Mining Frequent Patterns with Counting Inference at Multiple Levels

Mittar Vishav
Deptt. Of IT
M.M.University,
Mullana

Ruchika Yadav
Deptt. Of CSE
H.C.T.M.
Kaithal

Deepika Sirohi
Deptt. Of IT
M.M.University,
Mullana

ABSTRACT
Mining association rules at multiple levels helps in finding more specific and relevant knowledge. While computing the number of frequency of an item we need to scan the given database many times. So we used counting inference approach for finding frequent itemsets at each concept levels which reduce the number of scan. In this paper, we purpose a new algorithm LWFT which follow the top-down progressive deepening method and it is based on existing algorithms for finding multiple level association rules. This algorithm is efficient for finding frequent itemsets from large databases**.**

Keywords
Multiple-Level Association Rules, Counting inference approach, Level wise filtered tables, Data mining, non-uniform support, Confidence.

1. INTRODUCTION
Finding association rules is one of the most important tasks in data mining. Many industries are taking interest in mining association rules from their databases. The discovery of interesting association relationships among huge amounts of business transaction records can help in many business decision making process, association rules is one of the main popular pattern discovery techniques in knowledge discovery and data mining (KDD). *Association rules mining* finds interesting association among a large set of data items. [1]

The process of extracting the association rules complete in two-phases: the first phase is to mine all frequent patterns; each of these patterns will happen at least as frequently as preset minimum support count. The second phase is to produce strong association rules from the frequent patterns; these rules must assure minimum support and minimum confidence. The performance of discovering association rules is largely determined by the first phase, Association Rule mining techniques can be used to discover unknown or hidden correlation between items found in database of transactions. An association rule [1, 2, 4] is a rule, which implies certain association relationships among a set of objects (such as 'occurs together' or 'one implies to other') in a database. Discovery of association rules can help in business decision making, planning marketing strategies etc. [1, 4]

Mining association rules using basic algorithms may require iterative scanning of large databases, which is costly in processing. Many researchers have focused their work on efficient mining of association rules in databases.

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.[2,4] 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 A→B in S is the probability that pattern B occurs in S if pattern A occurs in S.[7]

Multilevel association rule mining works in two different processes. First of all it finds frequent items at multiple levels and then on the basis these frequent items it generate association rules. The first requirement can be full filled by providing concept taxonomies from the primitive level concepts to higher level. User will provide minimum support and confidence, if minimum support and minimum confidence thresholds at each level are uniform then it may lead to some undesirable result. Because, to find data items at multiple level under the same minimum support and minimum confidence thresholds will not give the desirable result. For example there is a hierarchy in which at level 0 there is food, at level one there are bread, milk and fruit and at level 2 we further put the various brands of these items. Large support is more likely to exist at high concept level such as bread and butter rather than at low concept levels, such as a particular brand of bread and butter. Therefore, if we want to find strong relationship at relatively low level in hierarchy, the minimum support threshold must be reduced substantially.

To remove this problem one should apply different minimum support to different concept levels. This leads to mining
interesting association rules at multiple concept levels, which will find nontrivial, informative association rules because of its flexibilities for focusing the attention to different sets of data and applying different thresholds at different levels [3].

Association rule mining has a wide range of applicability such Market basket analysis, Medical diagnosis/ research, Website navigation analysis, Homeland security and so on. Association rules are used to identify relationships among a set of items in database. These relationships are not based on inherent properties of the data themselves (as with functional dependencies), but rather based on co occurrence of the data items. Association rule and frequent itemset mining became a widely researched area, and hence faster and faster algorithms have been presented. [5]

Most of the previous studies on mining multiple level association rules, adopt an Apriori approach, which required more number of operations for counting pattern supports in the database. So counting inference approach [8] is used in this study. This approach is based on the extraction of maximal frequent patterns, from which all supersets are infrequent and all subsets are frequent. This approach combines a level wise bottom-up traversal with a top-down traversal in order to quickly find the maximal frequent patterns. Then all frequent patterns are derived from these ones and one last database scan is carried on to count their support. [8]

