Available online at www.sciencedirect.com
SCIENCE DIRECT®
International Journal of Approximate Reasoning 40 (2005) 44–54
ELSEVIER
INTERNATIONAL JOURNAL OF APPROXIMATE REASONING
www.elsevier.com/locate/ijar

Mining association rules with multiple minimum supports using maximum constraints

Yeong-Chyi Lee [a], Tzung-Pei Hong [b,*], Wen-Yang Lin [c]

[a] *Institute of Information Engineering, I-Shou University, Kaohsiung 840, Taiwan, ROC*
[b] *Department of Electrical Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan, ROC*
[c] *Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan, ROC*

Received 1 July 2004; accepted 1 November 2004
Available online 6 January 2005

Abstract

Data mining is the process of extracting desirable knowledge or interesting patterns from existing databases for specific purposes. Most of the previous approaches set a single minimum support threshold for all the items or itemsets. But in real applications, different items may have different criteria to judge its importance. The support requirements should then vary with different items. In this paper, we provide another point of view about defining the minimum supports of itemsets when items have different minimum supports. The maximum constraint is used, which is well explained and may be suitable to some mining domains. We then propose a simple algorithm based on the Apriori approach to find the large-itemsets and association rules under this constraint. The proposed algorithm is easy and efficient when compared to Wang et al.'s under the maximum constraint. The numbers of association rules and large itemsets obtained by the proposed mining algorithm using the maximum constraint are also less than those using the minimum constraint. Whether to adopt the proposed approach thus depends on the requirements of mining problems. Besides, the granular computing technique of bit strings is used to speed up the proposed data mining algorithm.

Keywords: Data mining; Multiple minimum supports; Association rule; Maximum constraint

* Corresponding author. Tel.: +886 7 591 9031; fax: +886 7 591 9049.
E-mail address: tphong@nuk.edu.tw (T.-P. Hong).

0888-613X/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.ijar.2004.11.006

1. Introduction

Knowledge discovery in databases (KDD) has become a process of considerable interest in recent years as the amounts of data in many databases have grown tremendously large. KDD means the application of non-trivial procedures for identifying effective, coherent, potentially useful, and previously unknown patterns in large databases [6]. The KDD process generally consists of pre-processing, data mining and post-processing. Due to the importance of data mining to KDD, many researchers in database and machine learning fields are primarily interested in this new research topic because it offers opportunities to discovering useful information and important relevant patterns in large databases, thus helping decision-makers easily analyze the data and make good decisions regarding the domains concerned.

Depending on the types of databases processed, mining approaches may be classified as working on transaction databases, temporal databases, relational databases, and multimedia databases, among others. On the other hand, depending on the classes of knowledge derived, mining approaches may be classified as finding association rules, classification rules, clustering rules, and sequential patterns [4], among others. Among them, finding association rules in transaction databases is most commonly seen in data mining [1,3,5–8,16–19].

An association rule can be expressed as the form $A \rightarrow B$, where A and B are sets of items, such that the presence of A in a transaction will imply the presence of B. Two measures, support and confidence, are evaluated to determine whether a rule should be kept. The support of a rule is the fraction of the transactions that contain all the items in A and B. The confidence of a rule is the conditional probability of the occurrences of items in A and B over the occurrences of items in A. The support and the confidence of an interesting rule must be larger than or equal to a user-specified minimum support and a minimum confidence, respectively.

Most of the previous approaches set a single minimum support threshold for all the items or itemsets. But in real applications, different items may have different criteria to judge its importance. The support requirements should then vary with different items. For example, the minimum supports for cheaper items may be set higher than those for more expensive items. In the past, Liu et al. [14] proposed an approach for mining association rules with non-uniform minimum support values. Their approach allowed users to specify different minimum supports to different items. They also defined the minimum support value of an itemset as the lowest minimum supports among the items in the itemset. This assignment of minimum supports to itemsets is, however, not always suitable for application requirements. For example, assume the minimum supports of items A and B are respectively set at 20% and 40%. As well known, the minimum support of an item means the occurrence frequency of that item must be larger than or equal to the threshold to be further considered in the later mining process. If the support of an item is not larger than or equal to the threshold, this item is not thought of as worth considering.

