Asian Journal of Information Management

ISSN 1819-334X

Academic Journals Inc.

www.academicjournals.com

Asian Journal of Information Management 1 (1): 19-26, 2007
ISSN 1819-334X

Fast Algorithm for Mining Multi-Level Association Rules in Large Databases

[1]R.S. Thakur, [2]R.C. Jain and [3]K.R. Pardasani
[1]Department of Master in Computer Application, UIT, RGPV, Bhopal (M.P.) India
[2]Department of Master in Computer Application, SATI, Vidisha (M.P.) India
[3]Department of Applied Mathematics, MANIT, Bhopal (M.P.) India

Abstract: Data Mining refers to extracting knowledge from large amount of data. The Discovery of interesting association relationships among huge amount of data will help marketing, decision making and business management. Previous studies on mining association rules find rules at single level. However mining association rules at multiple-level may lead to the discovery of more specific and concrete knowledge from data. In this research we propose a new algorithms for mining multi-level association rules in large databases. It uses concept of counting inference approach that allows performing as few support counts as possible. This new method reduces database scan at each concept level than the other existing algorithm for multi-level association rule mining from large databases.

Key words: Pattern counting inference, key pattern, multi-level association rule, data mining

Introduction

Association rule mining finds interesting association among a large set of data items. With massive amount of data continuously being collected and stored. Many industries are becoming interested in mining association rules from their databases. The discovery of interesting association relationship among huge amount of business transaction records can help in many business decision making process, such as catalogue design, cross marketing and loss leader analysis.

Studies on mining association rules have evolved from techniques for discovery of functional dependencies (Mannila and Raiha, 1987), strong rules (Piatetsky and Shapiro, 1991), classification rules (Han *et al*., 1993; Quinlan, 1992), causal rules (Michalski and Tecuci, 1994), clustering (Fishae, 1987), etc. to disk-based, efficient methods for mining association rules in large sets of transaction data (Agrawal *et al*., 1993; Agrawal and Srikant, 1994, 1995; Park *et al*., 1995). However, previous work has been focused on mining association rules at a single concept level. There are applications, which need to find associations at multiple concept levels. For example, besides finding 80% of customers that purchase milk may also purchase bread, it could be informative to also show that 75% of people buy wheat bread if they buy 2% milk. The association relationship in the latter statement is expressed at a lower level but often carries more specific and concrete information than that in the former. This requires progressively deepening the knowledge mining process for finding refined knowledge from data. The necessity for mining multiple level association rules or using taxonomy information at mining association rules has also been observed by other researchers (Agrawal and Srikant, 1994). To confine the association rules discovered to be strong ones, that is, the patterns which occur relatively frequently and the rules which demonstrate relatively strong implication relationships, the concepts of *minimum support* and *minimum confidence* have been introduced (Agrawal *et al*., 1993; Agrawal and Srikant, 1994). Informally, the support of a pattern A in a set of transactions S is the probability that a transaction in S contains pattern A and the confidence of

Corresponding Author: R.S. Thakur, Department of Master in Computer Application, UIT, RGPV, Bhopal (M.P.) India Tel: 9826241996

A • B in S is the probability that pattern B occurs in S if pattern A occurs in S. For mining multiple-level association rules, concept taxonomy should be provided for generalizing primitive level concepts to high level ones. In many applications, the taxonomy information is either stored implicitly in the database, such as Wonder wheat bread is a wheat bread which is in turn a bread, or computed elsewhere (Han *et al*., 1993). Thus, data items can be easily generalized to multiple concept levels. However, direct application of the existing association rule mining methods to mining multiple-level associations may lead to some undesirable results as presented below.

First large support is more likely to exits at high concept level, such as milk and bread, rather then at low concept levels, such as a particular brand of milk and bread. Therefore, if one wants to find strong association at relatively low concept levels, the minimum support threshold must be reduced substantially. However, this may lead to the generation of many uninteresting associations, such as toy • 2% milk before the discovery of some interesting ones, such as Dairyland 2% milk • wonder wheat bread, because the former may occur more frequently and thus have larger support then the latter.

Second, it is unlikely to find many strong association rules at a primitive concept level, such as the association among particular bar codes, because of the tiny average support for each primitive data item in a very large item set. However, mining association rules at high concept level may often lead to the rules corresponding to prior knowledge and expectations (Klemettinen *et al*., 1994), such as milk • bread, or lead to some uninteresting attribute combinations, such as toy • milk. In order to remove uninteresting rules generated in knowledge mining processes, researchers have proposed some measurements to quantify the usefulness or interestingness of a rule (Piatesky-Shapiro *et al*., 1994) or suggested to put a human in the loop and provide tools to allow human guidance (Borgida and Brachman, 1993). Nevertheless, automatic generation of relatively focused, informative association rules will be obviously more efficient than first generating a large mixture of interesting and uninteresting rules. These observations lead us to examining the methods for mining association rules at multiple concept levels, which may not only discover rules at different levels but also have high potential to find nontrivial, informative association rules because of its flexibility at focusing the attention to different sets of data and applying different thresholds at different levels.