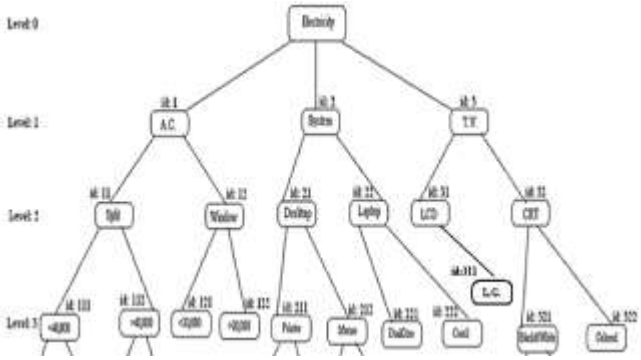
A new algorithm LWFT is proposed which works on top-down progressive deepening method by extension of some existing algorithms for mining multi-level association rules. The method finds frequent itemsets at each level and on the basis of theses itemsets, it will filter the table, which will reduce the size of the database. By using concept of key pattern it reduces database passes at each concept level.

2. Multiple-level Association Rules

In the study of mining multiple levels association rules, a series of algorithms have been proposed to facilitate a top down, progressive deepening method based on the algorithms for mining single 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 multiple-level association rule mining, the items in an itemset are characterized by using a concept hierarchy as shown in the diagram.

Fig: Concept Hierarchy



Mining occurs at multiple levels in the hierarchy. At lowest levels, it might be that no rules may match the constraints. At highest levels, rules can be extremely general. Generally, a top-down approach is used where the support threshold varies from level to level. [9]

3. A Method for Mining Multiple- Level Association Rules

A new method for mining association rules is introduced in this section, which is based on the existing association rules algorithm. This algorithm uses level wise filtered tables. This algorithm uses a hierarchy information encoded transaction table instead of original table. This is based on the following consideration. First collect the relevant set of data and then work repeatedly on the task related set. Second, encoding can be performed during the collection of task related data and thus there is no extra encoding pass required. Third, an encoded string, which represents a position in a hierarchy, requires lesser bits than the corresponding bar code.

Thus, it is often beneficial to use an encoded table. In LWFT algorithm, first of all we find 1-frequent itemsets, after this we filter out non frequent items and transactions from the table and then by using this filtered table k-large itemsets of that level are calculated. This process is repeated for each level *l*.

Generating several transaction tables may seem costly, but it will save a substantial amount of processing if only a small portion of data are large items at each level. Thus it may be a promising algorithm in this circumstance.

Example: Suppose that a shopping transaction database consists of two relations: (1) a sales-item (description) relation (Table 3.1), which consists of a set of attributes: bar-code, category, Type, Quantity, Brand, price, and (2) a sales-transaction table (Table3.2**),** which registers for each transaction, the transaction number and the set of items purchased.

Table 3.1: A sales item (description) relation

ar-code	Category	Type	Quantity	Brand	Price
17325	A.C.	Windows	1	Hitachi	$13399. 89
……..	……..	……..	……..	……..	……..

Table 3.2: A sales-transaction table

Transaction-id	Bar-code set
351428	{17325, 92108, 55349, 88157, …}
982510	{92458, 77451, 60395, …}
……..	{…,…,…,…,…}

As stated above, the taxonomy information for each (grouped) item is encoded as a sequence of digits in the transaction table T[1] (Table 3.3). For example, the item 'mouse desktop systems' is encoded as '212' in which the first digit, '2', represents 'system' at level-1, the second, 'l', for 'Desktop (system)' at level-2,and the third '2', for the 'mouse, at level-3.

Table3.3 Encoded transaction table: T [1]

Transaction Id	Items
T1	{211, 212, 313, 111,122}
T2	{112, 121, 211, 225, 321,313}
T3	{321, 122, 311,111,456}
T4	{122, 132, 555, 231, 313, 212}
T5	{132, 211, 212, 311, 111}
T6	{131, 112, 211, 212, 322, 311}
T7	{111, 121, 211, 221, 413}
T8	{211, 323, 524, 322, 132}
T9	{411, 524, 713}
T10	{111, 211, 222, 411}

The derivation of the large itemsets at level-1proceeds as follows. Let the minimum Support at level 1 be 6 transactions (i.e., minsup[l] = 6). Notice that since the total number of transactions is fixed, the support is expressed in an absolute value rather than a relative percentage, for simplicity. The level-1 large 1-itemset table freq[1,1] can be derived by scanning T[l],registering support of

each generalized item, such as 1**, …….., n**, if a transaction contains such an item (i.e., the item in the transaction belongs to the generalized item I**, ..., n**, respectively), and filtering out those whose accumulated support count is lower than the minimum support. freq[1,1] is then used to filter out:(1) any item which is not large in a transaction, and (2) the transactions in T[1] that contain only small items.

