CARIM: An Efficient Algorithm for Mining Class-Association Rules with Interestingness Measures

Loan Nguyen[1,2], Bay Vo[3], and Tzung-Pei Hong[4,5]
[1]Division of Knowledge and System Engineering for ICT, Ton Duc Thang University, Vietnam
[2]Faculty of Information Technology, Ton Duc Thang University, Vietnam
[3]Faculty of Information Technology, Ho Chi Minh City University of Technology, Vietnam
[4]Department of CSIE, National University of Kaohsiung, Taiwan
[5]Department of CSE, National Sun Yat-sen University, Taiwan

Abstract: *Classification based on association rules can often achieve higher accuracy than some traditional rule-based methods such as C4.5 and ILA. The right-hand-side part of an association rule is a value of the target (or class) attribute. This study proposes a general algorithm for mining class-association rules based on a variety of interestingness measures. The proposed algorithm uses a tree structure for maintaining the related information of itemsets in the nodes, thus speeding up the process of generation of rules. The proposed algorithm can be easily extended to integrate some measures together for ranking of rules. Experiments are also conducted to show the efficiency of the proposed approach under various settings.*

Keywords: *Accuracy, classification, class-association rule, interestingness measure, integration.*

Received July 19, 2012; accepted September 27, 2012; published online August 16, 2015

1. Introduction

Recently, a new method for classification from data mining, called the Classification Based on Associations (CBA), has been proposed for mining Class Association Rules (CARs). This method has more advantages than heuristic and greedy methods. It cannot only easily remove noise, but also generate a rule set that is more complete than C4.5 and ILA. Thus, some algorithms for CBA rule mining have been proposed.

The first algorithm, called CBA, was proposed by Liu *et al*. [13]. It found classification rules based on association rule mining. Several algorithms for mining CARs have then been proposed, such as CPAR [36], CMAR [11], CBA [13], MMAC [27], MCAR [25, 26], ACME [28], Noah [4], ECR-CARM [33], CSMC [15] and genetic algorithm based approaches [8, 21] Classifiers based on CARs have been shown to be more accurate than traditional methods such as C4.5 and ILA [22, 29, 30] in both theoretic [31, 32] and experimental studies [13].

Interestingness measures play an important role in association rule mining. They can be used for ranking of association rules. Tan *et al*. [24] found that no one measure was best in all application domains. Therefore, a general algorithm for mining CARs and their interestingness measure values is important for determining an appropriate measure for a given dataset.

This study thus proposes an efficient algorithm for mining all CARs along with their measure values for any interestingness measure. The proposed algorithm uses a tree structure for maintaining the related information in nodes to efficiently compute the measure value in a node. The proposed algorithm can also be extended to integrate multiple interestingness measures. Experimental results show that the execution time required for computing ten measures is nearly the same as that required for computing one measure.

The rest of this paper is organized as follows. Some works related to mining CARs and interestingness measures are reviewed in section 2. Preliminary concepts about CARs and interestingness measures are introduced in section 3. The developed MECR-tree data structure and the proposed algorithm based on it for mining CARs with various interestingness measures are described in section 4. Section 5 discusses the advantages of the proposed algorithm. The experimental results are shown and discussed in section 6. The conclusions and future work are given in section 7.

2. Related Works

2.1. Mining Class-Association Rules

CARs are mined to discover all classification rules that satisfy given minimum support (minSup) and minimum confidence (minConf) thresholds. The first method for mining CARs was proposed by Liu *et al.* [13]. It first generates all 1-ruleitems, where a rule item has the form <*condset*, *y*>, where *condset* is a set of items and *y* is a class label. This method then generates all candidate 2-ruleitems from the frequent 1-ruleitems and finds the large 2-ruleitems. This process is repeated until no more candidates are obtained. A heuristic algorithm is then used for building a classifier. The rule set is first sorted in descending order according to the confidence and support values. The algorithm then considers each rule for the dataset. A rule is chosen from the rule set if it satisfies at least one record. A record is deleted from the dataset if it matches at least one rule (i.e., the record is not considered in the next run). The weakness of the above approach is that a lot of candidates might be generated and the dataset might be scanned many times, making the process very time-consuming. Therefore, the algorithm uses a threshold K and only generates k-ruleitems with $k \le K$. Liu *et al.* [14], then proposed an improved algorithm for solving the problem of imbalanced datasets by using multiple minimum support thresholds. This method is more accurate since, a hybrid approach is used for prediction. Li *et al.* [11] then used the FP-tree structure to speed up the CBA mining process. A dataset is scanned only twice and a tree structure is used to compress it. A tree-projection technique is used to find frequent itemsets. To predict a new record, the method finds all the rules that satisfy the record and uses the weighted χ^2 measure to determine the class. Yin and Han [36] proposed the CPAR algorithm for prediction. This method used expected accuracy to evaluate rules and used the best k rules for prediction. Thabtah *et al.* [27] then proposed a multi-class, multi-label associative classification method for mining CARs. A rule in this method is in form of $\{(A_{i1}, a_{i1}), (A_{i2}, a_{i2}), \ldots, (A_{im}, a_{im})\} \rightarrow c_{i1} \vee c_{i2} \vee \ldots \vee c_{il}$, where a_{ij} is a value of attribute A_{ij} and c_{ij} is a class label. Thabtah *et al.* [26] also proposed the MCAR algorithm to improve the accuracy and the mining time. Vo and Le [33] then developed a tree structure called equivalence class rule tree and proposed an algorithm named ECR-CARM for mining CARs. The algorithm is based on the intersection of object identifications to efficiently compute the support values of itemsets. The dataset is scanned only once. Nguyen *et al.* [16] proposed a lattice-based approach for efficiently pruning redundant rules based on the lattice structure.