When the minimum support value of an itemset is defined as the lowest minimum supports of the items in it, the itemset may be large, but items included in it may be small. In this case, it is doubtable whether this itemset is worth considering. For the example described above, if the support of item B is 30%, smaller than its minimum support 40%, then the 2-itemset $\{A, B\}$ should not be worth considering. It is thus reasonable in some sense that the occurrence frequency of an interesting itemset must be larger than the maximum of the minimum supports of the items contained in it.

Wang et al. [20] proposed a mining approach, which allowed the minimum support value of an itemset to be any function of the minimum support values of items contained in the itemset. Although their approach is flexible in assigning the minimum supports to itemsets, its time complexity is high due to its generality. In this paper, we thus propose a simple and efficient algorithm based on the Apriori approach to generate the large itemsets under the maximum constraints. Note that if the mining problem is not under the maximum constraint, then Wang et al.'s approach is a good choice.

The remaining parts of this paper are organized as follows. Some related mining algorithms are reviewed in Section 2. The proposed data-mining algorithm under the maximum constraint is described in Section 3. An example to illustrate the proposed algorithm is given in Section 4. The granular computing technique of bit strings for speeding up the proposed algorithm is described in Section 5. Conclusion and discussion are given in Section 6.

2. Review of related mining algorithms

The goal of data mining is to discover important associations among items such that the presence of some items in a transaction will imply the presence of some other items. To achieve this purpose, Agrawal and his co-workers proposed several mining algorithms based on the concept of large itemsets to find association rules in transaction data [1–4]. They divided the mining process into two phases. In the first phase, candidate itemsets were generated and counted by scanning the transaction data. If the number of an itemset appearing in the transactions was larger than a pre-defined threshold value (called minimum support), the itemset was considered a large itemset. Itemsets containing only one item were processed first. Large itemsets containing only single items were then combined to form candidate itemsets containing two items. This process was repeated until all large itemsets had been found. In the second phase, association rules were induced from the large itemsets found in the first phase. All possible association combinations for each large itemset were formed, and those with calculated confidence values larger than a predefined threshold (called minimum confidence) were output as association rules. The above basic data mining process may be summarized as follows [10].

(1) Determine user-specified thresholds, including the minimum support value and the minimum confidence value.
(2) Find large itemsets in an iterative way. The count of a large itemset must exceed or equal the minimum support value.

(3) Utilize the large itemsets to generate association rules, whose confidence must exceed or equal the minimum confidence value.

A variety of mining approaches based on the Apriori algorithm were proposed, each for a specific problem domain, a specific data type, or for improving its efficiency. In these approaches, the minimum supports for all the items or itemsets to be large are set at a single value. But in real applications, different items may have different criteria to judge its importance. Liu et al. [14] thus proposed an approach for mining association rules with non-uniform minimum support values. Their approach allowed users to specify different minimum supports to different items. The minimum support value of an itemset is defined as the lowest minimum supports among the items in the itemset. Wang and Han [20] then generalized the above idea and allowed the minimum support value of an itemset to be any function of the minimum support values of items contained in the itemset. They proposed a bin-oriented, non-uniform support constraint. Items were grouped into disjoint sets called bins, and items within the same bin were regarded as non-distinguishable with respect to the specification of a minimum support. Although their approach is flexible in assigning the minimum supports to itemsets, the mining algorithm is a little complex due to its generality.

As mentioned before, it is meaningful to assign the minimum support of an itemset as the maximum of the minimum supports of the items contained in the itemset. Although Wang et al.'s approach can solve this kind of problems, the time complexity is high. Below, we will propose an efficient algorithm based on the Apriori approach to generate the large itemsets level by level. Some pruning can also be easily done to save the computation time.

3. The proposed mining algorithm under the maximum constraint

In the proposed algorithm, items may have different minimum supports and the maximum constraint is adopted in finding large itemsets. That is, the minimum support for an itemset is set as the maximum of the minimum supports of the items contained in the itemset. Under the constraint, the characteristic of level-by-level processing is kept, such that the original Apriori algorithm can be easily extended to find the large itemsets.

