Efficient Mining of High Confidence Association Rules without Support Thresholds

Jinyan Li[1], Xiuzhen Zhang[1],
Guozhu Dong[2], Kotagiri Ramamohanarao[1], and Qun Sun[1]

[1] Department of CSSE, The University of Melbourne, Parkville, Vic. 3052, Australia.
{jyli, xzhang, rao, qun}@cs.mu.oz.au
[2] Dept. of CSE, Wright State University, Dayton OH 45435, USA. gdong@cs.wright.edu

Abstract. Association rules describe the degree of dependence between items in transactional datasets by their confidences. In this paper, we first introduce the problem of mining *top rules*, namely those association rules with 100% confidence. Traditional approaches to this problem need a minimum support (*minsup*) threshold and then can discover the top rules with supports $\geq$ *minsup*; such approaches, however, rely on *minsup* to help avoid examining too many candidates and they miss those top rules whose supports are below *minsup*. The low support top rules (e.g. some unusual combinations of some factors that have always caused some disease) may be very interesting. Fundamentally different from previous work, our proposed method uses a dataset partitioning technique and two border-based algorithms to efficiently discover all top rules with a given consequent, without the constraint of support threshold. Importantly, we use borders to concisely represent all top rules, instead of enumerating them individually. We also discuss how to discover all zero-confidence rules and some very high (say 90%) confidence rules using approaches similar to mining top rules. Experimental results using the Mushroom, the Cleveland heart disease, and the Boston housing datasets are reported to evaluate the efficiency of the proposed approach.

1 Introduction

Association rules [1] were proposed to capture significant dependence between items in transactional datasets. For example, the association rule $\{tea,\ coffee\} \rightarrow \{sugar\}$ says it is highly likely that a customer purchasing *tea* and *coffee* also purchases *sugar*; its likelihood is measured by its confidence (the percentage of transactions containing *tea* and *coffee* which also contain *sugar*). In this work, we are mainly interested in the efficient mining of association rules with 100% confidence, which we call the *top rules*. Observe that if $X_1 \rightarrow X_2$ is a top rule, then any transaction containing X_1 *must* also contain X_2. The following example shows the usefulness of top rules.

Example 1. For the Cleveland heart disease dataset (taken from UCI ML repository), we have found many top rules. Two typical top rules are: {*having ST-T wave abnormality, exercise induced angina*} $\rightarrow Presence$ and {*left ventricular hypertrophy, downsloping of the peak exercise ST segment, thal: fixed defect*} $\rightarrow \{CP = 1, fbs = 1\}$. The first rule means that if the two symptoms on the left-hand side of the rule appear then the patients definitely suffer from a heart disease of some degree (either slight or serious). The second rule means that no matter male or female and no matter suffering from heart

J.M. Żytkow and J. Rauch (Eds.): PKDD'99, LNAI 1704, pp. 406–411, 1999.

disease or not, once holding the left-hand side symptoms the patients must suffer from a *typical angina* and their *fasting blood sugar > 120 mg/dl*. Knowing all rules of this type can be of great help to medical practitioners.

Traditional approaches to discovering association rules usually need two steps: (i) find all itemsets whose supports are $\geq minsup$; (ii) from the result of step (i), find all association rules whose confidences are $\geq minconf$ and whose support are $\geq minsup$. Observe, however, using this procedure, those top rules with supports less than $minsup$ are missed. Although the method proposed in [2] can effectively extract high confidence rules, a *minsup* threshold is still imposed on the discovered rules. Another disadvantage of these approaches is that they need to explicitly enumerate all discovered rules and to explicitly check all candidate rules; this would require a long processing time and I/O time if the number of top rules or the number of candidates is huge.

Fundamentally different from previous work, we propose in this paper an emerging pattern (EP) [4] based approach to efficiently discover all top rules, without *minsup* limitation, when given the *consequent* of the expected top rules. Given a dataset $\mathcal{D}$, the proposed approach first divides $\mathcal{D}$ into two sub-datasets according to the consequent of the expected top rules and then uses the border-based algorithms to mine a special kind of itemsets whose supports in one sub-dataset are zero but non-zero in the other sub-dataset. This special kind of itemsets are called *jumping* EPs [5]. All desired top rules can then be readily built using jumping EPs. As the border-based algorithms are very efficient, the proposed approach is also efficient. Furthermore, we do not enumerate all top rules; we use borders [4] to succinctly represent them instead. The significance of this representation is highlighted in the experimental results of Mushroom dataset, where there exist a huge number of top rules. In addition to top rules, we also address the problems of mining zero-confidence rules and mining very high (say $\geq 90\%$) confidence rules with similar approaches to mining top rules.