Most of the previous studies on mining multi-level association rules, such as (Han and Fu, 1999); adopt an *Apriori* approach, which required more number of operations for counting pattern supports in the database. In this study, proposed a top-down progressive deepening method by extension of some existing algorithms for mining multi-level association rules. The method first finds large data items at the top-most level and then progressively deepens the mining process into their large descendants at lower concept levels. In this new approach call PASCAL algorithm (Bastide *et al*., 2000) at each concept level for mining all frequent patterns of that level. Due to using concept of key pattern it reduces database passes at each concept level.

Definitions and Concept

We assume that the database contain 1) an item data set which contain the description of each item in I in the form of (A_i, description), where A_i • I and 2) a transaction data set, T, which consists of a set of transaction (T_i {A_p....., A_q}), where T_i is a transaction identifier and A_i • I (for i = p.....q).

Definition 1

A pattern A, is one item A_i or a set of conjunctive items A_i • ..... • A_j where A_i.... A_j • I. The support of a pattern A in a set S, • (A/S), is the number of transaction (in S) which contain A versus the total number of transactions in S. The confidence of A • B in S, • (A • B/S), is the ratio of • (A • B/S) versus • (A/S) i.e., the probability that pattern B occurs in S when pattern A occurs in S.

To find relatively frequent occurring patterns and reasonably strong rule implications, a user or an expert may specify two thresholds: minimum support, • and minimum confidence, • . Notice that for finding multiple-level association rules, different minimum support and/or minimum confidence can be specified at different levels.

Definition 2

A pattern A is large in set S at level *l* if the support of A is no less then its corresponding minimum support threshold • '$_l$. A rule A • B/S is strong if, for a set S, each ancestor (i.e., the corresponding high-level item) of every item in A and B, if any, is frequent at its corresponding level A • B/S is frequent (at the current level) and the confidence of A • B/S is no less then minimum confidence threshold at the current level. The definition implies a filtering process which confines the pattern to be examined at lower level to be only those with large support at their corresponding high level. Based on this definition, the idea of mining multiple-level association rules is illustrated below.

Example 1

Let the query be to find multiple-level strong association in the database in Table 1 for the purchase patterns related to category, content and brand of the food which can only be stored for less than three weeks.

For answering the above query the relevant part of the sales item description relation in Table 2 is fetched and generalized into a generalized sales item description table, as shown in Table 3, in which is tuple represent a generalized item which is the merge of a group of topples which share the same values in the interested attributes for example, the tuple with the same category, content and brand in Table 1 are merged into one, with their *bar codes* replaced by a bar-code set. Each group is then treated as an atomic item in the generation of the lowest level association rules. For example, the association rule generated regarding to milk will be only in relevance to (at the low concept levels) brand (such as Dairyland) and content (such as 2%) but not to size, producer, etc.

The taxonomy information is provided implicitly in Table 3. Let category (such as milk) represent the first-level concept, content (such as 2%) for the second level one and brand (such as Foremost) for the third level one. The table implies a concept tree like Fig. 1.

The process of mining association rules is expected to first discover large patterns and strong association rules at the top-most concept level. Let the minimum support at this level be 5% and the minimum confidence be 50%. One may find the following: a set of single large items (each called a large 1-itemset, with the support ratio in parentheses): bread (25%), meat (10%), milk (20%), vegetable (30%), a set of pair-wised large items (each called a large 2-itemset): (vegetable, bread (19%)....}, etc. and a set of strong association rules, such as, bread • vegetable (76%)....

Table 1: A sales transaction table

Ttransaction_id	Bar_code_set
351428	{17325, 92108, 55349, …..
982510	{92458, 77451, 60395,...}
	{...,...}

Table 2: A sales_item (description) relation

Bar_code	Category	Brand	Content	Size	Storage_pd	Price
17325	milk	foremost	2%	1(ga)	14(day)	$3.89
.	.....	.....	.....	.....	.....	.....

Table 3: A generalized sales_item description table

GID	Bar_code_set	Category	Content	Brand
112	{17325, 31414, 91265 }	Milk	2%	foremost
.....	{.....}	.....	.....	.....