Some other class-association rule mining approaches were proposed by Coenen *et al.* [3, 4, 6, 12, 13, 20, 28, 37, 38].

2.2. Interestingness Measures

An interestingness measure is a metric used for measuring the strength of a rule. Often, only rules with the k highest values of the measure are maintained for prediction. Several interestingness measures have been developed for ranking rules.

Piatetsky-Shapiro [19] applied statistical independence as an interestingness measure. Agrawal and Srikant [1] proposed support and confidence measures for mining association rules and designed a mining algorithm. Hilderman *et al.* [5, 24] compared various interestingness measures. Lee *et al.* [9, 17] found out that confidence, coherence and cosine measures were beneficial for mining correlation rules in transaction databases. Tan *et al.* [24] discussed the properties of 21 objective interestingness measures and analysed the impact on candidate pruning based on the support threshold. No one measure is best in all application domains and some measures are correlated to each other [7, 24]. Shekar and Natarajan [23] proposed three measures for determining the relations between item pairs. Some studies discussed how to choose appropriate measures for a given database [2, 10, 24]. Huynh *et al.* [7] introduced 35 interestingness measures for mining association rules. Vo and Le [34] proposed an algorithm for rapidly mining interesting association rules by combining lattices and hash tables. Yafi *et al.* [35] proposed a shocking measure for mining association rules.

3. Preliminary Concepts

Let D be a set of training data with n attributes $A_1, A_2, \ldots, A_n$ and $|D|$ objects (cases). Let $C = \{c_1, c_2, \ldots, c_k\}$ be a set of class labels. Some definitions used in this study are given below:

- ***Definition*** 1: An itemset is a set of m attribute-value pairs, denoted $\{(A_{i1}, a_{i1}), (A_{i2}, a_{i2}), \ldots, (A_{im}, a_{im})\}$, where A_{ij} is an attribute and a_{ij} is one of the values of A_{ij}.
- ***Definition*** 2: A class-association rule r has the form of $\{(A_{i1}, a_{i1}), \ldots, (A_{im}, a_{im})\} \rightarrow c$, where $\{(A_{i1}, a_{i1}), \ldots, (A_{im}, a_{im})\}$ is an itemset and $c \in C$ is a class label.
- ***Definition*** 3: The actual occurrence $ActOcc(r)$ of a rule r in D is the number of records in D that match *r's* condition.
- ***Definition*** 4: The support of a class-association rule r, denoted $Sup(r)$, is the number of records in D that match *r's* condition and belong to the class of r.

Example: Table 1 contains eight records with OIDs (object identifiers) from 1 to 8, three attributes named A, B, C, and a class attribute named *class*. Consider the rule $r=\{(A, a1)\} \rightarrow y$ for the dataset in Table 1. Since, there are three records with the attribute A and two of them belong to the class Y, $ActOcc(r)$=3 and $Sup(r)$=2.

Table 1. A dataset as an example.

OID	A	B	C	Class
1	a1	b1	c1	Y
2	a1	b2	c1	N
3	a2	b2	c1	N
4	a3	b3	c1	Y
5	a3	b1	c2	N
6	a3	b3	c1	Y
7	a1	b3	c2	Y
8	a2	b2	c2	N

An association rule can be expressed as $X \xrightarrow{q,vm} Y$, where $X \cap Y = \varnothing$, $q=Sup(X \cup Y)$, and vm is a measure value. For example, in traditional association rules, vm is the confidence of the rule, which is evaluated as $Sup(X \cup Y)/Sup(X)$. Let $vm(n, n_X, n_Y, n_{XY})$ be the measure value of the rule $X \rightarrow Y$, where the four variables represent the number of objects in D, and the numbers of objects with X, Y and $X \cup Y$, respectively. The measure vm can thus be computed based on n, n_X, n_Y, and n_{XY}. For example, consider the rule $\{(A, a1)\} \rightarrow y$ obtained from Table 1. For this rule, $X=\{(A, a1)\}$,

$Y=y$, $n=8$ (number of objects), $n_X=3$, $n_Y=4$, and $n_{XY}=2$. Some extended parameters can also be calculated as $n_{\bar{X}}=5$, $n_{\bar{Y}}=4$, and $n_{X\bar{Y}}=1$. Several measures based on these parameters and their values for the example are listed in Table 2.