Organization: §2 formally introduces the problem of mining top rules. §3 discusses our approach to this mining problem. §4 discusses how to discover high confidence rules (and zero-confidence rules). The experimental results are shown in §5 to evaluate the performance of our approaches. §6 discusses how to use the border mechanism to efficiently find top rules with support or length constraints. §7 concludes this paper.

2 Problem Definition

Given a set $I = \{i_1, i_2, \cdots, i_N\}$ of *items*, a *transaction* is a subset T of I, and a *dataset* is a set $\mathcal{D}$ of transactions. The *support* of an itemset X in a dataset $\mathcal{D}$, denoted as $supp_{\mathcal{D}}(X)$, is $\frac{count_{\mathcal{D}}(X)}{|\mathcal{D}|}$, where $count_{\mathcal{D}}(X)$ is the number of transactions in $\mathcal{D}$ containing X. An *association rule* is an implication of the form $X \to Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$; X is called the left-hand side (or antecedent) of the rule and Y the right-hand side (or consequent). A rule $X \to Y$ is associated with two measures: its *support* which is defined as $supp_{\mathcal{D}}(XY)$ and its *confidence* which is defined as $\frac{count_{\mathcal{D}}(XY)}{count_{\mathcal{D}}(X)}$ (equivalently, $\frac{supp_{\mathcal{D}}(XY)}{supp_{\mathcal{D}}(X)}$).

Definition 1. *A* top rule *is an association rule whose confidence is exactly* 100%.

In many applications, for example when examining the edibility of mushroom or making decisions on heart disease, the goal is to discover properties of a specific class of interest. We capture such situations by requiring the mined rules to have a given TARGET as consequent [1] and intend to find all such rules.

Definition 2. *Given a set of transactions $\mathcal{D}$ and an itemset* TARGET*, the problem of* mining top rules *is to discover all top rules with* TARGET *as the consequent.*

In Section 4 we also consider discovering zero-confidence rules and μ-level confidence rules based on the approach to mining top rules.

3 Discovering Top Rules

We solve the problem of mining top rules in two steps:

1. **Dataset Partitioning and Transformation**. Divide $\mathcal{D}$ into sub-datasets $\mathcal{D}_1$ and $\mathcal{D}_2$: $\mathcal{D}_1$ consists of the transactions containing TARGET; $\mathcal{D}_2$ consists of the transactions which do not contain TARGET. Then, we remove all items of TARGET from $\mathcal{D}_1$ and $\mathcal{D}_2$. As a result, $\mathcal{D}_1$ and $\mathcal{D}_2$ are transformed into $\mathcal{D}'_1$ and $\mathcal{D}'_2$ respectively and the set of items becomes $I' = I - Target$. All these can be done in one pass over $\mathcal{D}$.
2. **Discovery of Top Rules**. Find all those itemsets X which occur in $\mathcal{D}'_1$ but do not occur in $\mathcal{D}'_2$ at all. Then, for every such X, the rule $X \rightarrow Target$ is a top rule (with confidence of 100%) in $\mathcal{D}$.

Correctness. By definition, the confidence of the rule $X \rightarrow Target$ is the percentage of the transactions in $\mathcal{D}$ containing X which also contain TARGET. As the discovered itemsets X only occur in $\mathcal{D}'_1$ but not in $\mathcal{D}'_2$, those transactions in $\mathcal{D}$ which contain X must also contain TARGET. This means that $X \rightarrow Target$ has confidence of 100% in $\mathcal{D}$. In contrast, if some itemset Y occurs in both $\mathcal{D}'_1$ and $\mathcal{D}'_2$, then it is not true that those transactions in $\mathcal{D}$ which contain Y must also contain TARGET, by the constructions of $\mathcal{D}'_1$ and $\mathcal{D}'_2$.

