Selection of Significant Rules in Classification Association Rule Mining

Yanbo J. Wang, Qin Xin, and Frans Coenen

***Abstract*—Classification Rule Mining (CRM) is a Data Mining technique for the extraction of hidden Classification Rules (CRs) from a given database, the objective being to build a classifier to classify "unseen" data. One recent approach to CRM is to use Association Rule Mining (ARM) techniques to identify the desired CRs, i.e. Classification Association Rule Mining (CARM). Although the advantages of accuracy and efficiency offered by CARM have been established in many papers, one major drawback is the large number of Classification Association Rules (CARs) that may be generated (up to a maximum of 2^n-1, where n represents the number of attributes in a database). However, there are only a limited number (k) of CARs that are required to distinguish between classes. The problem addressed in this paper is how to efficiently select all k such CARs. An algorithm is presented, that addresses the above, that operates in binomial time $O(k^2n^2)$; as opposed to exponential time $O(2^n)$ – the time required to find all k CARs in a "one-by-one" manner.**

***Index Terms*—classification association rule, database, data mining, selectors.**

I. Introduction

DATA Mining is a popular area of current research and development in Computer Science that is attracting more and more attention from a wide range of different groups of people. It aims to extract a set of various types of hidden and interesting patterns, rules, regularities and trends from large databases. An Association Rule (AR) is a typical Data Mining pattern that describes co-occurring relationships between database attributes. Association Rule Mining (ARM) [1], a well-established Data Mining technique, aims to identify all ARs in a given database. One application of ARM is to define rules, called Classification Rules (CRs), that will *classify* database records. This kind of AR is called a Classification Association Rule (CAR). The process to build a classifier by employing CRs is called Classification Rule Mining (CRM) [11], which is another well-known Data Mining technique paralleling to ARM.

Classification Association Rule Mining (CARM) [2] is a CRM approach that builds an ARM based classifier using CARs. In [3], Coenen *et al.* suggest that results presented in [9] and [8] show that CARM seems to offer greater accuracy, in many cases, than other methods such as C4.5 [11]. However, one major drawback of this approach is the large number of CARs that may be generated (up to a maximum 2^n-1, where n represents the number of attributes in a database). However, there are only a limited number (k) of "significant" CARs (see Definition 3) that are required to distinguish between classes.

The problem addressed in this paper is how to efficiently select the k "significant rules" among the 2^n-1 possible CARs. We use the acronym SSR-CARM, "Selection of Significant Rules in Classification Association Rule Mining", to describe this problem.

Manuscript received August 10, 2005; revised October 6, 2005; accepted 20 October, 2005.

The authors are with the Department of Computer Science, the University of Liverpool, Chadwick Building, Peach Street, Liverpool, L69 7ZF, UK. E-mail: {jwang, qinxin, frans} @ csc.liv.ac.uk

II. Related Work

ARM is a well-established Data Mining technique, first introduced in [1]. The objective of ARM is to extract ARs, typically defined according to the co-occurrences of binary-valued attributes, from a transaction database (D_T). Let $I = \{a_1, a_2, \ldots, a_n\}$ be a set of items (database attributes), and $T = \{T_1, T_2, \ldots, T_m\}$ be a set of transactions, D_T is described by T, where each $T_i \in T$ contains a set of items I' and $I' \subseteq I$. In ARM, two threshold values are usually used to determine the significance of an AR:

- *Support*: The frequency that the items occur or co-occur in T. A *support threshold* s, defined by the users, is used to distinguish frequent items from the infrequent ones. A set of items S is called an itemset, where $S \subseteq I$, and $\forall a_i \in S$ co-occur in T. If the occurrences of some S in T exceeds s, we say that S is a *frequent itemset.*
- *Confidence*: represents how "strongly" an itemset X implies another itemset Y, where $X, Y \subseteq I$ and $X \cap Y = \{\varnothing\}$. A *confidence threshold* a, supplied by the user, is used to distinguish high confidence ARs from low confidence ARs.

An AR $X \Rightarrow Y$ is *valid* when the *support* for the co-occurrence of X and Y exceeds s, and the *confidence* of this AR exceeds a. The computation of *support* is: $(X \cup Y)$ / (*total number of transactions in* D_T). The computation of *confidence* is: *support*$(X \cup Y)$ / *support*(X). Informally, $X \Rightarrow Y$ can be interpreted as *"if X exists, it is likely that Y also exists"*.

CRM is another well-established Data Mining technique that parallels to ARM. The objective is to discover a set of CRs in a given training database (D_R) from which to build a classifier to *classify* "unseen" data. A D_R consists of n categorical attributes and m records. By convention the last attribute in each record usually indicates its pre-defined class. CRM can thus be described as the process of assigning a Boolean value to each pair $(d_j, c_i) \in D_R \times C$, where each $d_j \in D_R$ is a database record, $C = \{c_1, c_2, \ldots, c_{|C|}\}$ is a set of pre-defined classes, and (d_j, c_i) is a record in D_R being labelled. Well established mechanisms on which CRM algorithms have been based include: Decision Trees [11], Naive Bayes [5], Neural Networks [6], k-Nearest Neighbour [7], Support Vector Machine [10], etc.