Table 2. Some measures and values for the example

No.	Measure	Equation	Value for the Example
1	Confidence [1]	$\frac{n_{XY}}{n_X}$	$\frac{2}{3}$
2	Cosine [24]	$\frac{n_{XY}}{\sqrt{n_X n_Y}}$	$\frac{2}{\sqrt{3\times 4}}=\frac{1}{\sqrt{3}}$
3	Lift [18]	$\frac{n_{XY}\,n}{n_X\,n_Y}$	$\frac{2\times 8}{3\times 4}=\frac{4}{3}$
4	Rule Interest [19]	$\frac{n_X n_{\bar{Y}} - n_{X\bar{Y}} n}{n^2}$	$\frac{3\times 4-1\times 8}{64}=\frac{1}{16}$
5	Laplace [24]	$\frac{n_{XY}+1}{n_X+2}$	$\frac{3}{5}$
6	Jaccard [24]	$\frac{n_{XY}}{n_X+n_Y-n_{XY}}$	$\frac{2}{3+4-2}=\frac{2}{5}$
7	Phi-Coefficient [24]	$\frac{n_{XY}\,n - n_X n_Y}{\sqrt{n_X n_Y n_{\bar{X}} n_{\bar{Y}}}}$	$\frac{2\times 8-3\times 4}{\sqrt{3\times 4\times 5\times 4}}=\frac{1}{\sqrt{15}}$

The values of these interestingness measures for the example are different. Some are larger than 1 such as those obtained by Lift, etc., some are from 0 to 1 such as confidence and cosine, etc.

4. Mining Class-Association Rules with Interestingness Measures

In this section, an algorithm is proposed for CAR rules. The algorithm can be applied for any given interestingness measure for mining CARs. A Modified ECR-tree (MECR-tree) is first described. In the original ECR-tree, each node contains all the itemsets that belong to the same attributes. Here, each node is modified to contain only one itemset. The algorithm that joins each node with all the nodes that have the same equivalence class is then proposed. Finally, a process of mining CARs from the dataset with the Jaccard measure is presented.

4.1. MECR-Tree Structure

The MECR-tree structure is a modified version of the ECR-tree structure [34] for mining CARs with support and confidence measures. In the original ECR-tree structure, all the itemsets with the same attributes are clustered into one group. Itemsets in the each group are joined with all the itemsets that belong to the groups following it, which makes the process of generating and checking candidates time-consuming. In the proposed MECR-tree structure, each node in the tree contains an itemset including the following information:

- Obidset: A set of objects containing the itemset.
- $Count_i$: The number of objects containing the itemset and belonging to class i, for $i\in[1, k]$, where k is the number of classes.

Obidset is stored for fast computing the supports of itemsets. When joining two itemsets to create a new itemset, we can get the support of the new itemset by computing the intersection between two obidsets. The variable count is stored for fast computing measure values of rules that are generated from this itemset.

For example, consider the node containing the itemset $X=\{(A, a2), (B, b2)\}$ for the dataset in Table 1. Since, X is contained in Objects 3 and 8, and both of them belong to class n, a node $\underset{38(0,\,2)}{\{(A, a2), (B, b2)\}}$ is generated in the tree to represent the itemset, where 38 represents objects 3 and 8, and (0, 2) represents 0 (no) object belongs to class y and 2 objects belong to class n. The above representation can be further simplified as $\underset{38(0,\,2)}{AB \times a2b2}$. In an actual implementation, the bit presentation is used for storing the attributes of an itemset. For example, the itemset AB can be coded as 11, with the first and the second bits representing A and B, respectively. Therefore, the value of the attributes is 3 and node $\underset{38(0,\,2)}{AB \times a2b2}$ can be rewritten as $\underset{38(0,\,2)}{3 \times a2b2}$. With this representation, bitwise operations can be used to rapidly join itemsets.

With these descriptions, the itemset is divided into two parts, namely atts and vals. The former is a bit representation of attributes containing this itemset and the latter is a set of values that belong to this itemset.

A vertex in MECR-tree connects node X to node Y if the itemset of X is a prefix of the itemset of Y. For example, node $\underset{127(2,\,1)}{1 \times a1}$ is connected to node $\underset{1(1,\,0)}{3 \times a1b1}$, but node $\underset{15(1,\,1)}{2 \times b1}$ is not connected to node $\underset{1(1,\,0)}{3 \times a1b1}$.

4.2. Proposed Algorithm

In this section, an algorithm called the Class-Association Rule with Interestingness Measure (CARIM) is proposed for efficiently mining CARs from a given training dataset. The algorithm is shown in Algorithm 1. It considers each node l_i with all the other nodes l_j in L_r, with $j>i$ (Lines 2 and 5), to generate a candidate child node l. With each pair (l_i, l_j), the algorithm checks whether l_i.atts≠l_j.atts (Line 6). If they are different, it computes the four elements *atts*, vals, obidset, and count for the new node l (Lines 7-9). If the number of object identifiers is larger than zero (Line 10), then the algorithm computes the count of the objects in each class that contains *l.itemset* and adds this node to P_i (P_i is initialized as empty on Line 4). Finally, CARIM is recursively called with a new set P_i as its input parameter (Line 14).

