	X4	Y4	Z1	C2	Test
5	X4	Y1	Z2	C3	Training
6	X2	Y2	Z1	C2	Training
7	X3	Y3	Z3	C1	Test
8	X1	Y3	Z2	C1	Training
9	X2	Y2	Z3	C3	Test
10	X3	Y1	Z1	C3	Training
11	X3	Y2	Z1	C2	Test
12	X2	Y2	Z3	C1	Test
13	X4	Y1	Z3	C1	Training
14	X3	Y2	Z3	C3	Training
15	X3	Y3	Z2	C2	Training
16	X4	Y3	Z2	C1	Training
17	X4	Y3	Z3	C1	Training
18	X2	Y1	Z1	C1	Test
19	X1	Y2	Z2	C2	Test
20	X1	Y4	Z2	C3	Training
21	X1	Y1	Z1	C2	Training
22	X3	Y4	Z1	C2	Training
23	X2	Y4	Z2	C3	Training
24	X1	Y4	Z3	C2	Training

Table 2
The rule set generated by PCAR.

ID	Rule	Predictability	Conf*	Supp**	Cardinality	Frequency
1	⟨X4, Z3⟩ → C1	0.4	1	0.125	2	6
2	⟨Y4, Z2⟩ → C3	0.4	1	0.125	2	5
3	⟨X4, Y3⟩ → C1	0.2	1	0.125	2	6
4	⟨X2, Y1⟩ → C1	0.2	1	0.0625	2	6
5	⟨X1, Y3⟩ → C1	0.2	1	0.0625	2	6
6	⟨Y2, Z3⟩ → C3	0.2	1	0.0625	2	5
7	⟨X2, Y2⟩ → C2	0.2	1	0.0625	2	5
8	⟨X1, Z3⟩ → C2	0.2	1	0.0625	2	5
9	⟨X3, Y1⟩ → C3	0.2	1	0.0625	2	5
10	⟨X3, Y3⟩ → C2	0.2	1	0.0625	2	5
11	⟨X1, Y1⟩ → C2	0.2	1	0.0625	2	5
12	⟨X4, Y1, Z2⟩ → C3	0.2	1	0.0625	3	5
13	⟨Z1⟩ → C2	0.12	0.6	0.1875	1	5
14	⟨X3, Y4⟩ → C1	0.1	0.5	0.0625	2	6

* Conf: confidence,
** Supp: support

Table 3
The rule set generated by MCAR.

ID	Rule	Conf*	Supp**	Cardinality	Frequency	Original Rank
1	⟨X4, Z3⟩ → C1	1	0.125	2	6	1
2	⟨X4, Y3⟩ → C1	1	0.125	2	6	3
3	⟨Y4, Z2⟩ → C3	1	0.125	2	5	2
4	⟨X2, Y1⟩ → C1	1	0.0625	2	6	4
5	⟨X1, Y3⟩ → C1	1	0.0625	2	6	5
6	⟨X2, Y2⟩ → C2	1	0.0625	2	5	7
7	⟨Y2, Z3⟩ → C3	1	0.0625	2	5	6
8	⟨X1, Z3⟩ → C2	1	0.0625	2	5	8
9	⟨X3, Y3⟩ → C2	1	0.0625	2	5	10
10	⟨X1, Y1⟩ → C2	1	0.0625	2	5	11
11	⟨X3, Y1⟩ → C3	1	0.0625	2	5	9
12	⟨X4, Y1, Z2⟩ → C3	1	0.0625	3	5	12
13	⟨Z1⟩ → C2	0.6	0.1875	1	5	13
14	⟨X3, Y4⟩ → C1	0.5	0.0625	2	6	14

* Conf: confidence,
** Supp: support

Table 4
Seventeen UCI datasets used in the experiment.

Dataset	Transactions	Attributes	Classes
Balloon	20	4	2
Contact	24	4	3
Tic-tac-toe	958	9	3
Led7	3200	7	10
Balance	625	4	3
Crx	690	9	2
Breast-w	699	9	2
Breast Tissue	106	9	6
Ecoli	336	8	8
Wine	178	13	3
Iris	150	4	3
Pima	768	8	2
Yeast	1484	9	10
Monks1	556	6	2
Monks2	601	6	2
Monks3	554	6	2
Flare	1066	9	6

Table 5
Structures of PCAR and PCAR2.