An overlap between ARM and CRM is CARM, which strategically solves the traditional CRM problem by applying ARM techniques. Thus a set of CARs is generated from the given transactional training database (D_{TR}). A CAR is an AR of the form $X \Rightarrow c$, where X is a frequent itemset, and c is a pre-defined class to which database records can be assigned. The idea of CARM was first introduced in [2]. CARM algorithms include CBA algorithm [9], CMAR algorithm [8], CPAR algorithm [12], TFPC algorithm [3], etc. These algorithms have been shown to enhance the performance of CRM with respect to accuracy and efficiency.

In this paper, we investigate the SSR-CARM problem and propose an efficient algorithm, based on the concept of selectors [4], to solve the problem in binomial time by avoiding the need to select "significant rules" (in exponential tine) on a one-by-one basis.

III. Selectors

We say that a set P *hits* a set Q on element q, if $P \cap Q = \{q\}$, and a family F of sets hits a set Q on element q, if $P \cap Q = \{q\}$ for at least one $P \in F$. De Bonis *et al.* [4] introduced a definition of a family of subsets of set $[N] \equiv \{0, 1, \ldots, N-1\}$ which hits each subset of $[N]$ of size at most k on at least m distinct elements, where N, k and m are parameters, $N \geq k \geq m \geq 1$. They proved the existence of such a family of size $O((k^2 / (k - m + 1)) \log N)$. For convenience of our presentation, we prefer the following slight modification of this definition, obtained by using the parameter $r = k - m$ instead of the parameter m. For integers N and k, and a real number r such that $N \geq k \geq r \geq 0$, a family F of subsets of $[N]$ is a (N, k, r)-selector, if for any subset $Q \subseteq [N]$ of size at most k, the number of all elements q of Q such that F does not hit Q on q is at most r. That is,

$$|\{q \in Q: \forall P \in \mathcal{F}, P \cap Q \neq \{q\}\}| \leq r$$

In terms of this definition, De Bonis *et al.* [4] showed the existence of a (N, k, r)-selector of size $T(N, k, r) = O((k^2 / (r + 1)) \log N)$. In particular, there exists a $(N, k, 0)$-selector of size $O(k^2 \log N)$ – such a "strong" selector hits each set $Q \subseteq [N]$ of size at most k on each of its elements.

IV. Solving the SSR-CARM Problem

A. Some Definitions

Let $R = \{R_0, R_1, R_2, \ldots, R_{2^n-2}, R_{2^n-1}\}$ be the complete set of possible CARs, and R_i represents a rule in set R with label i.

Definition 1. Let $c^{(A)}(R_i)$ denotes the contribution of $R_i \in R$ to class A, which represents how significant that R_i determines class A, where $0 < c^{(A)}(R_i) < 1$.

Definition 2. If $c^{(A)}(R_i) < \varepsilon$, we recognise $R_i \in R$ as a noisy rule for class A, where ε is a small constant. We use $R^{(A)}{}_N$ to denote the set of noisy rules for class A.

Definition 3. If $\exists R_j \in R$ and $c^{(A)}(R_j)$ satisfies the following inequality,

$$\sum_{i=1}^{2^n} c^{(A)}(R_i \in R^{(A)}{}_N) << c^{(A)}(R_j)$$

We recognise this R_j as a significant rule for class A.

B. The Strategy of the SSR-CARM Algorithm

To solve the SSR-CARM problem, we provide an algorithm that employs a single application of a "strong" $(2^n, k, 0)$-selector. This algorithm ensures that every significant rule in set R will be hit at least once. To apply a family F of subsets of $[2^n]$ means first to arrange the sets of F into a sequence $F_1, F_2, \ldots, F_{|F|}$. Then in the ith step, only CARs in R with labels in F_i will be involved in procedure *SIGNIFICANCE-TEST*, while other CARs can be ignored. Thus, we have an $O(k^2 \log 2^n)$-complexity to hit each of the k significant rules independently at least once due to the property of the "strong" selector. If the current test for F_i contributes to class A significantly, then we call the function *LOG-TEST*, which is based on a binary search and finally finds one particular significant rule from R with labels in F_i.

C. The SSR-CARM Algorithm

The following function identifies a significant rule in R.