The proposed algorithm first finds all the large 1-itemsets L_1 for the given transactions by comparing the support of each item with its predefined minimum support. After that, candidate 2-itemsets C_2 can be formed from L_1. Note that the supports of all the large 1-itemsets comprising each candidate 2-itemset must be larger than or equal to the maximum of the minimum supports of them. This feature provides a good pruning effect before the database is scanned for finding large 2-itemsets.

The proposed algorithm then finds all the large 2-itemsets L_2 for the given transactions by comparing the support of each candidate 2-itemset with the maximum of the minimum supports of the items contained in it. The same procedure is repeated until all large itemsets have been found. The details of the proposed mining algorithm under the maximum constraint are described below.

The multiple min-supports mining algorithm using maximum constraints

INPUT: A set of n transaction data T, a set of p items to be purchased, each item t_i with a minimum support value m_i, $i = 1$ to p, and a minimum confidence value λ.

OUTPUT: A set of association rules in the criterion of the maximum values of minimum supports.

STEP 1: Calculate the count c_k of each item t_k, $k = 1$ to p, as its occurrence number in the transactions; derive its support value s_{t_k} as

$$s_{t_k} = \frac{c_k}{n}. \tag{1}$$

STEP 2: Check whether the support s_{t_k} of each item t_k is larger than or equal to its predefined minimum support value m_{t_k}. If t_k satisfies the above condition, put it in the set of large 1-itemsets (L_1). That is:

$$L_1 = \{t_k \mid s_{t_k} \geqslant m_{t_k}, 1 \leqslant k \leqslant p\}. \tag{2}$$

STEP 3: Set $r = 1$, where r is used to keep the current number of items in an itemset.

STEP 4: Generate the candidate set C_{r+1} from L_r in a way similar to that in the Apriori algorithm [3] except that the supports of all the large r-itemsets comprising each candidate $(r + 1)$-itemset I_k must be larger than or equal to the maximum (denoted as m_{I_k}) of the minimum supports of items in these large r-itemsets.

STEP 5: Calculate the count c_{I_k} of each candidate $(r + 1)$-itemset I_k in C_{r+1}, as its occurrence number in the transactions; derive its support value s_{I_k} as

$$s_{I_k} = \frac{c_{I_k}}{n}. \tag{3}$$

STEP 6: Check whether the support s_{I_k} of each candidate $(r + 1)$-itemset I_k is larger than or equal to m_{I_k} (obtained in STEP 4). If I_k satisfies the above condition, put it in the set of large $(r + 1)$-itemsets (L_{r+1}). That is:

$$L_{r+1} = \{I_k \mid s_{I_k} \geqslant m_{I_k}, 1 \leqslant k \leqslant |C_{r+1}|\}. \tag{4}$$

STEP 7: IF L_{r+1} is null, do the next step; otherwise, set $r = r + 1$ and repeat STEPs 4 to 7.

STEP 8: Construct the association rules for each large q-itemset I_k with items $\{I_{k_1}, I_{k_2}, \ldots, I_{k_q}\}$, $q \geqslant 2$, by the following substeps:

(a) Form all possible association rules as follows:

$$I_{k_1} \wedge \cdots \wedge I_{k_{j-1}} \wedge I_{k_{j+1}} \wedge \cdots \wedge I_{k_q} \to I_{k_j},\ j = 1 \text{ to } q. \tag{5}$$

(b) Calculate the confidence values of all association rules using the formula:

$$\frac{s_{I_k}}{s_{I_{k_1} \wedge \cdots \wedge I_{k_{j-1}} \wedge I_{k_{j+1}} \wedge \cdots \wedge I_{k_q}}}. \tag{6}$$

STEP 9: Output the rules with confidence values larger than or equal to the predefined confidence value λ.

4. An example

In this section, an example is given to demonstrate the proposed data-mining algorithm. This is a simple example to show how the proposed algorithm can be used to generate association rules from a set of transactions with different minimum support values defined on different items. Assume the 10 transactions shown in Table 1 are used for mining. Each transaction consists of two features, transaction identification (TID) and items purchased. Also assume that the predefined minimum support values for items are defined in Table 2. Moreover, the confidence value λ is set at 0.85 to be a threshold for the interesting association rules.