We call those itemsets X, which only occur in $\mathcal{D}'_1$ but not in $\mathcal{D}'_2$, the *jumping emerging patterns* (jumping EPs) from $\mathcal{D}'_2$ to $\mathcal{D}'_1$ [5]. Observe that their supports in $\mathcal{D}'_1$ are non-zero but zero in $\mathcal{D}'_2$.

It is now obvious that the key step in mining top rules in $\mathcal{D}$ is to discover the jumping EPs between two relevant datasets because the discovered jumping EPs are the antecedents of top rules.

In the work of [5], the problem of mining jumping EPs is formally defined and well solved. The high efficiency of those algorithms is a consequence of their novel use of borders [4] - an efficient representation mechanism.

For the problem of mining jumping EPs, two border-based algorithms have been proposed in [5]. The first one is called HORIZON-MINER and the second is based on MBD-LLBORDER of [4]. HORIZON-MINER is used to discover a special border, called *horizontal border* in [5], from a dataset such that this special border represents precisely

[1] As an aside, we note that it is very easy to generate such top rules as $Target \rightarrow X$. First, create a new dataset which consists of all those transactions in $\mathcal{D}$ containing TARGET. Second, remove all items of TARGET from this new dataset to form $\mathcal{D}'$. Then, all itemsets X with 100% support in $\mathcal{D}'$ can be used to build the top rules $Target \rightarrow X$.

all itemsets with non-zero supports in this dataset. When taking two horizontal borders derived from $\mathcal{D}_1'$ and $\mathcal{D}_2'$ by HORIZON-MINER as input, the MBD-LLBORDER algorithm can produce all itemsets whose supports in $\mathcal{D}_1'$ are non-zero but zero in $\mathcal{D}_2'$. Therefore, this output of MBD-LLBORDER is just our desired jumping EPs from $\mathcal{D}_2'$ to $\mathcal{D}_1'$. More details of HORIZON-MINER and MBD-LLBORDERcan be found in [4,5].

4 Discovering Zero-Confidence Rules and μ-Level Confidence Rules

As shown above, a jumping EP X from $\mathcal{D}_2'$ to $\mathcal{D}_1'$ corresponds to a top rule $X \rightarrow Target$ and vice versa in $\mathcal{D}$. On the other hand, any jumping EP Z from $\mathcal{D}_1'$ to $\mathcal{D}_2'$ corresponds to the rule $Z \rightarrow Target$ with *zero-confidence* in $\mathcal{D}$. This is because itemset Z only occurs in $\mathcal{D}_2'$, and so all transactions in $\mathcal{D}$ which contain Z must not contain TARGET. Observe that zero-confidence rules reveal absolutely (100%) *negative correlation* between two events. Procedurally, there is only slight difference between the problems of mining top rules and mining zero-confidence rules.

In this paper, we are also interested in the problem of mining μ-level confidence rules: We refer μ-level confidence rules as those association rules whose confidences are $\geq 1-\mu$. We will show that the parameter μ strongly depends on the ratio of two supports in $\mathcal{D}_1'$ and $\mathcal{D}_2'$ of itemsets Y. Let $supp_i(Y)$ denote the support of Y in $\mathcal{D}_i'$, $i = 1, 2$.

