Mining a Complete Set of Both Positive and Negative Association Rules from Large Databases

Hao Wang, Xing Zhang, and Guoqing Chen

Department of Management Science and Engineering, Tsinghua University,
Beijing 100084, China
w-hao02@mails.tsinghua.edu.cn

Abstract. Association rule mining is one of the key issues in knowledge discovery. In recent years, negative association rule mining has attracted remarkable attention. This paper presents a notion of validity for both positive and negative association rules, which is considered intuitive and necessary. Then, a mining algorithm to find all rules in light of completeness is proposed. In doing so, several pruning strategies based on the upward closure property are developed and incorporated into the algorithm so as to guarantee the computational efficiency.

Keywords: Negative association rules, upward closure, Apriori, data mining.

1 Introduction

As one of the promising areas of research for knowledge discovery, association rule mining (ARM) attempts at finding the relationships between the different items in databases [1-3]. Researchers have extended the association rule (AR) concept — originally specific to binary data tables— to a multitude of domains, involving quantitative, hierarchical, fuzzy, and many other kinds of databases. The main characteristic of the efforts is to predict the presence of some data items (itemsets) from the presence of other data items. In other words, the focal point of interest is the positive association of itemsets, namely, a presence-to-presence relationship. On the other hand, in many real applications, negative associations (i.e., the relationship between the presence and the absence of itemsets, or the absence and the absence of itemsets) are meaningful and therefore attracting more and more attention nowadays (e.g., [4-13]). For example, "office workers who did NOT buy cars turned to rent homes near subway stations", and "customers who were NOT interested in big screen mobile phones would NOT buy other value-added services (e.g., games, web connections, etc.)". These kinds of ARs reflect certain negative patterns of data items and are usually referred to as negative ARs.

Mining negative ARs, however, raises a number of critical issues [13]. First, the density of data in databases becomes much higher. Second, the computational cost may skyrocket when each item in the database and its corresponding negated item (indicating absence of the original item) are considered, since the mining complexity may increase significantly in terms of the number of data items. Moreover, negative

T. Washio et al. (Eds.): PAKDD 2008, LNAI 5012, pp. 777–784, 2008.
© Springer-Verlag Berlin Heidelberg 2008

ARs may invalidate some important pruning strategies used to restrict the search space and guarantee efficiency in classical ARM algorithms.

In order to address these issues and explore efficient algorithms, a number of efforts have been made to develop improvements and extensions. Savasere et al. [5] and Yuan et al. [8] incorporate domain knowledge (taxonomy structure) into the algorithms. These approaches compare expected support or confidence of an itemset (based on itemsets' positions in the taxonomy structure) with the actual value of these measures. The limitations of these approaches are: first, negative ARs are mainly restricted to relative negative ARs compared with other sibling itemsets; and second, the domain knowledge (taxonomy structure) needed may often not be readily available. Wu et al. [12] and Antonie et al. [9] focus on certain notions of negative ARs (such as $\neg X \Rightarrow Y$, $X \Rightarrow \neg Y$, $\neg X \Rightarrow \neg Y$) and present approaches to mine them. However, it is found that their approaches can hardly guarantee to generate a complete set of valid rules, that is, some valid negative ARs defined may not be obtained using their algorithms [13]. Furthermore, Brin et al. [4], and Cornelis et al. [13] concentrate on similar notions of negative ARs and provide algorithms to generate all rules of concern. However, their notions are, though meaningful, restrictive in semantics and deemed necessary to extend.

In this paper, another notion of validity for both positive and negative ARs is presented, which reflects semantics in a broader sense and appears to be intuitive. Then, a mining algorithm is proposed, which is sound and complete in terms of generating all rules of interest. Pruning strategies based on the upward closure property are developed and incorporated into the algorithm so as to guarantee the computational efficiency.

2 Valid Association Rules

In association rule mining, two measures, namely the Degree of Support (*supp*) and the Degree of Confidence (*conf*), are used to define a rule [2-3]. According to [9, 12-13], In the case of the negation of a set of items (itemset) X, denoted by $\neg X$, the degree of support is: $supp(\neg X) = 1 - supp(X)$. A rule of the form $X \Rightarrow Y$ is called a positive rule, whereas rules of the other forms ($\neg X \Rightarrow Y$, $X \Rightarrow \neg Y$, $\neg X \Rightarrow \neg Y$) are negative rules. Specifically, the degrees of support and the degrees of confident are defined as follows:

$$supp(X \Rightarrow Y) = supp(X \cup Y)$$
$$supp(\neg X \Rightarrow Y) = supp(Y) - supp(X \cup Y)$$
$$supp(X \Rightarrow \neg Y) = supp(X) - supp(X \cup Y)$$
$$supp(\neg X \Rightarrow \neg Y) = 1 - supp(X) - supp(Y) + supp(X \cup Y)$$
$$conf(C_1 \Rightarrow C_2) = \frac{supp(C_1 \Rightarrow C_2)}{supp(C_1)}$$

where $X \cap Y = \varnothing, C_1 \in \{X, \neg X\}, C_2 \in \{Y, \neg Y\}$. Subsequently, we have definition 1 for valid association rules.

Definition 1 (valid association rule)**.** Let X and Y be itemsets. A valid association rule (AR) is an expression $C_1 \Rightarrow C_2, C_1 \in \{X, \neg X\}, C_2 \in \{Y, \neg Y\}, X \cap Y = \varnothing$, such that $posbound(C_1 \Rightarrow C_2) = 1$ and $negbound(C_1 \Rightarrow C_2) = 1$, where *posbound* and *negbound* are mappings from the set of possible ARs to {0, 1} with:

$$posbound(C_1 \Rightarrow C_2) = \begin{cases} 0 \ if \ \ Supp(C_1 \cup C_2) < ms \text{ or } Supp(X) < ms \\ \qquad \text{or } Supp(Y) < ms \text{ or } conf(C_1 \Rightarrow C_2) < mc \\ 1 \ otherwise \end{cases}$$

$$negbound(C_1 \Rightarrow C_2) = \begin{cases} 0 \ \ if \ \ C_2 = \neg Y \text{ and } \exists\, Y' \subset Y, s.t.\ posbound(C_1 \Rightarrow \neg Y') = 1 \\ 1 \ \ otherwise \end{cases}$$

It is worth mentioning that, according to the definition, if $C_1 \Rightarrow \neg Y$ is valid, then there should not exist $Y' \subset Y$ such that $C_1 \Rightarrow \neg Y'$ is also valid. This is based on the fact that, if $posbound(C_1 \Rightarrow \neg Y') = 1$, then $posbound(C_1 \Rightarrow \neg Y) = 1$ is always true, for all $Y' \subset Y$. Moreover, it is important to note that several useful properties hold as follows, which can be used in pruning strategies for efficiency and to guarantee the completeness of the proposed AR mining algorithm that will be discussed in the next section.

Property 1

(1.1) $supp(X) \geq supp(X')$, for all $X \subseteq X'$ (downward closure)

(1.2) $supp(\neg X') \geq supp(\neg X)$, for all $X \subseteq X'$ (upward closure)

Property 2

(2.1) $supp(C_1 \Rightarrow Y) \geq supp(C_1 \Rightarrow Y')$, for all $Y \subseteq Y'$.

(2.2) $supp(\neg X' \Rightarrow C_2) \geq supp(\neg X \Rightarrow C_2)$, for all $X \subseteq X'$.

(2.3) $supp(C_1 \Rightarrow \neg Y') \geq supp(C_1 \Rightarrow \neg Y)$, for all $Y \subseteq Y'$.

Property 3

(3.1) $conf(C_1 \Rightarrow Y) \geq conf(C_1 \Rightarrow Y')$, for all $Y \subseteq Y'$.

(3.2) $conf(C_1 \Rightarrow \neg Y') \geq conf(C_1 \Rightarrow \neg Y)$, for all $Y \subseteq Y'$.

Property 4

Let $\overline{conf}(\neg X \Rightarrow Y) = \dfrac{supp(Y)}{1 - supp(X)}$ then,

(4.1) $\overline{conf}(\neg X \Rightarrow Y) \geq conf(\neg X \Rightarrow Y)$.

(4.2) $\overline{conf}(\neg X \Rightarrow Y) \geq \overline{conf}(\neg X \Rightarrow Y')$, for $Y \subseteq Y'$.

3 Generating Valid Association Rules

As can be seen in previous discussions, all valid association rules of concern, namely the AR space, are composed of four types of ARs. In other words, the AR space could

be partitioned into four parts: Part I: positive valid ARs, in forms of $X \Rightarrow Y$; Part II: negative valid ARs, in forms of $\neg X \Rightarrow Y$; Part III: negative valid ARs, in forms of $X \Rightarrow \neg Y$; and Part IV: negative valid ARs, in forms of $\neg X \Rightarrow \neg Y$.