In order to find the association rules from the data in Table 1 with the multiple predefined minimum support values, the proposed mining algorithm proceeds as follows.

Table 1
The set of 10 transaction data for this example

TID	Items
1	*ABDG*
2	*BDE*
3	*ABCEF*
4	*BDEG*
5	*ABCEF*
6	*BEG*
7	*ACDE*
8	*BE*
9	*ABEF*
10	*ACDE*

Table 2
The predefined minimum support values for items

Item	*A*	*B*	*C*	*D*	*E*	*F*	*G*
Min-Sup	0.4	0.7	0.3	0.7	0.6	0.2	0.4

STEP 1: The count and support of each item occurring in the 10 transactions in Table 1 are to be found. Take item A as an example. The count of item A is 6, and its support value is calculated as 6/10 (=0.6). The support values of all the items for the 10 transactions are shown in Table 3.

STEP 2: The support value of each item is compared with its predefined minimum support value. Since the support values of items A, B, C, E and F are respectively larger than or equal to their predefined minimum supports, these five items are then put in the large 1-itemsets L_1.

STEP 3: r is set at 1, where r is used to keep the current number of items in an itemset.

STEP 4: The candidate set C_2 is generated from L_1, and the supports of the two items in each itemset in C_2 must be larger than or equal to the maximum of their predefined minimum support values. Take the possible candidate 2-itemset $\{A, C\}$ as an example. The supports of items A and C are 0.6 and 0.4 from STEP 1, and the maximum of their minimum support values is 0.4. Since both of the supports of these two items are larger than 0.4, the itemset $\{A, C\}$ is put in the set of candidate 2-itemsets. On the contrary for another possible candidate 2-itemset $\{A, B\}$, since that the support (0.6) of item A is smaller than the maximum (0.7) of their minimum support values, the itemset $\{A, B\}$ is not a member of C_2. All the candidate 2-itemsets generated in this way are found as: $C_2 = \{\{A, C\}, \{A, E\}, \{B, E\}, \{C, F\}\}$.

STEP 5: The count and support of each candidate itemset in C_2 are found from the given transactions. Results are shown in Table 4.

STEP 6: The support value of each candidate 2-itemset is then compared with the maximum of the minimum support values of the items contained in the itemset. Since the support values of all the candidate 2-itemsets $\{A, C\}$ and $\{B, E\}$ satisfy the above condition, these four itemsets are then put in the set of large 2-itemsets L_2.

STEP 7: Since L_2 is not null, r is set at 2 and STEPs 4 to 7 are repeated. No candidate 3-itemset, C_3, is generated and L_3 is thus null. The next step is then executed.

Table 3
The support values of all the items for the given 10 transactions

Item	A	B	C	D	E	F	G
Support	0.6	0.8	0.4	0.5	0.9	0.3	0.3

Table 4
The support values of all the candidate 2-itemsets

2-Itemset	A, C	A, E	B, E	C, F
Support	0.4	0.5	0.7	0.2

STEP 8: The association rules for each large q-itemsets, $q \geqslant 2$, are constructed by the following substeps:

(a) All possible association rules are formed as follows:
   (1) "If A is bought, then C is bought";
   (2) "If C is bought, then A is bought";
   (3) "If B is bought, then E is bought";
   (4) "If E is bought, then B is bought".

(b) The confidence factors of the above association rules are calculated. Take the first possible association rule "If A is bought, then C is bought" as an example. The confidence factor for this rule is then:

$$\frac{s_{A\cap C}}{s_A} = \frac{0.4}{0.6} = 0.67.$$

Results for all the four association rules are shown as follows:
   (1) "If A is bought, then C is bought" with a confidence factor of 0.67;
   (2) "If C is bought, then A is bought" with a confidence factor of 1.0;
   (3) "If B is bought, then E is bought" with a confidence factor of 0.875;
   (4) "If E is bought, then B is bought" with a confidence factor of 0.875.

