Mining Generalized Multiple-level Association Rules

Show-Jane Yen

Department of Computer Science and Information Engineering
Fu Jen Catholic University, Taipei 242, Taiwan, R.O.C.
sjyen@csie.fju.edu.tw

Abstract. Mining association rules is an important task for knowledge discovery. We can analyze past transaction data to discover customer behaviors such that the quality of business decision can be improved. The strategy of mining association rules focuses on discovering large itemsets which are groups of items which appear together in a sufficient number of transactions. In this paper, we propose a graph-based approach to generate generalized multiple-level association rules from a large database of customer transactions, which describes the associations among items in any concept level. This approach is to scan the database once to construct an association graph, and then traverse the graph to generate large itemsets.

1 Introduction

An *association rule* [1], [3] describes the associations among items in which when some items are purchased in a transaction, the others are purchased, too. In order to find association rules, we need to discover all *large itemsets* from a large database of customer transactions. A large itemset is a set of items which appears often enough within the same transactions.

The following definitions are adopted from [1]. A transaction t *supports* an item x if x is in t. A transaction t supports an itemset X if t supports every item in X. The *support for an itemset* is defined as the ratio of the total number of transactions which support this itemset to the total number of transactions in the database. To make the discussion easier, occasionally, we also let the total number of transactions which support the itemset denote the support for the itemset. Hence, a large itemset is an itemset whose support is no less than a certain user-specified *minimum support*. An itemset of length k is called a k-itemset and a large itemset of length k a large k-itemset.

After discovering all large itemsets, the association rules can be generated as follows: If the large itemset $Y=I_1I_2...I_k$, $k \geq 2$, all rules that reference items from the set $\{I_1, I_2, ..., I_k\}$ can be generated. The antecedent of each of these rules is a subset X of Y, and the consequent Y-X. The confidence of $X \Rightarrow Y\text{-}X$ in database D is the probability that when itemset X occurs in a transaction in D, itemset Y-X also occurs in the same transaction. That is, the ratio of the support for itemset Y to the support for itemset X. A generated rule is an association rule if its confidence achieves a certain user-specified minimum confidence.

D.A. Zighed, J. Komorowski, and J. Zytkow (Eds.): PKDD 2000, LNAI 1910, pp. 679-684, 2000.

A multiple-level association rule [2] is an association rule, in which all items are described by a set of relevant attributes. Each attribute represents a certain concept, and these relevant attributes form a set of multiple-level concepts. For example, in Table 1, food items can be described by the relevant attributes "category", "content" and "brand," and attribute "category" represents the first-level concept (i.e., the highest level concept), attribute "content" the second-level concept and attribute "brand" the third-level concept. There is a set of domain values for an attribute. Each item in the database contains a domain value for each relevant attribute. For example, if the "category", "content" and "brand" of an item have the domain values "bread," "wheat" and "Wonder", respectively, then this item is described as "Wonder wheat bread" in the database.

Table 1. Item description

Category	Content	Brand
bread	wheat	Wonder
milk	chocolate	Dairyland
.....	.....	.....
milk	*2%*	Firemost

From the items in the database, we can derive other items at different concept levels. The domain values of the attribute at the first (i.e., the highest) concept level are the items at the first concept level. An item at the kth concept level can be formed by combining a domain value of the attribute at the kth concept level with an item at the (k-1)th concept level. Hence, item "bread" is at the first concept level, item "wheat bread" at the second concept level, and item " Wonder wheat bread " at the third concept level. A *multiple-level association rule* [2] is an association rule which describes the associations among items at the same concept level. For each concept level, both minimum support and minimum confidence are specified.

One may relax the restriction of mining associations among items at the same concept level to allow the associations among items at any concept level. A *generalized multiple-level association pattern* is a large itemset in which each item may be at any concept level, and the support for the large itemset need to achieve the minimum support specified at the lowest concept level among the concept levels of all items in the large itemset. Besides, the support for each item in a generalized multiple-level association pattern needs to achieve the minimum support specified at its concept level. For an item I to be in a generalized multiple-level association pattern, the items at the corresponding higher concept levels of the item I need to be large at their corresponding concept levels. This is to avoid the generation of many meaningless combinations formed by the items at the corresponding lower concept level of the non-large items. For example, if "bread" is not a large item, item "wheat bread" which is at the corresponding lower concept level of item "bread" need not be further examined.