Table :freq[1,1]

Level-1, MinSupport=6, Frequent-1-itemsets	
Itemset	Support
1**	8
2**	8
3**	7

This results in the filtered transaction table T[2] of Figure 3.3.

Table T[2]

Transaction Id	**Items**
T1	{211, 212, 313, 111, 122}
T2	{112, 121, 211, 225, 321, 313}
T3	{321, 122, 311, 111}
T4	{122, 132, 231, 313, 212}
T5	{132, 211, 212, 311, 111}
T6	{131, 112, 211, 212, 322, 311}
T7	{111, 121, 211, 221}
T8	{211, 323, 322, 132}
T9	{ }
T10	{111, 211, 222}

Moreover, since there are only three entries in freq[1,1], the level-1 large-2 itemset table freq[1,2] may contain 3

Table :freq[1,2]

Level-1, MinSupport=6, Frequent-2-itemsets	
Itemset	Support
{1**, 2**}	8
{1**, 3**}	7
{2**, 3**}	6

candidate item {I**, 2****),{2****, 3****} and {1****, 3****} which is supported by 8, 7 and 6 transactions in T[2].In the same manner freq[1,3] can be generated.

Table : freq[1,3]

Level-1, MinSupport=6, Frequent-3-itemsets	
Itemset	Support
{1**,2**,3**}	6

According to the definition of multiple-level association rules only the descendants of the large items at level-1 (i.e., in freq[1,1]) are considered as candidates for the level-2 large 1-itemsets. Let minsup[2] =6. The level-2 large 1-itemsets freq[2,1] can be derived from the filtered transaction table T[2] by accumulating the support count and removing those whose support is smaller than the minimum support, which results in freq[2,1].

Table: freq[2,1]

Level-2, MinSupport=4, Frequent-1-itemsets	
Itemset	Support
11*	7
12*	5
21*	8
31*	6
32*	4

After that table T[2] is filtered using level-2 large 1-itemsets i.e. freq[2,1]. This results in the filtered transaction table T[3].

Table: T[3]

Transaction Id	**Items**
T1	{211, 212, 313, 111, 122}
T2	{112, 121, 211, 321, 313}
T3	{321, 122, 311, 111}
T4	{122, 313, 212}
T5	{211, 212, 311, 111}
T6	{112, 211, 212, 322, 311}
T7	{111, 121, 211, 221}
T8	{211, 323, 322}
T9	{ }
T10	{111,211}

Similarly, the large 2-itemset table freq[2,2] is formed by the combinations of the entries in freq[2,1].

Table: freq[2,2]

Level-2, MinSupport=4, Frequent-2-itemsets	
Itemset	Support
{11*, 12*}	4
{11*, 21*}	6
{11*, 31*}	5
{12*, 21*}	4
{21*, 31*}	4

Likewise, the large 3-itemset table freq[2,3] is formed by the combinations of the entries in freq[2,2] and filtered table T[3].

Table: freq[2,3]

Level-2, MinSupport=4, Frequent-3-itemsets	
Itemset	Support
{11*, 12*, 21*}	4
{11*, 21*, 31*}	4

Finally at level-3 the Minsupport is 2 and the frequent itemset for 1-large itemset of level-3 can be calculated and a new filtered table T[4] can be generated. on the basis of these frequent itemsets and Table T[4], the table freq[3,2] and table freq[3,3] can also be generated.The computation terminates since there is no deeper level in the hierarchy. Note that the derivation also terminates when an empty large 1-itemset table is generated at any level.

Table: freq[3,1]

Level-3, MinSupport=2, Frequent-1-itemsets	
Itemset	Support
111	5
112	2
121	2
211	5
212	4
311	3
321	2
322	2

Table T[4]

Transaction Id	**Items**
T1	{211, 212, 111}
T2	{112, 121, 211, 321}
T3	{321, 311, 111}
T4	{ 212}
T5	{211, 212, 311, 111}
T6	{112, 211, 212, 322, 311}
T7	{111, 121, 211}
T8	{211, 322}
T9	{ }
T10	{111, 211}

Table: freq[3,2]

Level-3, MinSupport=2, Frequent-2-itemsets	
Itemset	Support
{111, 211}	4
{111, 212}	2
{111, 311}	2
{112, 211}	2
{121, 211}	2
{211, 212}	3
{211, 322}	2
{212, 311}	2

Table: freq[3,3]

Level-3, MinSupport=2, Frequent-3-itemsets	
Itemset	Support
{211, 212, 111}	2

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

3.1 Algorithm LWFT

The above discussion leads to the following algorithm for mining interesting multiple-level association rules.