STEP 9: The confidence factors of the above association rules are compared with the predefined confidence threshold λ. Assume the confidence λ is set at 0.85 in this example. The following four rules are thus output:
   (1) "If C is bought, then A is bought" with a confidence factor of 1.0;
   (2) "If B is bought, then E is bought" with a confidence factor of 0.875;
   (3) "If E is bought, then B is bought" with a confidence factor of 0.875.

In this example, two large q-itemsets, $q \geqslant 2$, and three association rules are generated. Note that if the transactions are mined using the minimum constraint proposed in [14], 18 large q-itemsets, $q \geqslant 2$, are found. The proposed mining algorithm using the maximum constraint thus finds less large itemsets and association rules than that using the minimum constraint. The proposed algorithm can, however, find the large itemsets level by level without backtracking. It is thus more time-efficient than that with the minimum constraint.

5. Speeding up by granular computing

In [11,12], Lin successfully applied the granular computing technique of bit strings to mining association rules from relational databases and showed the computational time was less than the Aprior algorithm. He pointed out that attribute values could

Table 5
The granular representation for the above example

Item	Equivalence class	Granular representation	Count
A	{TID1, TID3, TID5, TID7, TID9, TID10}	{1010101011}	6
B	{TID1, TID2, TID3, TID4, TID5, TID6, TID8, TID9}	{1111110110}	8
C	{TID3, TID5, TID7, TID10}	{0010101001}	4
D	{TID1, TID2, TID4, TID7, TID10}	{1101001001}	5
E	{TID2, TID3, TID4, TID5, TID6, TID7, TID8, TID9, TID10}	{0111111111}	9
F	{TID3, TID5, TID9}	{0010100010}	3
G	{TID1, TID4, TID6}	{1001010000}	3

Table 6
Using the boolean AND operation to find the granules for C_2

2-Item	Granular operation	Granular representation	Count
A AND C	{1010101011} ∩ {0010101001}	{0010101001}	4
A AND E	{1010101011} ∩ {0111111111}	{0010101011}	5
B AND E	{1111110110} ∩ {0111111111}	{0111110110}	7
C AND F	{0010101001} ∩ {0010100010}	{0010100000}	2

be regarded as granules and a granule was a subset of entities that had the same property. A granule could then be thought of as an equivalence class of attribute values and represented by a bit pattern [12,13,15]. At a certain bit of a bit pattern, the value 1 indicated the corresponding tuple had the attribute value and the value 0 indicated the corresponding tuple did not. Bit operations were then used to speed up the processing of bit strings.

Lin's approach can easily be used in our algorithm for mining from a transaction database. An item or an itemset is regarded as an equivalence class (a granule). If a transaction contains a certain item, the transaction then belongs to the equivalence class of the item and the corresponding bit in its granular representation is set at 1. The granular representation for the data in Table 1 is shown in Table 5.

In this example, A, B, C, E and F are large 1-itemsets. According to our algorithm, the candidate 2-itemsets are found as: $C_2 = \{\{A,C\}, \{A,E\}, \{B,E\}, \{C,F\}\}$. The boolean AND operation can then be used to form the bit patterns of the 2-itemsets. The results are shown in Table 6.

Since the two 2-itemsets, $\{A,C\}$ and $\{B,E\}$, have their supports larger than the support constraint, they are then put in the large 2-itemsets. Itemsets with more items can be formed in the similar way.

6. Conclusion

In this paper, we have provided another point of view about defining the minimum supports of itemsets when items have different minimum supports. The maximum constraint is used, which has been well explained and may be suitable to some mining domains. We have then proposed a simple and efficient algorithm based on the Apriori approach to find the large-itemsets and association rules under this constraint. The proposed algorithm is much easier than that proposed by Wang et al. [20] under the maximum constraint. However, if the mining problem is not under the maximum constraint, Wang et al.'s approach is a good choice. The numbers of association rules and large itemsets obtained by the proposed mining algorithm using the maximum constraint are also less than those using the minimum constraint. Whether to adopt the proposed approach thus depends on mining requirements. Besides, the granular computing technique of bit strings can easily be used to speed up the proposed data mining algorithm.