	PCAR		PCAR2	
	Inner training	Inner test	Inner training	Inner test
Rule ranking	Yes	Yes	No	Yes

fore, the PCAR algorithm showed 50% accuracy, whereas MCAR was 37.5% accurate. We found that TIDs 3 and 9 were incorrectly covered in the MCAR algorithm. As shown in Table 3, since r_6 in MCAR, which was the same as r_7 in PCAR, was used preferably to r_7 in MCAR and matched r_6 in PCAR in a classifier, TID 3 was incorrectly classified.

4. Experiments

4.1. Experimental setting

The input data format is a file composed of a number of records that contain transactional information. A record, called *instance*, has row ids, attributes, and class fields. We adopted vertical data structures because vertical data format reduces the number of scans and therefore the overall computing time compared to a horizontal format (Zaki, 2000; Zaki & Gouda, 2003).

The experiments were conducted using Windows 7 OS, Intel Pentium CPU G2120, 3.1 GHz, 4GB RAM, and R (version 3.1.0). We obtained 16 datasets from the UCI Machine Learning Repository (Lichman, 2013) and one dataset (Flare) from the KEEL-dataset repository (Alcalá-Fdez et al., 2011). Table 4 shows the number of transactions, attributes, and classes of each dataset: datasets were categorical or numerical or both. Although associative classification was originally created to classify only categorical data, MCAR (Thabtah et al., 2005) used not only categorical datasets, but also mixed datasets including categorical and numerical attributes. MCAR converted continuous data to discrete data using a discretization technique (Fayyad & Irani, 1993) to manage numerical data. In accordance to the settings in their approach, we included both categorical and numerical types of dataset and set the rule generation algorithm to have a minimum support of 5% and a minimum confidence of 40%.

To observe the effect of the application scope of predictability in the entire process, we tested another version of PCAR, PCAR2, as summarized in Table 5. Both algorithms utilized inner training and test datasets for the rule generation phase to extract as many initial rulesets as possible. In the rule ranking phase, although PCAR2 utilized only the inner test dataset, PCAR utilized both inner training and test datasets to calculate predictive rates of each rule. For

Table 6
Accuracy (%) of C4.5, RIPPER, CBA, MCAR, CCAR1, and CCAR2 using 10-fold cross validation.

Dataset	C4.5	RIPPER	CBA	MCAR	PCAR2	PCAR
Balloon	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
Contact	**83.33**	79.17	66.67	66.67	75.00	75.00
Tic-tac-toe	85.07	97.81	**100.00**	**100.00**	**100.00**	**100.00**
Led7	73.28	69.38	71.78	71.75	72.94	**73.66**
Balance	63.20	72.32	66.08	75.20	**76.48**	75.52
Crx	86.09	84.93	85.51	**86.23**	**86.23**	85.80
Breast-w	94.42	94.29	89.70	94.85	95.14	**95.57**
Breast Tissue	68.87	63.21	62.26	69.81	**74.53**	70.75
Ecoli	**84.23**	81.25	75.60	75.89	75.30	75.89
Wine	93.82	95.51	88.20	97.19	97.19	**97.75**
Iris	**96.00**	95.33	94.00	93.33	93.33	92.67
Pima	73.83	75.13	**77.47**	76.69	75.00	76.17
Yeast	50.34	**58.42**	53.71	52.96	51.42	53.64
Monks1	97.66	89.39	**100.00**	**100.00**	**100.00**	**100.00**
Monks2	61.40	60.90	63.23	66.39	**74.38**	69.05
Monks3	**98.92**	98.38	98.74	98.74	98.74	98.74
Flare	73.17	68.86	67.07	74.77	**74.86**	74.39
Average	81.39	81.43	80.00	82.38	**83.56**	**83.21**

Table 7
Number of rules in C4.5, RIPPER, CBA, MCAR, CCAR1, and CCAR2 using 10-fold cross validation.

Dataset	C4.5	RIPPER	CBA	MCAR	PCAR2	PCAR
Balloon	3	2	3	3	3	3
Contact	4	3	9	8	8	8
Tic-tac-toe	95	9	26	27	45	27
Led7	37	20	75	199	272	200
Balance	33	13	4	16	20	16
Crx	49	7	13	95	143	97
Breast-w	41	15	23	46	47	46
Breast Tissue	24	7	8	19	22	19
Ecoli	22	8	16	20	27	20
Wine	24	6	3	10	10	10
Iris	5	3	11	14	14	14
Pima	20	3	46	68	115	69
Yeast	49	16	48	59	88	60
Monks1	41	14	4	16	46	16
Monks2	159	2	10	39	74	39
Monks3	19	6	20	23	29	24
Flare	96	9	2	32	45	31

the last phase of rule pruning, both PCAR and PCAR2 utilized only the inner test dataset to remove redundant and low-accuracy rules.