Algorithm 1: CARIM algorithm for mining CARs.

Input: *A dataset and a given interestingness measure vm.*
Output: *CARs and their measure values.*

Procedure:

```
CARIM(P, minSup)
CAR=∅;
for all li ∈ P do
    CAR=CAR∪ENUMERATE_RULE_IM(li); //find the strongest
                                    rule from li
    Pi=∅;
    for all lj∈P, with j>i do
        if li.atts≠lj.atts then
            l.atts=li.atts∪lj.atts;          //use bitwise operation
            l.vals= li.vals∪lj.vals;
            l.Obidset=li.Obidset∩ lj.Obidset;
            if |l.Obidset| > 0 then        // l.itemset exists in the dataset
                for all ob∈O.Obidset do       //compute O.count
                    O.count[ob]++;
                Pi=Pi∪l;              //add l into the set of nodes Pi
    CARIM(Pi, minSup); //call recursive to generate all children
                          nodes of Pi
ENUMERATE_RULE_IM(l)
    CARl=∅;
    for i∈[1, k] do              //traverse all classes of the dataset
        if l.count[i] > 0 then     // l.itemset contains at least one row
                                     belongs to class i
    nX=|l.Obidset|;
    nXY=l.count[i];
    nY=Count[i];
    CARl=CARl∪{l.itemset → ci (l.count[i], vm(n, nX, nY, nXY)};
return The rule with highest information from CARl;
```

The function ENUMERATE_RULE_IM(l) generates interesting rules from the node l. It first traverses each class (Line 16) to generate rules. If the count of this class is larger than zero (Line 17), it means that l can generate a rule from $l.itemset \rightarrow c_i$. The function then computes the parameter values for this rule, including n_X, n_Y and n_{XY} (Lines 18-20), where X is *l.itemset* and Y is class c_i. To get the support of X, the cardinality of its Obidset is counted. The support of Y (*Count*[i]) and n (number of objects) can be obtained when the dataset is scanned. After the four elements are obtained, the value of any measure adopted can be easily calculated (Line 21). Finally, the function returns the rule with the highest measure from the rule set CAR_l (Line 22).

4.3. An Example

The example in Table 1 is used here to describe the process of the CARIM algorithm with the Jaccard measure. Figure 1 shows the MECR-tree constructed from the dataset in Table 1, where the number before the symbol '×' is bit-presentation of the attributes.

Table 3 shows the execution process of generating rules from the nodes in Figure 1. The rules in bold are the strongest among the ones generated from the corresponding nodes in the tree.

Table 3. The process of mining CARs with the jaccard measure.

ID	Node	Generated Rules	n_X	n_Y	n_{XY}	Jaccard Measure
1	1×a1	**If *A = a1* then class = *y*** If *A = a1* then class = *n*	3 3	4 4	2 1	2/5 1/6
2	1×a2	If *A = a2* then class = *n*	2	4	2	2/4
3	1×a3	**If *A = a3* then class = *y*** If *A = a3* then class = *n*	3 3	4 4	2 1	2/5 1/6
4	2×b1	If *A = b1* then class = *y* If *A = b1* then class = *n*	2 2	4 4	1 1	1/4 1/4
5	2×b2	If *B = b2* then class = *n*	3	4	3	3/4
6	2×b3	If *B = b3* then class = *y*	3	4	3	3/4
7	4×c1	**If *C = c1* then class = *y*** If *C = c1* then class = *n*	5 5	4 4	3 2	3/6 2/7
8	4×c2	If *C = c2* then class = *y* If *C = c2* then class = *n*	3 3	4 4	1 2	1/6 2/5
9	3×a1b1	If *A = a1* and *B = b1* then class = *y*	1	4	1	1/4
10	3×a1b2	If *A = a1* and *B = b2* then class = *n*	1	4	1	1/4
11	3×a1b3	If *A = a1* and *B = b3* then class = *y*	1	4	1	1/4
12	5×a1c1	If *A = a1* and *C = c1* then class = *y* If *A = a1* and *C = c1* then class = *n*	2 2	4 4	1 1	1/4 1/4
13	5×a1c2	If *A = a1* and *C = c2* then class = *y*	1	4	1	1/4
14	3×a2b2	If *A = a2* and *B = b2* then class = *n*	2	4	2	2/4
15	5×a2c1	If *A = a2* and *C = c1* then class = *n*	1	4	1	1/4
16	5×a2c2	If *A = a2* and *C = c2* then class = *n*	1	4	1	1/4
17	3×a3b1	If *A = a3* and *B = b1* the class = *n*	1	4	1	1/4
18	3×a3b3	If *A = a3* and *B = b3* then class = *y*	2	4	2	2/4
19	5×a3c1	If *A = a3* and *C = c1* then class = *y*	2	3	2	2/4
20	5×a3c2	If *A = a3* and *C = c2* then class = *n*	1	4	1	1/4*
21	6×b1c1	If *B = b1* and *C = c1* then class = *y*	1	4	1	1/4*
22	6×b1c2	If *B = b1* and *C = c2* then class = *n*	1	4	1	1/4*
23	6×b2c1	If *B = b2* and *C = c1* then class = *n*	2	4	2	2/4*
24	6×b2c2	If *B = b2* and *C = c2* then class = *n*	1	1	1	1/4*
25	6×b3c1	If *B = b3* and *C = c1* then class = *y*	2	4	2	2/4*
26	6×b3c2	If *B = b3* and *C = c2* then class = *y*	1	4	1	1/4*
27	7×a1b1c1	If *A = a1* and *B = b1* and *C = c1* then class = *y*	1	4	1	1/4*
28	7×a1b2c1	If A = a1 and B = b2 and C = c1 then class = n	1	4	1	1/4*
29	7×a1b3c2	If A = a1 and B = b3 and C = c2 then class = y	1	4	1	1/4*
30	7×a2b2c1	If A = a2 and B = b2 and C = c1 then class = n	1	4	1	1/4*
31	7×a2b2c2	If A = a2 and B = b2 and C = c2 then class = n	1	4	1	1/4*
32	7×a3b1c2	If A = a3 and B = b1 and C = c2 then class = n	1	4	1	1/4*
33	7×a3b3c1	If A = a3 and B = b3 and C = c1 then class = y	2	4	2	2/4*

