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

Loan Nguyen¹,², Bay Vo³, and Tzung-Pei Hong⁴,⁵
¹Division of Knowledge and System Engineering for ICT, Ton Duc Thang University, Vietnam
²Faculty of Information Technology, Ton Duc Thang University, Vietnam
³Faculty of Information Technology, Ho Chi Minh City University of Technology, Vietnam
⁴Department of CSIE, National University of Kaohsiung, Taiwan
⁵Department 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 generates 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 \(<\text{condset}, y>\), where \(\text{condset}\) 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.
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 $k$-rule items with $k \leq K$. 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 $\chi^2$ 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 the form of $(A_{i1}, a_{i1}), (A_{i2}, a_{i2}), \ldots, (A_{im}, a_{im}) \rightarrow c_{j1} \lor c_{j2} \lor \ldots \lor c_{jk}$, where $a_{ij}$ is a value of attribute $A_{ij}$ and $c_{jk}$ 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 $n$ 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 $A$, $B$, and $C$, and a class attribute named class. Consider the rule $r = \{(A, a_1)\} \rightarrow Y$ 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.

<table>
<thead>
<tr>
<th>OID</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$a_1$</td>
<td>$b_1$</td>
<td>$c_1$</td>
<td>$Y$</td>
</tr>
<tr>
<td>2</td>
<td>$a_1$</td>
<td>$b_1$</td>
<td>$c_1$</td>
<td>$N$</td>
</tr>
<tr>
<td>3</td>
<td>$a_2$</td>
<td>$b_2$</td>
<td>$c_1$</td>
<td>$Y$</td>
</tr>
<tr>
<td>4</td>
<td>$a_3$</td>
<td>$b_1$</td>
<td>$c_1$</td>
<td>$Y$</td>
</tr>
<tr>
<td>5</td>
<td>$a_3$</td>
<td>$b_1$</td>
<td>$c_2$</td>
<td>$N$</td>
</tr>
<tr>
<td>6</td>
<td>$a_3$</td>
<td>$b_1$</td>
<td>$c_1$</td>
<td>$Y$</td>
</tr>
<tr>
<td>7</td>
<td>$a_1$</td>
<td>$b_3$</td>
<td>$c_2$</td>
<td>$Y$</td>
</tr>
<tr>
<td>8</td>
<td>$a_2$</td>
<td>$b_2$</td>
<td>$c_2$</td>
<td>$N$</td>
</tr>
</tbody>
</table>

An association rule can be expressed as $X \rightarrow Y$ or $X \rightarrow Y$, where $X$ and $Y$ are sets, $q = Sup(X \cup Y)$, and $vm$ is a measure value. For example, in traditional association rules, $vm$ is the confidence of the rule, which is evaluated as $Sup(X \cap Y)/Sup(X)$. 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 $X \cap Y$, respectively. The measure $vm$ can thus be computed based on $n, n_x, n_y, n_{xy}$. For example, consider the rule $\{(A, a_1)\} \rightarrow Y$ obtained from Table 1. For this rule, $X=\{(A, a_1)\}$,
$Y^=$, $n^=$ (number of objects), $n^=$, $n^=$, and $n^=$. Some extended parameters can also be calculated as $n^=5$, $n^=4$, and $n^=1$. 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

<table>
<thead>
<tr>
<th>No.</th>
<th>Measure</th>
<th>Equation</th>
<th>Value for the Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Confidence [1]</td>
<td>$\frac{a_{ij}}{n_i}$</td>
<td>2/3</td>
</tr>
<tr>
<td>2</td>
<td>Cosine [24]</td>
<td>$\frac{a_{ij}}{\sqrt{n_i n_j}}$</td>
<td>2/1 $\sqrt{3+4}$</td>
</tr>
<tr>
<td>3</td>
<td>Lift [18]</td>
<td>$\frac{n_i}{n_j}$</td>
<td>$2 \times 8$ $4$, $3 \times 3$</td>
</tr>
<tr>
<td>4</td>
<td>Rule Interest [19]</td>
<td>$\frac{n_i \cdot n_j - n_i \cdot n_j}{n_i - n_j}$</td>
<td>$3 \times 4 - 1 \times 8$ $64$, $1 \times 16$</td>
</tr>
<tr>
<td>5</td>
<td>Laplace [24]</td>
<td>$\frac{o_{ij} + 1}{o_{ij} + 2}$</td>
<td>2/5</td>
</tr>
<tr>
<td>6</td>
<td>Jaccard [24]</td>
<td>$\frac{n_i + n_j - n_{ij}}{n_i + n_j}$</td>
<td>$2 \times 3 + 4 - 2$ $5 - 2$</td>
</tr>
<tr>
<td>7</td>
<td>Phi-Coefficient [24]</td>
<td>$\frac{n_i \cdot n_j - n_i \cdot n_j}{\sqrt{n_i \cdot n_j \cdot n_i \cdot n_j}}$</td>
<td>$2 \times 8 - 3 \times 4$ $\sqrt{3+4} \times 5$ $4$, $1 \times \sqrt{15}$</td>
</tr>
</tbody>
</table>

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: 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 $X^=\{(A, a_2), (B, b_2)\}$ 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 $\{(A, a_2), (B, b_2)\}$ 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 $4B \times a_2b_2$. 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 $4B \times a_2b_2$ can be rewritten as $3 \times a_2b_2$. With this presentation, 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 $l^X$ to node $y$ if the itemset of $X$ is a prefix of the itemset of $Y$. For example, node $\frac{1}{17}a_1l_{(2, 1)}$ is connected to node $3 \times a_1b_1l_{(1, 1)}$, but node $2 \times b_1l_{(3, 1)}$ is not connected to node $3 \times a_1b_1l_{(1, 1)}$.

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 $l_1$ with all the other nodes $l_j$ in $L_2$, with $j \neq i$ (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 $l$attsset 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.
The function \( \text{ENUMERATE\_RULE\_IM}(l) \) 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 \( l.\text{itemset} \rightarrow c_i \). The function then computes the parameter values for this rule, including \( n_X, n_Y \) and \( n_{YX} \) (Lines 18-20), where \( X \) is \( l.\text{itemset} \) and \( Y \) is class \( c_i \). To get the support of \( X \), the cardinality of its Obidset is counted. The support of \( Y \) (\( \text{Count}[i] \)) 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 \( \text{CAR}_l \) (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.

![MECR-tree constructed from the dataset in table 1.](image-url)
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.
<table>
<thead>
<tr>
<th>Dataset</th>
<th># of Attributes</th>
<th># of Classes</th>
<th># of Distinct Items</th>
<th># of Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breast</td>
<td>12</td>
<td>2</td>
<td>737</td>
<td>699</td>
</tr>
<tr>
<td>German</td>
<td>21</td>
<td>2</td>
<td>1077</td>
<td>1000</td>
</tr>
<tr>
<td>Lymph</td>
<td>18</td>
<td>4</td>
<td>63</td>
<td>148</td>
</tr>
<tr>
<td>Poker-hand</td>
<td>11</td>
<td>10</td>
<td>95</td>
<td>1000000</td>
</tr>
<tr>
<td>Led7</td>
<td>8</td>
<td>10</td>
<td>24</td>
<td>3200</td>
</tr>
<tr>
<td>Vehicle</td>
<td>19</td>
<td>4</td>
<td>1434</td>
<td>846</td>
</tr>
</tbody>
</table>

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.
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


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


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.