4.2. Experimental results

Table 6 presents the accuracy percentages of C4.5, RIPPER, CBA, MCAR, PCAR, and PCAR2 using 10-fold cross validation. The boldfaced numbers represent the best performance for the dataset. PCAR generally outperformed the tested algorithms in the datasets. We implemented MCAR and our proposed algorithms in R and used other algorithms (C4.5, RIPPER and CBA) in WEKA data mining software (Hall et al., 2009). Overall, our proposed algorithm outperformed the other algorithms. The win-loss-tie record of PCAR against MCAR in terms of accuracy was 8-4-5. This record of PCAR against C4.5, RIPPER, and CBA was 11-5-1, 12-4-1, and 10-3-4, respectively. Previous results reported by Thabtah et al. (2005) showed that MCAR outperformed C4.5 and Ripper. In the present experiment, however, MCAR showed worse performance than C4.5 and Ripper. We hypothesized that prediction performances were impacted by the characteristics of datasets. The win-loss-tie record of PCAR against PCAR2 was 6-6-5. We discovered that overall PCAR2 and PCAR outperformed the other algorithms: the boldfaced numbers in Table 6 represent the best performance. Table 7 shows the number of rules in C4.5, RIPPER, CBA, MCAR, PCAR, and PCAR2 using 10-fold cross validation. The total number of rules increased in the following order; RIPPER, CBA, C4.5, MCAR, PCAR, and PCAR2.

The result showed that the performance of PCAR was similar to that of PCAR2 and the number of best-performance datasets of PCAR outrun that of PCAR2. We selected PCAR more favorable than PCAR2 because the application scope of in PCAR was wider than in PCAR2. In addition, PCAR, producing a smaller number of rules than PCAR2, found more predictable, effective, rules than PCAR2. We noticed that PCAR consistently surpassed MCAR in accuracy for the majority of tested datasets, while the number of rules in PCAR was quite similar to that in MCAR.

Our proposed approach used an existing algorithm such as Eclat for the rule generation step. Nevertheless, it created more rules in the process of rule ranking and pruning than the algorithms in comparison (C4.5, RIPPER, CBA, and MCAR) by introducing a new rule evaluation measure, predictability. Table 6 shows that the proposed algorithms (PCAR and PCAR2) increased overall classification accuracy comparing with others. Therefore, the experimental results reveal that our proposed algorithm recovered more useful rules and assigned proper priorities to those.

5. Application of the proposed algorithm

We applied the proposed algorithm to a dataset of participants with/without atopic dermatitis (hereafter, Atopy dataset) during 2003–2013: the dataset had 49 attributes and 7504 observations, represented as a 7504×49 matrix. The attributes were conditions of a participant; for example, sex, age, existence of *Dermatophagoides pteronssinus* (Dp), existence of *Dermatophagoides farina* (Df), and existence of *Tyrophagus* (Tyro). All attributes were binary data except age. We needed to select appropriate features among the 49 attributes since several attribute column matrices were sparse. We first selected the two attributes of sex and age as defaults. We counted the frequency of each attribute in the matrix, identified the attributes that belong to the upper 10% of frequency, and selected four additional attributes (Dp, Df, Cat, and Mugwort) as input attributes. The class label name was 'Atopy': if the participant had atopic dermatitis, then we set $\langle \text{Atopy} = 1 \rangle$.

We applied the proposed algorithm with 10-fold CV for classification since it performed best, as described in Section 4. The average number of rules per fold was approximately 25, and the rules with a predictive rate greater than 90% are shown in Table 8. In addition, we added several performance indicators of the algorithm to the Atopy dataset: accuracy, sensitivity, specificity, precision, F-score, and area under the curve (AUC) are shown in Table 9, and the receiver operating characteristic curve (ROC) is shown in Fig. 2. Based on Table 9 and Fig. 2, the algorithm showed an effective rule set in understanding the relationship between atopic syndrome and its possible symptoms.

6. Conclusion

In this study, we described the limitations of previous rule evaluation methods including ranking and pruning for development of outperforming associative classifiers. As a result, we proposed a novel and outperforming CAR evaluation method and described its advantages.