The mining process, therefore, constitutes four major steps to generate all frequent itemsets and all valid ARs in part I, all negative valid ARs in part II, III and IV respectively. For part I, all frequent itemsets and valid ARs could be generated using the Apriori-type approaches etc. While the effective Apriori's pruning strategy based on downward closure property (Property 1.1) still pertains in generating part I, it does not suit in generating parts II, III, and IV. Hence, new pruning strategies, say, based on upward closure property (Property 1.2) and other properties mentioned in section 2, need to be developed so as to enable an effective and efficient generation of all negative valid ARs.

3.1 Pruning Strategies

In addition to property 1.1 used as a pruning strategy to generate positive valid ARs, other pruning strategies are needed in generating negative valid ARs. According to properties 1, 2 and 3, an important property (Property 5) can be derived, which is downward-closure-like and could be incorporated in the mining process for part II as a pruning strategy. Furthermore, in discovering valid ARs in forms of $C_1 \Rightarrow \neg Y$ (i.e., valid ARs in parts III and IV), another proven property (Property 6) is important as well.

Property 5. If $\neg X \Rightarrow Y'$ is valid, then $\neg X \Rightarrow Y$ ($Y \subset Y'$) is valid.

Property 6. If $C_1 \Rightarrow \neg Y$ is valid, then $C_1 \Rightarrow \neg Y'$ ($Y' \subset Y$) is not valid.

Property 5 enables us to generate valid ARs of the form $\neg X \Rightarrow Y$ by extending the consequents of already obtained valid ARs and prune candidate ARs by examining their consequents' (k-1)-length sub-itemsets in valid ARs (k is the length of their consequents). Property 6 enables us to use potential ARs (not valid ARs, but having potential to generate valid rules by extending their consequents) to generate valid ARs by extending the consequents of them and prune candidate ARs by examining their consequents' (k-1)-length sub-itemsets in potential ARs.

Notably, the difference between the Apriori-type approach and the proposed approach for parts II, III and IV is that the former uses frequent itemsets to generate and prune candidate itemsets, whereas the latter uses valid ARs to generate and prune candidate ARs for part II, and uses potential ARs for parts III and IV. More details are presented in the following subsection.

3.2 Algorithmic Details

The following notations are used in discussing algorithmic ideas.

X, Y: positive itemsets;

$|X|, |Y|$: the number of items in X, Y

$L(P_{\mathrm{I}})$: frequent itemsets in part I;

$L(P_{\mathrm{I}})_k$: k-length frequent itemsets in part I;

$VR(P_i)$: valid ARs in part i;

$VR(P_i)_{k,p}$: valid ARs with k-length antecedent and p-length consequent in part i;

$NR(P_i)$: potential ARs; not valid ARs, but having potential to generate valid rules by extending its consequent in part i. It is used to generate $CR(P_i)$ for parts III and IV.

$CR(P_i)$: candidate ARs in part i, including $VR(P_i)$ and $NR(P_i)$.

$S(P_i)$: positive itemsets whose support needs to be calculated via DB scan in part i; (analogously: $NR(P_i)_{k,p}$, $CR(P_i)_{k,p}$, $S(P_i)_{k,p}$).

Procedure 1. Generate all Negative Valid ARs in Parts II, III and IV (with minimal confidence mc)

1: $VR(P_i) = \varnothing$

2: $$CR(P_i)_{1,1} = \begin{cases} \{\neg X \Rightarrow Y \mid X, Y \in L(P_{\mathrm{I}})_1, X \cap Y = \varnothing, \overline{conf}(\neg X \Rightarrow Y) \geq mc\} & \text{for } i\text{=II} \\ \{X \Rightarrow \neg Y \mid X, Y \in L(P_{\mathrm{I}})_1, X \cap Y = \varnothing\} & \text{for } i\text{=III} \\ \{\neg X \Rightarrow \neg Y \mid X, Y \in L(P_{\mathrm{I}})_1, X \cap Y = \varnothing\} & \text{for } i\text{=IV} \end{cases}$$