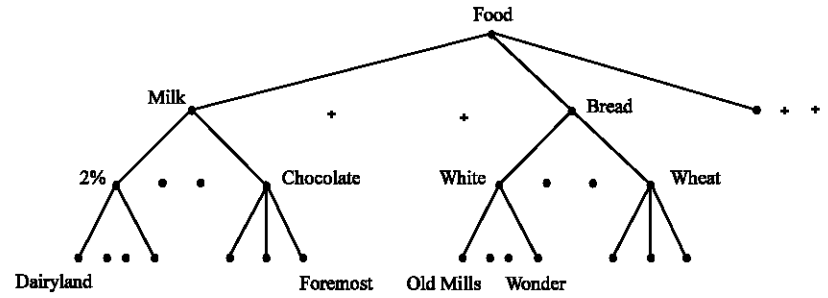

Fig. 1: A taxonomy of the relevant data items

At the second level, only the transactions, which contain the large items at the first level, are examined. Let the minimum support at this level be 2% and the minimum confidence be 40%. One may find the frequent 1-itemsets: lettuce (10%), wheat bread (15%),.... and large 2-itemsets: (2% milk, wheat bread (6%),..... and strong association rule: 2% milk • wheat bread (60%)....., etc.
The process repeats at even lower concept level until no large patterns can be found.

A New Mehod for Mining Multi-Level Association Rules

A method for mining multiple-level association rules is introduced in this section, which uses a hierarchy information encoded transaction table, instead of original transaction table (Han and Fu, 1999). This is based on the following consideration. First, a data mining query is usually in relevance to only a portion of the transaction database, such as food instead of all the items. It is beneficial to first collect the relevant set of data and then work repeatedly on the task-relevant set. Second, encoding can be performed during the collection of task-relevant data and thus there is no extra encoding pass required. Third, an encoding string, which represents a position in a hierarchy, required less bits than the corresponding object-identifier or bar-code. Moreover, encoding makes more items to be merged or removed due to their identical encoding, which further reduces the size of the encoded transaction table. Thus it is often beneficial to use an encoded table although our method does not rely on the derivation of such an encoded table because the encoding can always be performed on the fly.

This new method uses concept of counting inference approach (Bastide *et al*., 2000) at each level rather then Apriori. It allows to perform as few support counts as possible. Using this method, the support of a pattern is determined without accessing the database whenever possible, using the supports of some of its sub-patterns called key patterns. For implementation of this method we used PASCAL algorithm (Bastide *et al*., 2000) that is an optimization of the Apriori algorithm. In PASCAL, frequent patterns are extracted in a levelwise manner: During each iteration, candidate patterns of size k are created by joining the frequent patterns of size k- 1, their supports are determined and infrequent ones are discarded. Using counting inference, if a candidate pattern of size k is a non-key pattern, then its support is equal to the minimal support among the patterns of size k- 1 that are its subsets. This allows to reduce the number of patterns considered during each database pass while counting supports and, even more important, to reduce the total number of passes. This optimization is valid since key patterns have a property that is compatible with the pruning of Apriori: all subsets of a key pattern are key patterns and all supersets of a non-key pattern are non-key patterns.

The above discussion leads to the following algorithm for mining strong multi-level association rules.

Algorithm 1

Find multiple-level large item sets for mining strong ML association rules in a transaction database.

Input

(1) $\mathcal{T}$[l], a hierarchy-information-encoded and task-relevant set of transaction database, in the format of <TID, Itemset>, in which each item in the Itemset contains encoded concept hierarchy information and (2) the minimum support threshold (minsup[*l*]) for each concept level *l*.

Output

Multi-level large item sets.

Method

A top-down, progressively deepening process which collects large item sets at different concept levels as follows.

Starting at level *1*, derive for each level *l*, the large k-items sets, $\mathcal{L}$[*l*, k], for each k and the large item set, $\mathcal{L}\mathcal{L}$[*l*] (for all k's), as follows.

```
for (l: = 1; L[l, 1] • • and l< max_level; l++) do
        { if l = 1 then
                { L[l, 1]: = large_1_itemsets(T[1], l);
                  T[2]: = filtered_t_table(T [1], L[1, 1]);
 }
 else    L[l, 1]: = large_1_itemsets(T[2], 1);
 LL[l ]: = call_PASCAL(L[l, 1], T[2], minsup[l])
}
```

Here PASCAL procedure calls for generating all frequent k-itemsets (k • 2) at each level *l*. The pseudo-code for PASCAL procedure as follows

Procedure

Pascal

Input

(1) $\mathcal{L}$[*l*,1] frequent 1-itemsets at level 1. (2) $\mathcal{T}$[2], a filtered transaction database by $\mathcal{L}$[*1*, 1] (3) minsup[*l*]: the minimum support threshold for each concept level *l*.