Previous algorithms use simple heuristic techniques such as database coverage and lazy pruning or scoring methods such as Chi-square test (χ^2) and pessimistic error for the rule evaluation phase. However, the proposed PCAR algorithms calculated the predictive power of candidate rules, indicating that our approach removed not only redundant rules, but also rules with low predictive power. The experimental results revealed that the accuracy of the proposed algorithm was higher than that of MCAR even though a

Table 8
The portion of rules with a predictive rate higher than 90%.

ID	Rule	Accuracy
1	$\langle Sex = 0, Dp = 1\rangle \rightarrow \langle Atopy = 1\rangle$	1.0000
2	$\langle Cat = 1\rangle \rightarrow \langle Atopy = 1\rangle$	1.0000
3	$\langle Age = 1, Df = 1\rangle \rightarrow \langle Atopy = 1\rangle$	1.0000
4	$\langle Age = 1, Dp = 1\rangle \rightarrow \langle Atopy = 1\rangle$	1.0000
5	$\langle Age = 3, Df = 1\rangle \rightarrow \langle Atopy = 1\rangle$	1.0000
6	$\langle Age = 3, Dp = 1\rangle \rightarrow \langle Atopy = 1\rangle$	1.0000
7	$\langle Dp = 1\rangle \rightarrow \langle Atopy = 1\rangle$	0.9995
8	$\langle Sex = 1, Df = 1\rangle \rightarrow \langle Atopy = 1\rangle$	0.9993
9	$\langle Df = 1\rangle \rightarrow \langle Atopy = 1\rangle$	0.9991
10	$\langle Mugwort = 1\rangle \rightarrow \langle Atopy = 1\rangle$	0.9961
11	$\langle Sex = 0, Age = 6, Dp = 0, Df = 0, Cat = 0, Mugwort = 0\rangle \rightarrow \langle Atopy = 0\rangle$	0.9363
12	$\langle Age = 6, Dp = 0, Df = 0, Cat = 0, Mugwort = 0\rangle \rightarrow \langle Atopy = 0\rangle$	0.9242

Table 9
Performance results for the Atopy dataset.

	Accuracy	Sensitivity	Specificity	Precision	F-score	AUC
Value	0.9203	0.9990	0.8333	0.8689	0.9294	0.9161

similar number of rules was produced. We demonstrated adequate PCAR performance when applied to the Atopy dataset.

Despite the aforementioned advantages, the proposed algorithm has several limitations. The execution speed of the proposed algorithm is slower than MCAR since it includes the inner cross validation phase for the calculation of rules' predictive power. In specific, when the number of attributes is large, the execution speed deteriorates considerably. However, this issue is universal to algorithms in association rule mining. We envision that the proposed algorithm will be extended to effectively handle multi-class labels, numerical attributes, and missing observations. Since the proposed rule mining is able to produce interpretable decision criteria while maintaining good performance, we also envision that the algorithm will be applied to classification and pattern recognition in sequence and time series data.

Acknowledgments

This research was supported by the grant (C0443077) funded by Small and Medium Business Administration (SMBA) in the Republic of Korea and Korea Association of University, Research Institute and Industry (AURI). This research was also supported by the National Safety Promotion Technology Development Program (201600000002094, Smart crime prevention solution development through machine learning based on Image Big Data), funded by the Ministry of Trade, Industry and Energy (MOTIE).

References

Abdelhamid, N., Ayesh, A., Thabtah, F., Ahmadi, S., & Hadi, W. (2012). MAC: A multiclass associative classification algorithm. *Journal of Information & Knowledge Management, 11*(02), 1250011.

Abdelhamid, N., Ayesh, A., & Hadi, W. (2014). Multi-label rules algorithm based associative classification. *Parallel Processing Letters, 24*(01), 1450001.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th international conference on very large data bases: 1215* (pp. 487–499).

Alcalá-Fdez, J., Fernandez, A., Luengo, J., Derrac, J., García, S., Sánchez, L., et al. (2011). KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing, 17*(2-3), 255–287.

Baralis, E., & Garza, P. (2002). A lazy approach to pruning classification rules. In *Proceedings of the 2002 IEEE international conference on data mining* (pp. 35–42).

Chen, W. C., Hsu, C. C., & Chu, Y. C. (2012a). Increasing the effectiveness of associative classification in terms of class imbalance by using a novel pruning algorithm. *Expert Systems with Applications, 39*(17), 12841–12850.