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

Input: (1) D[1], a transaction database, in the format of (ID, Itemset), in which each item in the Itemset contains encoded concept hierarchy information, and (2) the minimum support threshold (minsup[level]) for each concept level 1.

Output: Multiple-level large item sets.

Method: A progressively deepening process, which collects large itemsets at different concept, levels as follows. Starting at level 1, derive for each level 1, the frequent i-items sets, freq[level, i], for each i, and the frequent item set, freq[level] (for all i's), as follows: -

Steps:

(1) for level=1 to freq[level,1] != NULL && (level < max_level)

(2) if level =1

(3) set freq[level,1] = get_frequent_itemsets (D[1], level)

(4) set D[level +1]=get_filtered_database(D[level], freq[level,1])

(5) end if

(6) else

(7) set freq[level+1,1] = get_frequent1_itemsets (D[level], freq[level,1])

(8) set D[level +1] = get_filtered_database(D[level], freq[level,1])

(9) end else

(10) LL[level]=PASCAL(freq[level,1],D[level+1], minsup[l])

(11) level++;

(12) end

in this algorithm we used PASCAL algorithm for generating all frequent k-itemsets for k>2 at each level *l*. The algorithm PASCAL is given as below:

Algorithm PASCAL: Find frequent itemsets for mining strong association rules in a transaction database.

Input: (1) D, a transaction database, in which each item in the Itemset contains encoded concept hierarchy information, and (2) The minimum support threshold (minsup) for each concept level.

Output: frequent k-patterns.

Method: the Counting Inference method is used for the frequent itemsets generation at each level.

Steps:

(1) $ø.sup=1; ø.key=true;$

(2) $p_0 = \{ø\};$

(3) $p_1 = \{frequent\ 1\text{-}patterns\};$

(4) $for\ all\ p \in P_1\ do\ begin$

(5) $p.pred\_sup=1; p.key=(p.sup \neq 1)$

(6) end

(7) $for\ k= 2; p_{k-1} \neq ø; k++)\ do\ begin$

(8) $C_k = Candidate\_set\_generation\ (p_{k-1})\ ;$

(9) $if\ (c \in C_k\ where\ c.key= true)\ then$

(10) $For\ all\ o \in D\ do\ begin$

(11) $C_o = subset\ (C_k, o);$

(12) $For\ all\ c \in C_o\ where\ c.key=true\ do$

(13) $c.sup++;$

(14) $end;$

(15) $for\ all\ c \in C_k\ do$

(16) $if\ c.sup >= minsup\ then\ begin$

(17) $if\ c.key\ and\ c.sup= c.pred\_sup\ then$

(18) $c.key= false;$

(19) $P_k = p_k\ U\ \{c\};$

(20) $end;$

(21) $end;$

(22) $return\ U_k\ p_k$

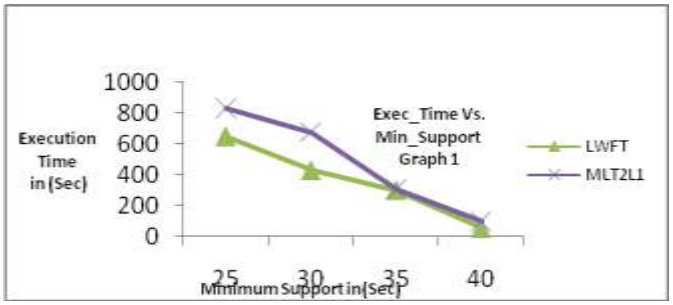
The algorithm starts with the empty set, which always has support 1 and which is a key pattern. Then frequent 1-patterns are determined. They are marked as key patterns unless their support is 1. The main loop is similar to the one in Appriori.In worst case the PASCAL algorithm is worked as Appriori.