3: **for** $\{k = 1; CR(P_i)_{k,1} \neq \varnothing; k{+}{+}\}$ **do**

4: **for** $\{p = 1; CR(P_i)_{k,p} \neq \varnothing; p{+}{+}\}$ **do**

5: generate $S(P_i)_{k,p}$

6: compute support of all itemsets in $S(P_i)_{k,p}$

7: generate $VR(P_i)_{k,p}$ and $NR(P_i)_{k,p}$

8: $VR(P_i) = VR(P_i) \cup VR(P_i)_{k,p}$

9: generate $CR(P_i)_{k,p+1}$

10: delete $NR(P_i)_{k,p}$ for $i \neq$ II

11: **end for**

12:

$$CR(P_i)_{k+1,1} = \begin{cases} \{\neg X \Rightarrow Y \mid X \in L(P_{\mathrm{I}})_{k+1}, Y \in L(P_{\mathrm{I}})_1, X \cap Y = \varnothing, \overline{conf}(\neg X \Rightarrow Y) \geq mc\} & \text{for } i\text{=II} \\ \{X \Rightarrow \neg Y \mid X \in L(P_{\mathrm{I}})_{k+1}, Y \in L(P_{\mathrm{I}})_1, X \cap Y = \varnothing\} & \text{for } i\text{=III} \\ \{\neg X \Rightarrow \neg Y \mid X \in L(P_{\mathrm{I}})_{k+1}, Y \in L(P_{\mathrm{I}})_1, X \cap Y = \varnothing\} & \text{for } i\text{=IV} \end{cases}$$

13: **end for**

Procedure 1 first generates candidate, valid and potential ARs with k-length antecedents and 1-lengh consequents, then generates candidate, valid and potential ARs with k-length antecedents and p+1-lengh consequents from valid ARs (for part II) or from potential ARs (for parts III and IV) with k-length antecedents and p-lengh

consequents. More concretely, let us consider certain lines of algorithmic treatments in Procedure 1 as follows, whereas corresponding (sub-) procedural codes are omitted due to the limitation of space.

For line 5, $S(P_i)_{k,p}$ is positive itemsets, the support of any element in it is unknown. It is needed in the computation of *pospound* for generating $VR(P_i)_{k,p}$ and $NR(P_i)_{k,p}$. Line 6 computes itemsets in $S(P_i)_{k,p}$ via database scan. For line 9, $CR(P_i)_{k,p+1}$ from $VR(P_i)_{k,p}$ for part II and from $NR(P_i)_{k,p}$ for parts III and IV are generated, in that pruning strategies discussed in subsection 3.1 are used for $CR(P_i)_{k,p+1}$.

Note that the generation of $CR(P_i)_{k,p+1}$ is only related to $VR(P_i)_{k,p}$ (for part II) or $NR(P_i)_{k,p}$ (for parts III and IV), the generation of $CR(P_i)_{k,p+1}$ can start after the generation of $VR(P_i)_{k,p}$ (for part II) or $NR(P_i)_{k,p}$ (for parts III and IV), and do not have to wait generations of other ARs with $\hat{k}$-length antecedents ($\hat{k} \neq k$). This is a very good feature that parallel computing may be possible, where dynamically specified cores (or processors) could be executed for Procedure 1 from lines 4 to 12 with certain parallel computing algorithms.

Importantly, it can be proven that the above-mentioned algorithm will generate a complete set of all positive and negative valid ARs. That is, the proposed approach (and the corresponding algorithm) is both sound and complete. It is also easy to show that the proposed approach in this paper is considered advantageous over existing ones (e.g., [4-5], [8-9], [12-13]) in terms of meaningfulness in rule validity and completeness in rule generation.

4 Experiment Results

To study the effectiveness of our approach, we have performed data experiments based on synthetic databases generated by IBM Synthetic Data Generator for Associations and Sequential Patterns (http://www.cse.cuhk.edu.hk/~kdd/data_collection.html). In the experiments, we used C++ on a Lenovo PC with 3G of CPU and 4GB memory. The main parameters of the databases are as follows. The total number of attributes is 1000; the average number of attributes per row is 10; the number of rows is 98358, approximately 100K; the average size of maximal frequent sets is 4.

The experiments were to illustrate that, though data-dependent, there are much more negative ARs than positive ones due to the nature of negation of data items semantically, and that the proposed approach (namely Algorithm VAR) is effective in generating negative ARs in terms of throughput rate (number of rules per time unit), which is higher than that of Apriori algorithm [3] (namely Apriori) for generating positive ARs. Table 1 shows the results.