A *generalized multiple-level association rule* is an association rule which describes the associations among items at any concept level, whose confidence achieves the minimum confidence specified at the lowest concept level among the concept levels of all items in the corresponding generalized multiple-level association pattern. This

paper focuses on the association pattern generation, because after generating the association patterns, the association rules can be generated from the corresponding association patterns.

2 Mining Generalized Multiple-level Association Patterns

In this section, we present the data mining algorithm GMLAPG (Generalized Multiple-Level Association Pattern Generation) to generate generalized multiple-level association patterns, which includes four phases: Large item generation phase, Numbering phase, Association graph construction phase and Association pattern generation phase.

2.1 Large Item Generation

Because each item in the database contains domain values of the relevant attributes, for an attribute, each domain value is arbitrarily given a unique number. Besides, each item in a transaction is denoted according to its domain values.

In the large item generation phase, GMLAPG scans the database to find all large items at every concept level and build the *bit vector* for each large item. The length of each bit vector is the number of transactions in the database. If an item appears in the ith transaction, the ith bit of the bit vector associated with this item is set to 1. Otherwise, the ith bit of the bit vector is set to 0. The bit vector associated with item i is denoted as BV_i. The number of 1's in BV_i is equal to the number of transactions which support the item i, that is, the support for the item i.

Example 1. Consider the database TDB in Table 2 in which there are three concept levels defined, and the items at each concept level are numbered. For example, in Table 2, item "211" can be the item "Wonder wheat bread", where the first number "2" represents the domain value "bread" of the attribute "category" at level-1, the second number "1" for the domain value "wheat" of the attribute "content " at level-2, and the third number "1" for the domain value "Wonder" of the attribute "brand" at level-3. Assume that the minimum supports $\Im_1$, $\Im_2$ and $\Im_3$ specified at the concept levels 1, 2 and 3 are 4, 3 and 3 transactions, respectively.

In the large item generation phase, GMLAPG finds all large items for every concept level from Table 2. For the first level, only the level-1 items in the transactions are considered. After the large item generation phase, the found level-1 large items are "1**" and "2**", because their supports are above the minimum support, and the associated bit vectors are (1111100) and (1110110), respectively, where the notation "*" represents any item. The level-2 large items are 11*, 12*, 21* and 22*, and the level-3 large items are 111, 211 and 221.

Table 2. A database TDB of transactions

TID	Itemset
100	{111, 121, 211, 221}
200	{111, 211, 222, 323}
300	{112, 122, 221, 411}
400	{111, 121}
500	{111, 122, 211, 221, 413}
600	{211, 323, 524}
700	{323, 411, 524, 713}

2.2 A Numbering Method

Property 1: The support for the itemset $\{i_1, i_2, ..., i_k\}$ is the number of 1's in $BVi_1 \wedge BVi_2 \wedge ... \wedge BVi_k$, where the notation "$\wedge$" is a logical AND operation.

Lemma 1. The support for an itemset X that contains both an item x_i and the item χ_i at the corresponding higher concept level of item x_i will be the same as the support for the itemset X-χ_i.

From Lemma 1, we want to discover all generalized multiple-level association patterns which do not contain both an item x and the item which is at the corresponding higher concept level of item x. For this purpose, all large items for each concept level need to be numbered. In the numbering phase, GMLAPG first apply the *TAXC algorithm* to construct taxonomies for all large items. Each level-1 large item constitutes the root node of a taxonomy. For a node which contains a level-k large item Z, each child node of this node contains a level-k+1 large item whose corresponding level-k item is the level-k large item Z.

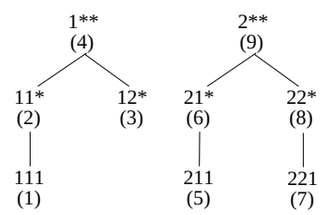

Fig. 1. Two taxonomies for Example 1

For Example 1, two taxonomies with root nodes which contain items 1** and 2**, respectively, are created by applying the algorithm TAXC. Node 1** has two child nodes which contain items 11* and 12*, respectively, because the two items are level-2 large items and their corresponding level-1 item is 1**. The two taxonomies for Example 1 are shown in Fig 1. For a taxonomy, we call the corresponding higher level items of an item x the ancestors of the item x, and the corresponding lower level items of an item y the descendants of the item y.

After constructing the taxonomies, all large items are numbered by applying the *PON method* on the taxonomies. For each taxonomy, PON numbers each item at the taxonomy according to the following order: for each item at the taxonomy, after all descendants of the item are numbered, PON numbers this item immediately, and all items are numbered increasingly. After all items at a taxonomy are numbered, PON numbers items at another taxonomy. For example, all items in the taxonomies in Fig 1 are numbered, where the number within the parentheses below each item is the number of the item.