By definition, the confidences of all μ-level confidence rules $Y \rightarrow Target$ in $\mathcal{D}$ satisfy: $\frac{supp_1(Y)*|\mathcal{D}_1'|}{supp_1(Y)*|\mathcal{D}_1'|+supp_2(Y)*|\mathcal{D}_2'|} \geq 1-\mu$, where $|\mathcal{D}_i'|$ is the number of transactions in $\mathcal{D}_i'$. So, $\frac{supp_1(Y)}{supp_2(Y)} \geq \frac{|\mathcal{D}_2'|}{|\mathcal{D}_1'|} * \frac{1-\mu}{\mu}$. This means that, for any itemset Y, if its support ratio, $\frac{supp_1(Y)}{supp_2(Y)}$, is $\geq \frac{|\mathcal{D}_2'|}{|\mathcal{D}_1'|} * \frac{1-\mu}{\mu}$, then the rule $Y \rightarrow Target$ has a confidence $\geq 1-\mu$ in $\mathcal{D}$. Therefore, the problem of mining μ-level confidence rules is transformed to discovering all itemsets Y whose support ratio from $\mathcal{D}_2'$ to $\mathcal{D}_1'$ is $\geq \frac{|\mathcal{D}_2'|}{|\mathcal{D}_1'|} * \frac{1-\mu}{\mu}$. Obviously, this is equivalent to the problem of mining *ρ-emerging patterns* (ρ-EPs) [4] from $\mathcal{D}_2'$ to $\mathcal{D}_1'$, where $\rho = \frac{|\mathcal{D}_2'|}{|\mathcal{D}_1'|} * \frac{1-\mu}{\mu}$. In [4], ρ-EPs from $\mathcal{D}_2'$ to $\mathcal{D}_1'$ is defined as those itemsets whose support ratio (or growth rate) from $\mathcal{D}_2'$ to $\mathcal{D}_1'$ is $\geq \rho$. Typically, it is difficult to find all ρ-EPs over two large datasets though some border-based algorithms [4] have been proposed to find some ρ-EPs (including some long EPs which cannot be efficiently discovered by naive algorithms). While this reduction allows us to find some μ-level confidence rules, it is still a problem needing further investigation to find all of them.

5 Experimental Results

We selected three datasets from UCI Machine Learning repository [8]: the Mushroom dataset, the Cleveland heart disease dataset, and the Boston housing dataset, to evaluate our ideas and algorithms. In this work, discretization of numeric attributes is performed using the techniques discussed in [7,6].

As discussed in Section 3, each jumping EP from $\mathcal{D}_2'$ to $\mathcal{D}_1'$ corresponds to a desired top rule. Therefore, we mainly present the results about jumping EPs. We use a certain number of borders in the form of $<\mathcal{L}, \mathcal{R}>$ to represent the discovered jumping EPs, where $\mathcal{L}$ may contain many itemsets but $\mathcal{R}$ is a singleton set.

For the Mushroom dataset, setting TARGET as {POISONOUS} or {EDIBLE} or $\{Population = several, Habitat = leaves\}$, we considered these questions.

- How many top rules are in this dataset approximately?
- What are the shortest and longest lengthes of the discovered antecedents?
- Among the discovered top rules, what are the biggest and smallest supports?

The answers to these questions may help us classify new mushroom instances and help us recognize multi-feature characteristics of mushroom. The smallest support among all discovered top rules is used to show the advantages of our proposed algorithms: our approach can efficiently find some top rules which cannot be found by other approaches. Interestingly, there are *no* jumping EPs from $\mathcal{D}'_2$ to $\mathcal{D}'_1$ when TARGET is set as $\{Population = scattered, Habitat = grasses\}$; this knowledge is useful as it reveals that this TARGET does *not* have 100% dependency on *any* itemsets except TARGET itself.

We also address similar questions for the Cleveland heart disease dataset and the Boston housing dataset. Regarding the first question, we set TARGET as {PRESENCE}, {ABSENCE}, and $\{CP = 1, fbs = 1\}$ for the Cleveland dataset, HICRIME, LOWCRIME, and GOODHOUSE for the Boston housing dataset. We believe the discovered top rules would be useful for domain experts to have a better understanding about the symptom dependency in heart disease patients or to obtain deeper demographic information about the suburbs of Boston.

The experiments were done on a DEC Alpha Server 8400 machine with CPU Speed 300MHZ and 8G memory. This machine works in a network environment and there are many users (e.g. 74) and high load average. The purpose of the experiments is to show the efficiency of the algorithms. We summarize the experimental results as follows.

TARGET	#borders	itemsets in $\mathcal{L}$	itemsets in $\mathcal{R}$	#rules	support(%)	time
		s'est,l'est	Avg. length		largest, smallest	
Edible	4208	1, 5	21	$\approx 4.33 * 10^8$	33.09, 0.01	6690.1s
Poisonous	3916	1, 6	21	$\approx 3.75 * 10^8$	27.42, 0.01	6686.4s
PopHab*	48	2, 6	20	$\approx 2.72 * 10^7$	0.59, 0.01	485.8s
Absence	143	1, 6	13	12636	23.91, 0.34	7.32s
Presence	123	2, 6	13	195552	9.76, 0.34	8.65s
TypicalSymp*	3	3, 7	12	2000	0.67, 0.34	0.24s
HiCrime*	40	2, 7	13	56232	2.57, 0.2	3.41s
Lowcrime*	165	1, 7	13	328750	9.29, 0.2	11.80s
GoodHouse*	3	3, 4	11	704	0.2, 0.2	0.37s