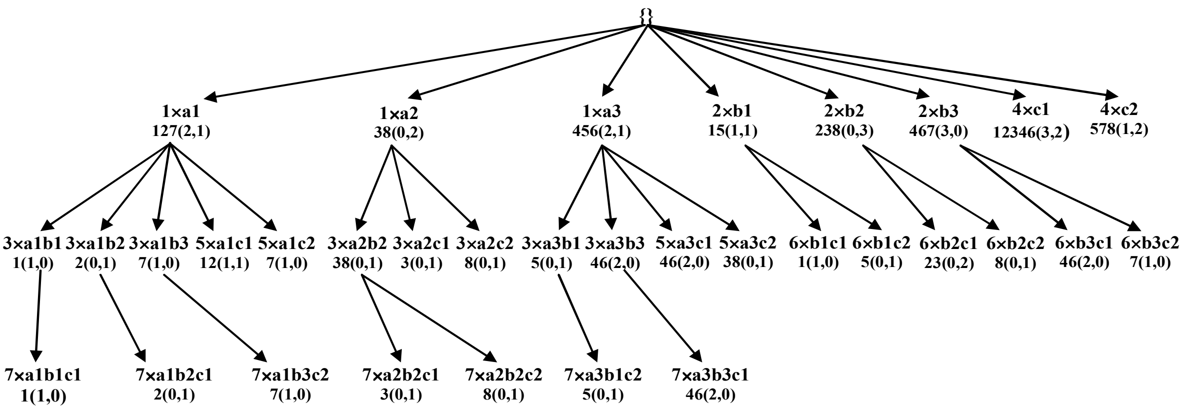

Figure 1. The MECR-tree constructed from the dataset in table 1.

5. Discussions

The proposed algorithm has the following advantages. Firstly, it uses a tree structure for maintaining node information, which speeds up the process of generating rules. The information includes obidset and count. With obidset, we can get the support of a new itemset fast by computing the intersection of two obidsets. With count, we can use this information to compute the measure values and choose the best one. Secondly, obidset allows the count of each class to be rapidly calculated. With these counts, the algorithm can easily determine the rules with the highest measure value. Thirdly, the proposed algorithm can be used for mining CARs with any interestingness measure. The proposed algorithm can also integrate some measures together for ranking of rules. In this case, only Lines 21 and 22 in the proposed algorithm need to be modified.

6. Experimental Results

The experimental datasets have different characteristics. The Breast, German, and Vehicle datasets have many attributes and distinct items, but small numbers of objects. The Led7 dataset has a few attributes, distinct items, and objects. The poker-hand dataset has a few attributes and distinct items, but a large number of objects. Table 5 shows the numbers of rules generated from the datasets in Table 4 for various minimum support thresholds.

Table 4. Characteristics of the experimental datasets.

Dataset	# of Attributes	# of Classes	# of Distinct Items	# of Objects
Breast	12	2	737	699
German	21	2	1077	1000
Lymph	18	4	63	148
Poker-hand	11	10	95	1000000
Led7	8	10	24	3200
Vehicle	19	4	1434	846

Table 5. Numbers of rules generated for various minimum supports.