Acknowledgments

The authors would like to thank Professor T.Y. Lin for his valuable suggestion on granular computing. This research was supported by the National Science Council of the Republic of China under contract NSC93-2213-E-390-001.

References

[1] R. Agrawal, T. Imielinksi, A. Swami, Mining association rules between sets of items in large database, in: The ACM SIGMOD Conference, 1993, pp. 207–216.
[2] R. Agrawal, T. Imielinksi, A. Swami, Database mining: a performance perspective, IEEE Transactions on Knowledge and Data Engineering 5 (6) (1993) 914–925.
[3] R. Agrawal, R. Srikant, Fast algorithm for mining association rules, in: The International Conference on Very Large Data Bases, 1994, pp. 487–499.
[4] R. Agrawal, R. Srikant, Mining sequential patterns, in: The Eleventh IEEE International Conference on Data Engineering, 1995, pp. 3–14.
[5] R. Agrawal, R. Srikant, Q, Vu, Mining association rules with item constraints, in: The Third International Conference on Knowledge Discovery in Databases and Data Mining, 1997, pp. 67–73.
[6] W.J. Frawley, G. Piatetsky-Shapiro, C.J. Matheus, Knowledge discovery in databases: an overview, The AAAI Workshop on Knowledge Discovery in Databases (1991) 1–27.
[7] T. Fukuda, Y. Morimoto, S. Morishita, T. Tokuyama, Mining optimized association rules for numeric attributes, in: The ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, 1996, pp. 182–191.
[8] J. Han, Y. Fu, Discovery of multiple-level association rules from large database, in: The Twenty-first International Conference on Very Large Data Bases, Zurich, Switzerland, 1995, pp. 420–431.
[9] T.P. Hong, C.S. Kuo, S.C. Chi, Mining association rules from quantitative data, Intelligent Data Analysis 3 (5) (1999) 363–376.
[10] T.P. Hong, C.Y. Wang, Y.H. Tao, A new incremental data mining algorithm using pre-large itemsets, Intelligent Data Analysis 5 (2) (2001) 111–129.
[11] T.Y. Lin, Data mining: granular computing approach, in: The Third Pacific-Asia Conference on Knowledge Discovery and Data Mining, Beijing, 1999, pp. 24–33.
[12] T.Y. Lin, Data mining and machine oriented modeling: a granular computing approach, Journal of Applied Intelligence 13 (2) (2000) 113–124.
[13] T.Y. Lin, E. Louie, Finding association rules by granular computing: fast algorithms for finding association rules, Data Mining, Rough Sets and Granular Computing, Physica-Verlag, 2002, 23–42.
[14] B. Liu, W. Hsu, Y. Ma, Mining association rules with multiple minimum supports, in: The 1999 International Conference on Knowledge Discovery and Data Mining, 1999, pp. 337–341.

[15] E. Louie, T.Y. Lin, Finding association rules using fast bit computation: machine-oriented modeling, in: Foundations of Intelligent Systems, Lecture Notes in Artificial Intelligence, Springer-Verlag, 2000, pp. 486–494.
[16] H. Mannila, H. Toivonen, A.I. Verkamo, Efficient algorithm for discovering association rules, The AAAI Workshop on Knowledge Discovery in Databases (1994) 81–192.
[17] J.S. Park, M.S. Chen, P.S. Yu, Using a hash-based method with transaction trimming for mining association rules, IEEE Transactions on Knowledge and Data Engineering 9 (5) (1997) 812–825.
[18] R. Srikant, R. Agrawal, Mining generalized association rules, in: The Twenty-first International Conference on Very Large Data Bases, Zurich, Switzerland, 1995, pp. 407–419.
[19] R. Srikant, R. Agrawal, Mining quantitative association rules in large relational tables, in: ACM SIGMOD International Conference on Management of Data, Montreal, Canada, 1996, pp. 1–12.
[20] K. Wang, J. Han, Mining frequent itemsets using support constraints, in: Proceedings of the 26th International Conference on Very Large Data Bases, 2000, pp. 43–52.