Lemma 2. If the numbering method PON is adopted to number items, and for every two items i and j (N(i) < N(j)), item ϑ is an ancestor of item i but not an ancestor of item j, then N(ϑ) < N(j), where N(x) denotes the number of large item x after applying the PON method.

2.3 Generalized Association Graph Construction

In the association graph construction phase, GMLAPG applies the GAGC (Generalized Association Graph Construction) algorithm to construct a generalized association graph to be traversed. For every two large items i and j (i < j), if item j is not an ancestor of item i and the number of 1's in $BV_i \wedge BV_j$ achieves the user-specified minimum support, a directed edge from item i to item j is created. Also, itemset (i, j) is a large 2-itemset. In the following, we use the number of an item to represent this item.

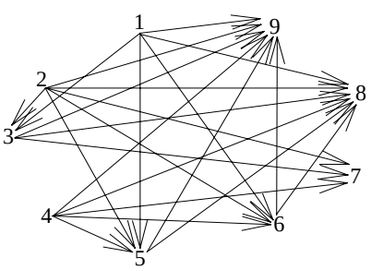

Fig. 2. The generalized association graph for Example 1

After applying the GAGC algorithm in the association graph construction phase, the generalized association graph for Example 1 is constructed in Fig 2, where there are no edges between an item and its ancestors.

2.4 Generalized Multiple-level Association Pattern Generation

In the association pattern generation phase, the algorithm LGDE (Large itemset Generation by Direct Extension) is proposed to generate large k-itemsets (k>2). For each large k-itemset ($k \geq 2$), the last item of the k-itemset is used to extend the large itemset into k+1-itemsets.

Lemma 3. For a large itemset (i_1, i_2, ..., i_k), if there is no directed edge from item i_k to an item v, then itemset (i_1, i_2, ..., i_k, v) cannot be a large itemset.

Suppose (i_1, i_2, ..., i_k) is a large k-itemset. If there is no directed edge from item i_k to an item v, then the itemset need not be extended into k+1-itemset, because (i_1, i_2, ..., i_k, v) must not be a large itemset according to Lemma 3. If there is a directed edge from item i_k to an item u, then the itemset (i_1, i_2, ..., i_k) is extended into k+1-itemset (i_1, i_2, ..., i_k, u). The itemset (i_1, i_2, ..., i_k, u) is a large k+1-itemset if the number of 1's in $BVi_1 \wedge BVi_2 \wedge ... \wedge BVi_k \wedge BV_u$ achieves the minimum support. If no large k-itemsets can be generated, the algorithm LGDE terminates.

For example, there is a directed edge from the last item 3 of the itemset (2, 3) to item 7 in Fig 2. Hence, the 2-itemset (2, 3) is extended into 3-itemset (2, 3, 7). The number of 1's in $BV_2 \wedge BV_3 \wedge BV_7$ is 3. Hence, the 3-itemset (2, 3, 7) is a large 3-itemset, since the number of 1's in its bit vector is no less than the minimum support threshold.

Theorem 1. If the numbering method PON is adopted to number items and the algorithm GAGC is applied to construct a generalized association graph, then any itemset generated by traversing the generalized association graph (i.e., performing LGDE algorithm) will not contain both an item and its ancestor.

3 Conclusion and Future Work

We propose a graph-based approach to discover generalized multiple-level association patterns. Because the GMLAPG algorithm needs only one database scan and performs logical AND operations, this algorithm is efficient when it is compared with other similar problems [1], [2], [3], [4] which need multiple passes over the database.

For our approach, the related information may not fit in the main memory when the size of the database is very large. In the future, we shall consider this problem by reducing the memory space requirement. Also, we shall apply our approach on different applications, such as document retrieval and resource discovery in the world-wide web environment.

References

1. Agrawal R., Srikant R.: Fast Algorithm for Mining Association Rules. Proceedings of the International Conference on Very Large Data Bases (1994) 487–499
2. Han J., Fu Y.: Mining Multiple-Level Association Rules in Large Databases. IEEE Transactions on Knowledge and Data Engineering (1999) 798–805
3. Park J.S., Chen M.S., Yu P.S.: An Effective Hash-Based Algorithm for Mining Association Rules. Proceedings of ACM SIGMOD, Vol. 24, No. 2 (1995) 175–186
4. Srikant R., Agrawal R.: Mining Generalized Association Rules. Proceedings of the International Conference on Very Large Data Bases (1995) 407–419