Dataset	minSup (%)	# of Rules
Breast	1	6016
	0.5	10664
	0.3	15302
	0.1	488356
German	4	63420
	3	116564
	2	265066
	1	1020804
Lymph	4	1177805
	3	1809130
	2	5783910
	1	14253440
Poker-hand	5	1680
	4	1680
	3	1680
	2	1680
Led7	1	5710
	0.5	6430
	0.3	6590
	0.1	7010
Vehicle	0.8	10645
	0.6	15270
	0.4	41930
	0.2	579970

Table 5 shows that a lot of rules were generated for some datasets. For example, the Lymph dataset has more than 14 million rules with *minSup*=1% and the German dataset had more than one million rules with *minSup*=1%. The number of rules generated from the Poker-hand dataset did not change with *minSup* (from 5% to 2%).

Experiments were then conducted to compare the execution time of various interestingness measures. The results for various minimum supports for the 6 datasets are shown in Figures 2 to 7, respectively. The datasets with more numbers of attributes have more rules generated and required a longer execution time.

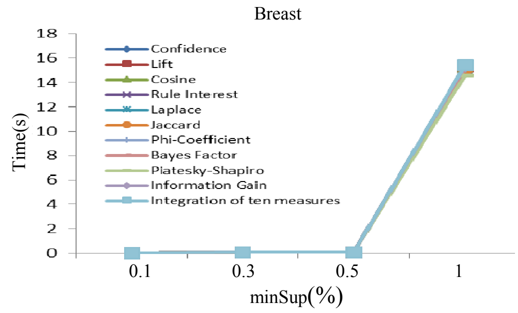

Figure 2. Execution time of ten interestingness measures and integration for the Breast dataset.

The experimental results show that the mining time increases with decrease of the minimum support. The time required for the various interestingness measures varies only slightly. For example, the minimum execution time for the Breast dataset with *minSup*= 0.1% was 15.1516 seconds whereas the maximum was 15.4398 seconds. When the ten measures were integrated together, the mining time was 15.4664 seconds.

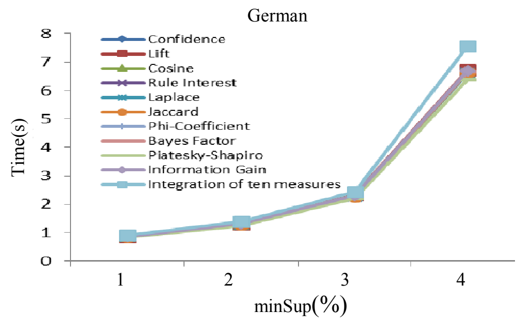

Figure 3. Execution time of ten interestingness measures and integration for the German dataset.

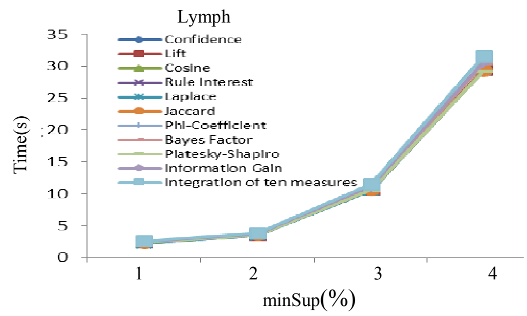

Figure 4. Execution time of ten interestingness measures and integration for the Lymph dataset.

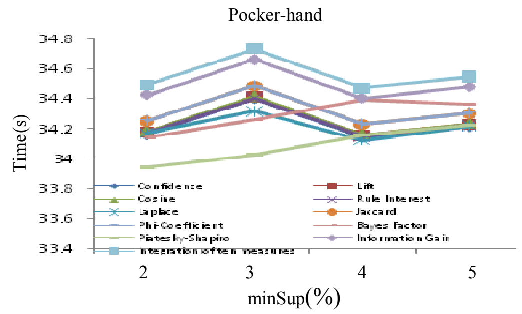

Figure 5. Execution time of ten interestingness measures and integration for the Poker-hand dataset

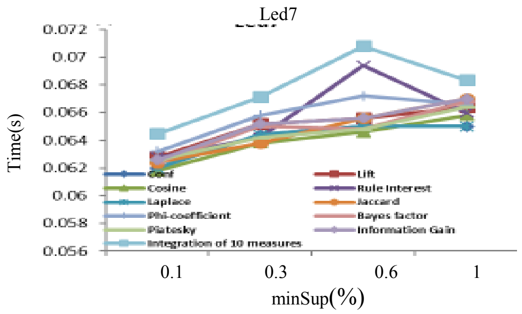

Figure 6. Execution time of ten interestingness measures and integration for the Led7 dataset.

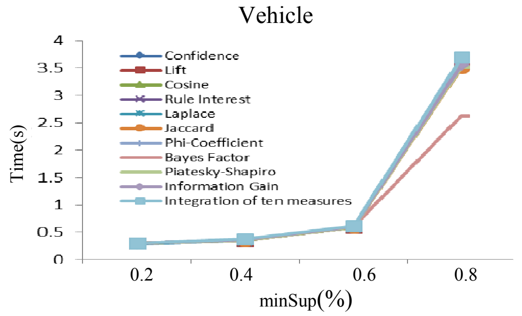

Figure 7. Execution time of ten interestingness measures and integration for the vehicle dataset.