Function LOG-TEST(F_i, R);
input: F_i (the ith element in F) and set R;
output: R_w (a significant rule in R);
(1) **begin**
(2) Temp = F_i;
(3) **while** |Temp| > 1 **do**
(4) choose an arbitrary subset Temp_0 with half CARs in Temp to test;
(5) **if** the test significantly contributes to this class
(6) **then** Temp = Temp_0;
(7) **else** Temp = Temp – Temp_0;
(8) **return**(R_w);
(9) **end**

The following procedure solves the SSR-CARM problem, which identifies all significant rules in R.

Procedure SIGNIFICANCE-TEST;
input: F ($(2^n, k, 0)$-selector) and set R;
output: the set SR (the set of significant rules);
(1) **begin**
(2) $SR = 0$;
(3) **for** $i = 1$ to $|F|$ **do**
(4) **if** the label of a CAR R_j in F_i
(5) **then** R_j will be adopted to test with others together;
(6) **else** R_j will be ignored in current test;
(7) **if** the ith test significantly contributes to this class
(8) **then** $SR = SR + \{\text{LOG-TEST}(F_i, R)\}$;
(9) **else** ignore this test;
(10) **end**

Lemma 1. *A $(2^n, k, 0)$-selector has size at most $O(k^2 n)$.*

Proof. It directly comes from the property of the selectors.

Theorem 1. *The SSR-CARM problem can be solved in time $O(k^2 n^2)$.*

Proof. Function *LOG-TEST* takes at most $log2^n$ time to find a significant rule from a subset of R, which contains at least one significant rule. From Lemma 1, we know that a $(2^n, k, 0)$-selector has the size at most $O(k^2 n)$. Consequently, the amount of time spent to solve the SSR-CARM problem can be bounded by $O(k^2 n^2)$.

V. Conclusion

In this paper, we propose an efficient algorithm to solve the SSR-CARM problem in binomial time $O(k^2 n^2)$, which avoids selecting all k significant rules in a one-by-one manner in exponential time $O(2^n)$. If we only wish to seek a fraction of the k significant rules in CARs, the "weak" selector technique [4] can be adopted to solve this problem with time complexity $O(kn^2)$ by substituting the "strong" selector used in our algorithm.

In section 4 we selected all k significant rules from the full set of CARs. However, a combination of two or more different CARs, addressed as a multi-CAR, may also form a significant rule. Therefore, further research of SSR-CARM is directed at multi-CAR significant rules. Another obvious direction of the further research is to investigate other techniques to substitute selectors with a better performance.

Acknowledgment

The authors would like to thank Paul Leng of the Department of Computer Science at the University of Liverpool for his support with respect to the work described here.

References

[1] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Database," *Proc. of the 1993 ACM SIGMOD International Conference on Management of Data*, Washington, D.C., United States, 1993, pp. 207-216.

[2] K. Ali, S. Manganaris, and R. Srikant, "Partial Classification using Association Rules," *Proc. of the 3rd International Conference on Knowledge Discovery and Data Mining*, Newport Beach, California, United States, 1997, pp. 115-118.

[3] F. Coenen, P. Leng, and L. Zhang, "Threshold Tuning for Improved Classification Association Rule Mining," *Proc. of the 9th Pacific–Asia Conference on Knowledge Discovery and Data Mining*, Hanoi, Vietnam, 2005, pp. 216-225.

[4] A. De Bonis, L. Gasieniec, and U. Vaccaro, "Generalized Framework for Selectors with Applications in Optimal Group Testing," *Proc. of the 30th International Colloquium on Automate, Languages and Programming*, Eindhoven, The Netherlands, 2003, pp. 81-96.

[5] P. Domingos, and M. Pazzani, "On the Optimality of the Simple Bayesian Classifier under Zero-one Loss, *Machine Learning*, 29(2/3), 1997, pp. 103-130.

[6] S. J. Hanson, and D. J. Burr, "Minkowski-r Back-propagation: Learning in Connectionist Models with Non-euclidean Error Signals," *Neural Information Processing Systems*, 1987, pp. 348-357, *American Institute of Physics*, New York, United States, 1988.

[7] M. James, *Classification Algorithms*, John Wiley & Sons, New York, New York, 1985.

[8] W. Li, J. Han, and J. Pei, "CMAR: Accurate and Efficient Classification based on Multiple Class-association Rules," *Proc. of the 2001 IEEE International Conference on Data Mining*, San Jose, California, United States, 2001, pp. 369-376.

[9] B. Liu, W. Hsu, and Y. Ma, "Integrating Classification and Association Rule Mining," *Proc. of the 4th International conference on Knowledge Discovery and Data Mining*, New York, New York, United States, 1998, pp. 80-86.

[10] E. Osuna, R. Freund, and F. Girosi, "Support Vector Machines: Training and Applications. *Tech. Report AI Memo No. 1602*, MIT, Cambridge, MA, 1997, pp. 144.

[11] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[12] X. Yin and J. Han, "CPAR: Classification based on Predictive Association Rules." *Proc. SIAM International Conference on Data Mining*, San Francisco, CA, 2003, pp. 331-335.