Table 1. Running time (seconds) and numbers of positive and negative ARs

		Positive rules (generated by Apriori)			Negative rules (generated by VAR)		
Ms	*Mc*	Time	Number	Number /Time	Time	Number	Number /Time
0.001	0.6	41	91045	2221	1868	17345516	9286
0.001	0.7	40	86004	2150	1863	17346435	9311
0.001	0.8	40	68654	1716	1890	17357488	9184
0.001	0.9	40	37438	936	1913	17367665	9079
0.015	0.6	24	6211	259	307	5573850	18156
0.015	0.7	24	5947	248	309	5572996	18036
0.015	0.8	24	5488	229	309	5569366	18024
0.015	0.9	24	4092	171	313	5554373	17746

Moreover the advantage became larger with the increase in minimal support *ms*. The fact that Number/Time in VAR decreased with the increase in minimal confidence *mc* is because larger *mc* made the negative ARs' negative consequents to become longer to satisfy it. For example, if $supp(C_1 \Rightarrow \neg Y) \geq ms$ and $conf(C_1 \Rightarrow \neg Y) < mc$, in our algorithm, we may generate some Y', $Y \subseteq Y'$ such that $supp(C_1 \Rightarrow \neg Y') \geq ms$ and $conf(C_1 \Rightarrow \neg Y') \geq mc$. This process costs a litter more time when *mc* increases, however, the minimal Number/Time of VAR is still advantageous over the maximal of Apriori.

5 Conclusion

Negative association rules are considered useful in many real world applications. This paper has proposed a notion of valid association rules and developed an effective approach, along with the corresponding algorithm, to mining all positive and negative ones in a sound and complete manner. Several rule properties have been investigated and incorporated into the mining process as pruning strategies in order to gain algorithmic efficiency. The main advantage of the proposed approach over others could be characterized in terms of meaningfulness in rule validity and completeness in rule generation.

Acknowledgements

The work was partly supported by the National Natural Science Foundation of China (70621061/70231010) and Research Center for Contemporary Management at Tsinghua University.

References

1. Piatetsky-Shapiro, G.: Discovery, Analysis and Presentation of Strong Rules. In: Knowledge Discovery in Databases, pp. 229–248. AAAI/MIT, Menlo Park (1991)
2. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Databases. In: Proc. ACM-SIGMOD Intl. Conf. on Management of Data, pp. 207–216 (1993)

3. Srikant, R., Agrawal, R.: Fast Algorithms for Mining Association Rules. In: Proc. VLDB Conference, pp. 487–499 (1994)
4. Brin, S., Motwani, R., Silverstein, C.: Beyond Market Baskets: Generalizing Association Rules to Correlations. In: Proc. ACM SIGMOD on Management of Data, pp. 265–276 (1997)
5. Savasere, A., Omiecinski, E., Navathe, S.: Mining for Strong Negative Associations in a Large Database of Customer Transactions. In: Proc. Intl. Conf. on Data Engineering, pp. 494–502 (1998)
6. Aggarwal, C.C., Yu, P.S.: A New Framework for Itemset Generation. In: Proc. ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, pp. 18–24 (1998)
7. Wei, Q., Chen, G.: Association Rules with Opposite Items in Large Categorical Database. In: Proc. Intl. Conf. on Flexible Query Answering Systems, pp. 507–514 (2000)
8. Yuan, X., Buckles, B.P., Yuan, Z., Zhang, J.: Mining Negative Association Rules. In: Proc. Seventh Intl. Symposium on Computers and Communication, Italy, pp. 623–629 (2002)
9. Antonie, M.L., Zaïane, O.R.: Mining Positive and Negative Association Rules: an Approach for Confined Rules. In: Proc. Intl. Conf. on Principles and Practice of Knowledge Discovery in Databases, pp. 27–38 (2004)
10. Daly, O., Taniar, D.: Exception Rules Mining Based On Negative Association Rules. In: Laganá, A., Gavrilova, M.L., Kumar, V., Mun, Y., Tan, C.J.K., Gervasi, O. (eds.) ICCSA 2004. LNCS, vol. 3046, pp. 543–552. Springer, Heidelberg (2004)
11. Thiruvady, D.R., Webb, G.I.: Mining Negative Association Rules Using GRD. In: Proc. Pacific-Asia Conf. on Advances in Knowledge Discovery and Data Mining, pp. 161–165 (2004)
12. Wu, X., Zhang, C., Zhang, S.: Efficient Mining of Both Positive and Negative Association Rules. ACM Trans. on Information Systems 22(3), 381–405 (2004)
13. Cornelis, C., Yan, P., Zhang, X., Chen, G.: Mining Positive and Negative Association Rules from Large Databases. In: Wang, Y., Cheung, Y.-m., Liu, H. (eds.) CIS 2006. LNCS (LNAI), vol. 4456, Springer, Heidelberg (2007)