7. Conclusions and Future Work

A new algorithm for mining CAR with interestingness measures was proposed. The algorithm can efficiently mine CARs using the proposed MECR-tree structure. Various interestingness measures can be integrated together for ranking rules. Experimental results show that the execution time for the integration of interestingness measures is only slightly more than that for individual measures.

In the future, the impact of interestingness measures on accuracy will be investigated. Other algorithms for ranking rules and building classifiers will be developed, and methods for effectively integrating interestingness measures for mining CARs with more accuracy will be explored.

References

[1] Agrawal R. and Srikant R., "Fast Algorithms for mining Association Rules," *in Proceedings of the 20th Very Large Data Bases*, Chile, pp. 1-13, 1994.

[2] Aljandal W., Hsu W., Bahirwani V., Caragea D., and Weninger T., "Validation-based Normalization and Selection of Interestingness Measures for Association Rules," *in Proceedings of the 18th International Conference on Artificial Neural Networks in Engineering*, pp. 1- 8, 2008.

[3] Coenen F., Leng P., and Zhang L., "The Effect of Threshold Values on Association Rule based Classification Accuracy," *Data and Knowledge Engineering*, vol. 60, no. 2, pp. 345-360, 2007.

[4] Giuffrida G., Chu W., and Hanssens D., "Mining Classification Rules from Datasets with Large Number of Many-Valued Attributes," *in Proceedings of the 7th International Conference on Extending Database Technology: Advances in Database Technology*, Munich, Germany, pp. 335-349, 2000.

[5] Hilderman R. and Hamilton H., *Knowledge Discovery and Measures of Interest*, Kluwer Academic publishers, 2001.

[6] Hu H. and Li J., "Using Association Rules to Make Rule-based Classifiers Robust," *in Proceedings of the 16th Australasian Database Conference*, Newcastle, Australia, pp. 47-54, 2005.

[7] Huynh H., Guillet F., Blanchard J., Kuntz P., Gras R., and Briand H., "A Graph-based Clustering Approach to Evaluate Interestingness Measures: A Tool and a Comparative Study," *in Proceedings of Quality Measures in Data Mining*, Springer, pp. 25-50, 2007.

[8] Kaya M., "Autonomous Classifiers with Understandable Rule using Multi-Objective Genetic Algorithms," *Expert Systems with Applications*, vol. 37, no. 4, pp. 3489-3494, 2010.

[9] Lee Y., Kim W., Cai Y., and Han J., "CoMine: Efficient Mining of Correlated Patterns," avalible at: http://hanj.cs.illinois.edu/pdf/icdm03_comine.pdf, last visited 2003.

[10] Lenca P., Meyer P., Vaillant P., and Lallich S., "On Selecting Interestingness Measures for Association Rules: User Oriented Description and Multiple Criteria Decision Aid," *European Journal of Operational Research*, vol. 184, no. 2, pp. 610-626, 2008.

[11] Li W., Han J., and Pei J., "CMAR: Accurate and Efficient Classification based on Multiple Class-Association Rules," *in Proceedings of the 1st IEEE International Conference on Data Mining*, California, USA, pp. 369-376, 2001.

[12] Lim A. and Lee C., "Processing Online Analytics with Classification and Association Rule Mining," *Knowledge-Based Systems*, vol. 23, no. 3, 248-255, 2010.

[13] Liu B., Hsu W., and Ma Y., "Integrating Classification and Association Rule Mining," *in Proceedings of the 4th International Conference*

on Knowledge Discovery and Data Mining, New York, USA, pp. 80-86, 1998.

[14] Liu B., Ma Y., and Wong C., "Improving an Association Rule based Classifier," *in Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, Lyon, France, pp. 80-86, 2000.

[15] Liu Y., Jiang Y., Liu X., and Yang S., "CSMC: A Combination Strategy for Multiclass Classification based on Multiple Association Rules," *Knowledge-Based Systems*, vol. 21, no. 8, pp. 786-793, 2008.

[16] Nguyen L., Vo B., Hong T., and Thanh H., "Classification based on Association Rules: A Lattice-based Approach," *Expert Systems with Applications*, vol. 39, no. 13, pp. 11357-11366, 2012.

[17] Omiecinski E., "Alternative Interest Measures for Mining Associations in Databases," *IEEE Transaction on Knowledge and Data Engineering*, vol. 15, no. 1, pp. 57- 69, 2003.

[18] Piatetsky-Shapiro G. and Steingold S., "Measuring Lift Quality in Database Marketing," *SIGKDD Explorations*, vol. 2, no. 2, pp. 76-80, 2000.

[19] Piatetsky-Shapiro G., "Discovery, Analysis, and Presentation of Strong Rules," *Knowledge Discovery in Databases*, pp. 229-248, 1991.

[20] Priss U., "A Classification of Associative and Formal Concepts," avalible at: http://upriss.org.uk/papers/cls02.pdf, last visited 2002.