Output

All frequent k-itemsets at level *l*.

- ∅ supp: = 1; ∅. key = true
- $\mathcal{L}$[l,0] = ∅;
- for all *l* • $\mathcal{L}$[*l*, 1] do begin
- *l*.pred_supp = 1; *l*.key = (*l*.supp • 1)
- end;
- for (k = 2; $\mathcal{L}$[*l*, k-1]• ∅; k++) do begin
- C_k = Pascal-Gen (L [l, k-1])
- if • c • C_k | c.key then
- forall o • $\mathcal{T}$[2] do begin
- C_o: = subset(C_k, o)
- forall c • C_o | c.key do
- c.supp + +;
- end;

- forall c • C_k do
- if c.supp • minsup[l] then begin
- if c.key and c.supp = c.pred_supp then
- c.key: = false;
- $\mathcal{L}[l, k]$: = $\mathcal{L}[l, k]$ • {c}
- end;
- end;
- return • $_K \mathcal{L}[l, k]$;

The algorithm starts with the empty set, which always has a support of 1 and which is a (by definition as Bastide *et al*., 2000) key pattern (step 1 and 2). In step 3 to 5 frequent 1-patterns are marked as key patterns unless their support is 1. The main loop is similar to the one in Apriori (steps 6 to 20). First, Pascal-Gen is called to compute the candidate patterns. The support of key ones is determined via a database pass (steps 9 to 13).

Then (steps 14-19) the traditional pruning (step 15) is done. At the same time, for all remaining candidate key patterns, it is determined whether they are key or not (step 16 and 17).

Explanation of Algorithm 1

According to Algorithm 1, the discovery of large support items at each level *l* proceeds as follows. At level 1(1 = 1), the large 1-itemsets $\mathcal{L}[l,1]$ is derived from $\mathcal{T}[1]$ by large_1_itemsets ($\mathcal{T}[1]$, *l*), at any other level *l* (*l* • 2), $\mathcal{L}[l,1]$ is derived from $\mathcal{T}[2]$ by large_1_itemsets ($\mathcal{T}[2]$, *l*), after scanning the transaction table, filter out those items whose support is smaller then minsup[*l*]. The filtered transaction table $\mathcal{T}[2]$ is derived by filtered_t_table($\mathcal{T}[1]$, $\mathcal{L}[1,1]$), which uses $\mathcal{L}[1,1]$ as a filter to filter out any item which is not large at level 1 and the transactions which contain no large items.

For k>1 item set table at level *l* is derived as done in the PASCAL algorithm [12], i.e., first compute the candidate set from $\mathcal{L}$ [*l*, k-1] then count support of each item of candidate set in $\mathcal{T}$ [*l* + 1] and collect only those itemsets into $\mathcal{L}[l, k]$ which has support count no less then minsup[*l*]. PASCAL algorithm calls Pascal-Gen procedure for generating candidate sets, which discussed in later.

The large itemsets at level *l*, $\mathcal{L}\mathcal{L}[l]$, is the union of $\mathcal{L}[l, k]$ for all the k's. After finding the large itemsets, the set of association rules for each level l can be derived from the large itemsets $\mathcal{L}\mathcal{L}[l]$ based on the minimum confidence at this level, minconf[*l*]. this is performed as follows[2]. For every large itemset r, if a is a nonempty subset of r, the rule a • r - a is inserted into rule_set[l] if support(r)/ support (a) • minconf[l], where minconf[l] is the minimum confidence at level *l*.

Generation of Candidate Sets

In this section Pascal-Gen procedure have discussed which is used by PASCAL algorithm for generating candidate sets at each concept level. The pseudo-code of this procedure is as follows

Procedure

Pascal-Gen The candidate generation procedure of counting inference algorithm (Bastide *et al*., 2000) is given below.

Input

L_k, the set of frequent k-patterns p with their support p.supp and the p.key flag.

Output

C_k, the set of candidate k+1-patterns c each with the flag c.key, the value c.pred_supp and the support c.supp if c is not a key pattern.