In the first column of the table, the meanings of the targets with "*" are as follows. PopHab: POPULATION = several & HABITAT = leaves; TypicalSymp: CHEST_PAIN_TYPE = typical angina & FASTING_BLOOD_SUGAR $> 120mg/dl$; HiCrime: per capita crime rate $\geq 10\%$; LowCrime: per capita crime rate $\leq 0.1\%$; GoodHouse: pupil-teacher ratio $\geq 16\%$ & lower status of the population $\leq 10\%$ & \$10,000 $\leq$ median value of owner-occupied homes $<$ \$20,000.

The second column shows the number of borders representing the discovered jumping EPs. Column 3 shows the lengthes of the shortest and longest itemsets in the left-hand bounds of the borders. Column 4 shows the average length of all itemsets in the right-hand bounds. Column 5 shows the approximate or exact numbers of the top rules. Column 6 shows the largest and the smallest supports among all top rules.

6 Extracting Top Rules with Support or Length Constraints

Observe that the number of top rules is far larger than the number of borders. Importantly, this confirms the effectiveness of the border representation mechanism. In our border-based algorithms, we do not need to enumerate all jumping EPs (or, equivalently, top rules). However, if we are interested in some top rules, for example the first 100 (largest support) top rules, the border representations allow us to easily generate them. Indeed, because those top rules whose antecedents are the itemsets in the left-hand bounds of the discovered borders have the largest supports among all top rules, we can start the search with the itemsets in the left-hand bounds and then their immediate superset itemsets, and so on, covered by the borders. Furthermore, if we are interested in those top rules whose left-hand sides contain, for example, < 8 items, the discovered borders also allow us to find them quickly. Other kinds of interesting top rules, such as the ones in terms of the neighborhood-based unexpectedness, can be found by the techniques discussed in [3].

7 Conclusion

In this paper we have introduced the problem of mining high confidence association rules, and considered the efficient mining of top rules, of zero-confidence rules, and of μ-level confidence rules. Fundamentally different from the traditional approaches to discovering high-confidence rules, we have used a novel dataset partitioning technique and two border-based algorithms to discover the desired jumping EPs of the two relevant sub-datasets. Then, the discovered jumping EPs are used to construct the top rules. The advantages of our approach include: (i) the algorithms can find the top rules without the constraint of support threshold; (ii) the discovered top rules are succinctly represented by borders. The use of borders help us avoid exponential enumeration of huge collections of itemsets. This approach effectively and efficiently discovered top confidence rules from real high dimensional datasets.

References

1. Agrawal R., Imielinski T., Swami A.: Mining association rules between sets of items in large databases. Proc. 1993 ACM-SIGMOD Int. Conf. Management of Data. Washington, D.C., (1993) 207–216
2. Bayardo, R.J.: Brute-force mining of high-confidence classification rules. Proc. of the Third Int'l Conf. on Knowledge Discovery and Data Mining. (1997) 123–126.
3. Dong, G., Li, J.: Interestingness of discovered association rules in terms of neighborhood-based unexpectedness. Proceedings of Pacific Asia Conference on Knowledge Discovery in Databases (PAKDD'98), Melbourne. (1998) 72–86
4. Dong, G., Li, J.: Efficient mining of emerging patterns: Discovering trends and differences. SIGKDD'99, San Diego. (to appear)
5. Dong, G., Li, J., Zhang, X.: Discovering jumping emerging patterns and experiments on real datasets. Proc. the 9th International Database Conference (IDC'99), Hong Kong. (to appear)
6. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. IJCAI-93. (1993) 1022 – 1027
7. Kohavi, R., John, G., Long, R., Manley, D., Pfleger, K.: MLC++: a machine learning library in C++. Tools with artificial intelligence. (1994) 740 – 743
8. Murphy, P.M., Aha, D.W.: UCI Repository of machine learning database, [http://www.cs.uci.edu/ mlearn/mlrepository.html]. Irvine, CA: University of California, Department of Information and Computer Science (1994)