[21] Qodmanan H., Nasiri M., and Minaei-Bidgoli B., "Multi Objective Association Rule Mining with Genetic Algorithm without Specifying Minimum Support and Minimum Confidence," *Expert Systems with Applications*, vol. 38, no. 1, pp. 288-298, 2011.

[22] Quinlan J., *C4.5: Program for Machine Learning*, Morgan Kaufmann, 1992.

[23] Shekar B. and Natarajan R., "A Transaction-based Neighborhood-Driven Approach to Quantifying Interestingness of Association Rules," *in Proceedings of IEEE International Conference on Data Mining*, pp. 194-201, 2004.

[24] Tan P., Kumar V., and Srivastava J., "Selecting the Right Objective Measure for Association Analysis," *Information Systems*, vol. 29, no. 4, pp. 293-313, 2004.

[25] Thabtah F., Cowling P., and Hammoud S., "Improving Rule Sorting, Predictive Accuracy and Training Time in Associative Classification," *Expert Systems with Applications*, vol. 31, no. 2, pp. 414-426, 2006.

[26] Thabtah F., Cowling P., and Peng Y., "MCAR: Multi-Class Classification based on Association Rule," *in Proceedings of the 3rd ACS/IEEE International Conference on Computer Systems and Applications*, Tunisia, pp. 33-39, 2005.

[27] Thabtah F., Cowling P., and Peng Y., "MMAC: A New Multi-Class, Multi-Label Associative Classification Approach," *in Proceedings of the 4th IEEE International Conference on Data Mining*, Brighton, UK, pp. 217-224, 2004.

[28] Thonangi R. and Pudi V., "ACME: An Associative Classifier based on Maximum Entropy Principle," *in Proceedings of the 16th International Conference Algorithmic Learning Theory*, Singapore, pp. 122-134, 2005.

[29] Tolun M. and Abu-Soud S., "ILA: An Inductive Learning Algorithm for Production Rule Discovery," *Expert Systems with Applications*, vol. 14, no. 3, pp. 361-370, 1998.

[30] Tolun M., Sever H., Uludag M., and Abu-Soud S., "ILA-2: An Inductive Learning Algorithm for Knowledge Discovery," *Cybernetics and Systems*, vol. 30, no. 7, pp. 609-628, 1999.

[31] Veloso A., Meira J., and Zaki M., "Lazy Associative Classification," *in Proceedings of IEEE International Conference on Data Mining*, Hong Kong, China, pp. 645-654, 2006.

[32] Veloso A., Meira J., Goncalves M., Almeida H., and Zaki M., "Calibrated Lazy Associative Classification," *Information Sciences*, vol. 181, no. 13, pp. 2656-2670, 2011.

[33] Vo B. and Le B., "A Novel Classification Algorithm based on Association rule Mining," *in Proceedings of Pacific Rim Knowledge Acquisition Workshop*, Viet Nam, pp. 61-75, 2008.

[34] Vo B. and Le B., "Interestingness Measures for Association Rules: Combination between Lattice and Hash Tables," *Expert Systems with Applications*, vol. 38, no. 9, pp. 1630-11640, 2011.

[35] Yafi E., Al-Hegami A., Alam M., and Biswas R., "YAMI: Incremental Mining of Interesting Association Patterns," *the International Arab Journal of Information Technology*, vol. 9, no. 6, pp. 504-510, 2012.

[36] Yin X. and Han J., "CPAR: Classification based on Predictive Association Rules," *in Proceedings of SIAM International Conference on Data Mining*, San Francisco, USA, pp. 331-335, 2003.

[37] Zhang X., Chen G., and Wei Q., "Building a Highly-Compact and Accurate Associative Classifier," *Applied Intelligence*, vol. 34, no. 1, pp. 74-86, 2011.

[38] Zhao S., Tsang E., Chen D., and Wang X., "Building a Rule-based Classifier - A Fuzzy-Rough Set Approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 5, pp. 624-638, 2010.

Loan Nguyen received her BSc degree in 2002 from University of Science, Vietnam National University of Ho Chi Minh, Vietnam and MSc degree in 2007 from the University of Information Technology, Vietnam National University of Ho Chi Minh, Vietnam, and her PhD degree from Wroclaw University of Technology, Poland. Her research interests include: Association rules, classification, and incremental mining.

Bay Vo received his PhD degrees in computer science from the University of Science, Vietnam National University of Ho Chi Minh, Vietnam in 2011. His research interests include: Association rules, classification, mining in incremental database, distributed databases and privacy preserving in data mining.

Tzung-Pei Hong received his PhD degree in computer science and information engineering from National Chiao-Tung University in 1992. He served as the first director of the library and computer center, the Dean of Academic Affairs and the Vice President in National University of Kaohsiung. Currently, he is a distinguished professor at the Department of Computer Science and Information Engineering in NUK. He has published more than 400 research papers in International/National Journals and Conferences and has planned more than fifty information systems. He is also the board member of more than thirty journals and the program committee member of more than two hundreds conferences. His current research interests include: Parallel processing, machine learning, data mining, soft computing, management information systems, and www applications.