Chen, W. C., Hsu, C. C., & Hsu, J. N. (2012b). Adjusting and generalizing CBA algorithm to handling class imbalance. *Expert Systems with Applications, 39*(5), 5907–5919.

Chen, F., Wang, Y., Li, M., Wu, H., & Tian, J. (2014). Principal association mining: An efficient classification approach. *Knowledge-Based Systems, 67*, 16–25.

Cohen, W. W. (1995). Fast effective rule induction. In *Proceedings of the twelfth international conference on machine learning* (pp. 115–123).

Do, T. T., Hui, S. C., & Fong, A. C. M. (2005). Prediction confidence for associative classification. In *Proceedings of fourth international conference on machine learning and cybernetics* (pp. 1993–1998).

Fayyad, U. M., & Irani, K. B. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the 13th international joint conference on artificial intelligence* (pp. 1022–1029).

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter, 11*(1), 10–18.

Han, J., Pei, J., Yin, Y., & Mao, R. (2004). Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery, 8*(1), 53–87.

Hernández-León, R., Carrasco-Ochoa, J. A., Martínez-Trinidad, J. F., & Hernández–Palancar, J. (2012). CAR-NF: A classifier based on specific rules with high netconf. *Intelligent Data Analysis, 16*(1), 49–68.

Hernández-León, R., Hernández-Palancar, J., Carrasco-Ochoa, J. A., & Martínez-Trinidad, J. F. (2014). Studying netconf in hybrid rule ordering strategies for associative classification. In *Proceedings of the 6th Mexican conference on pattern recognition* (pp. 51–60).

Li, W., Han, J., & Pei, J. (2001). CMAR: Accurate and efficient classification based on multiple-class association rule. In *Proceedings of the 1st IEEE international conference on data mining* (pp. 369–376).

Li, J., & Zaiane, O. (2015). Associative classification with statistically significant positive and negative rules. In *Proceedings of the 24th ACM international on conference on information and knowledge management* (pp. 633–642).

Lichman, M. (2013). *UCI Machine Learning Repository*. Irvine, CA: University of California, School of Information and Computer Science *http://archive.ics.uci.edu/ml*.

Liu, B., Hsu, W., & Ma, Y. (1998). Integrating classification and association rule mining. In *Proceedings of the fourth international conference on knowledge discovery and data mining* (pp. 80–86).

Liu, B., Ma, Y., & Wong, C. K. (2000). Improving an association rule based classifier. In *Proceedings of the 4th European conference on principles of data mining and knowledge discovery* (pp. 504–509).

Quinlan, J. R. (1993). *C4.5: programs for machine learning*. San Mateo: Morgan Kaufmann.

Quinlan, J. R., & Cameron-Jones, R. M. (1993). FOIL: A midterm report. In *Proceedings of the European conference on machine learning* (pp. 1–20).

Thabtah, F. (2007). A review of associative classification mining. *The Knowledge Engineering Review, 22*(01), 37–65.

Thabtah, F., Cowling, P., & Peng, Y. (2004). MMAC: A new multi-class, multi-label associative classification approach. In *Proceedings of the fourth IEEE international conference on data mining* (pp. 217–224).

Thabtah, F., Cowling, P., & Peng, Y. (2005). MCAR: Multi-class classification based on association rule. In *Proceedings of the 3rd ACS/IEEE international conference on computer systems and applications* (pp. 1–7).

Thabtah, F., Hadi, W., Abdelhamid, N., & Issa, A. (2011). Prediction phase in associative classification mining. *International Journal of Software Engineering and Knowledge Engineering, 21*(06), 855–876.

Wang, K., Zhou, S., & He, Y. (2000). Growing decision trees on support-less association rules. In *Proceedings of the 6th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 265–269).

Xu, X., Han, G., & Min, H. (2004). A novel algorithm for associative classification of image blocks. In *Proceedings of the 4th international conference on computer and information technology* (pp. 46–51).

Zaki, M. J. (2000). Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering, 12*(3), 372–390.

Zaki, M., & Gouda, K. (2003). Fast vertical mining using diffsets. In *Proceedings of the 9th ACM international conference on knowledge discovery and data mining* (pp. 326–335).

Zhou, Z. (2014). A new classification approach based on multiple classification rules. *Mathematical Problems in Engineering, 2014*, 818253.