- for i from 1 to | L_k-1| begin
- for j from i+1 to | L_k| begin
- if L_k. itemset$_i$ and L_k. itemset$_j$ have the same (k-1)-prefix
- C_{k+1}: = C_{k+1}• {L_k.itemset$_i$ • L_k.itemset$_j$ }
- else
- break
- end;
- end;
- forall c • C_{k+1} do begin
- c.key:= true; c.pred_supp:= +• ;
- forall k-subsets s of c do begin
- if s • L_k then
- delete c from C_{k+1};
- else begin
- c.pred_supp: = min (c.pred_supp, s.supp);
- if not s.key then c.key: = false;
- end;
- end;
- if not c.key then c.supp: = c.pred_supp;
- end;
- return C_{k+1}.

The way Pascal-Gen operates is basically known from the generator function Apriori-Gen that was introduced by Agrawal *et al*. (1993). In addition to Apriori-Gen's join and prune steps, Pascal-Gen makes the new candidates inherit the fact of being or not a candidate key pattern (step 16) by using Theorem 2 discussed by Bastide *et al*. (2000) and it determines at the same time the support of all non key candidate patterns (step 19) by using Theorem 4 as discussed by Bastide *et al*. (2000).

Conclusions

We have extended the scope of the study of mining association rules among from concept at the same level of a hierarchy to concept of different level of hierarchy in multiple concept level and developed new method for mining multi-level association rules from large transaction databases. This new efficient top-down progressive deepening technique incorporate pattern counting inference approach which is based on the notion of key patterns of equivalence classes of patterns. At each level it allows to count from the dataset the support of some frequent patterns only, the frequent key patterns, rather than counting the support of all frequent patterns. Due to this, it reduces number of database passes at each concept level. When small portion of data are large items at each level, it save a substantial amount of processing. Thus it will be promising algorithms in this circumstance.

References

Agrawal, R., T. Imielinski and A. Swami, 1993. Mining association rules between sets of items in large databases. In: Proc. 1993 ACM-SIGMOD Intl. Conf. Management of Data, Washington, DC, pp: 207-216,

Agrawal, R. and R. Srikant, 1994. Fast algorithms for mining association rules. In Proc. 1994 Intl. Conf. Very Large Data Bases, Santiago, Chile, pp: 487-499.

Agrawal, R. and R. Srikant, 1995. Mining sequential patterns. In: Proc. 1995 Intl. Conf. Data Engineering, Taipei, Taiwan, pp: 3-14.

Bastide, Y., R. Taouil, N. Pasquier, G. Stumme and L. Lakhal, 2000. Mining frequent patterns with counting inference. ACM SIGKDD Explorations Newslett., 2: 66-75.

Borgida, A. and R.J. Brachman, 1993. Loading data into description reasoners. In: Proc. 1993 ACM-SIGMOD Intl. Conf. Management of Data, Washington, DC, pp: 217-226.

Fisher, D., 1987. Improving inference through conceptual clustering. In: Proc. 1987 AAAI Conf., Seattle, Washington, pp: 461-465.

Han, J., Y. Cai and N. Cercone, 1993. Data-driven discovery of quantitative rules in relational databases. IEEE Trans. Knowledge and Data Eng., 5: 29-40.

Han, J. and Y. Fu, 1999. Mining multiple-level association rules in large databases. IEEE Trans. Knowledge Data Eng., 11: 798-804.

Klemettinen, M., H. Mannila, P. Ronkainen, H. Toivo-nen and A.I. Verkamo, 1994. Finding interesting rules from large sets of discovered association rules. In: Proc. 3rd Intl. Conf. on Information and Knowledge Management, Gaithersburg, Maryland, pp: 401-408.

Mannila, H. and K-J. Raiha, 1987. Dependency inference. In: Proc. 1987 Intl. Conf. Very Large Data Bases, Brighton, England, pp: 155-158.

Michalski, R.S. and G. Tecuci, 1994. Machine Learning, A Multistrategy Approach, Vol. 4-Morgan Kaufmann, San Mateo, CA, pp: 141-164.

Park, J.S., M.S. Chen and P.S. Yu, 1995. An effective hash-based algorithm for mining association rules. In: Proc. 1995 ACM-SIGMOD Intl. Conf. Management of Data, San Jose, CA, pp: 175-186.

Piatesky-Shapiro, G. and C.J. Matheus, 1994. The inter-estingness of deviations. In: AAAI'94 Workshop on Knowledge Discovery in Databases, Seattle, WA, pp: 25-36.

Piatetsky-Shapiro, G., 1991. Discovery, Analysis and Presentation of Strong Rule. In: Piatetsky-Shapiro, G. and W.J. Frawley (Eds.), Knowledge Discovery in Databases, AAAI/MIT Press, pp: 229-238.

Quinlan, J.R., 1992. C4-5: Programs for Machine Learning. Morgan Kaufmann Publishers, 8: 302.