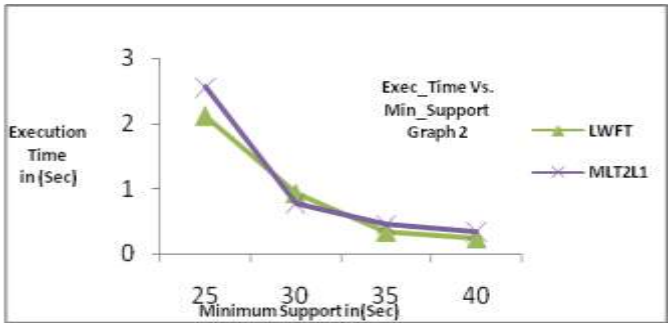
Performance Study

To study the performance of the proposed algorithm as compare to the existing algorithm MLT2 we used two dataset as given below:

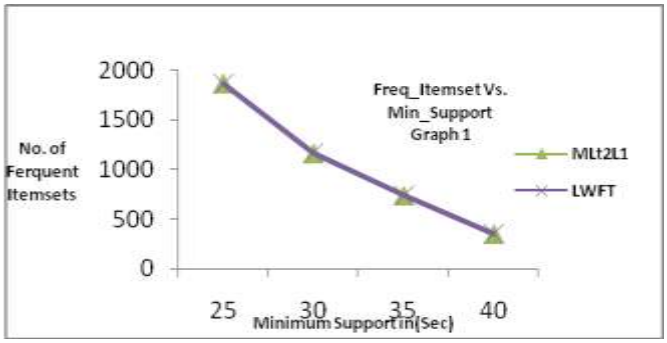
Datasets	No. of Transactions	Size
DB1	100K	2.7MB
DB2	150K	5MB

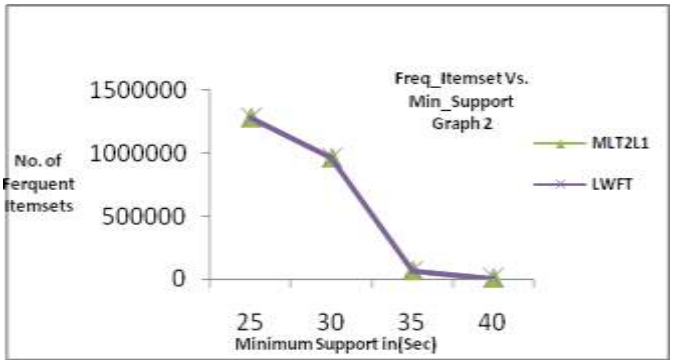
Each transaction, dataset is converted into an encoded transaction table, denoted as T[1], according to the information about the generalized items in the item description (hierarchy) table.
The following are the basic parameters for analyze the of algorithm: (1) the number of frequent itemsets generated (2) The execution time (3) The minimum support threshold and (4) The delta factor.





It is clear from above results that as Min_support decreases the execution time of the algorithm is increase. The execution time of the algorithm is variable for different datasets with a variation in Min_Support. The time for different frequent item set mining algorithms depends a lot on the structure of the data set. The execution time of LWFT algorithm is less than existing algorithm (MLT2L1).





As Min_support decreases at lower levels we find very specific information. The generation of References frequent itemsets at multiple-levels are greater then single levels. The no. of frequent itemsets in LWFT and MLT2L1 are same at similar values of parameters at each level. The mining of multiple-level rules can provide more specific information for the users due to reduced support at lower levels.

4. Conclusions

This study demonstrates that mining multiple-level knowledge is both practical and desirable. This work has successfully discovered multiple-level association rules using LWFT algorithm. The association rules discovered provides more specific information for the users at multiple levels of abstraction. Our algorithm has efficiently discovered Multiple-level association rules from datasets. We have noticed that the execution time of the algorithm depends on the size and complexity of concept hierarchy discovered and hence it is variable for different datasets. This algorithm discovers association rules for successive levels making use of rules already discovered for upper levels of concept hierarchy. Number of association rules discovered depends on value of parameters at each level like support and confidence.

This work is contribution towards representing knowledge at multiple-levels in the form of association rules that enhances the ease and comprehensibility of the users.

Reference

1. Jiawei Han, Micheline Kamber *"Data Mining Concepts and Techniques"* Harcourt India Private Limited ISBN:81-7867-023-2, 2001.
2. R. Agrawal, T. Imielinski, and A. Swami, "*Mining association rules between sets of items in large databases"*. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pages 207-216, Washington, DC, May 26-28 1993.
3. Jiawei Han and Yongjian Fu., *"Discovery of Multiple-Level Association Rules from Large Databases".* Proceeding in IEEE Trans. on Knowledge and Data Eng. Vol. 11 No. 5, pp 798-804, 1999.
4. R. Agrawal and R, Shrikanth, *"Fast Algorithm for Mining Association Rules".* Proceedings Of VLDB conference, pp 487 – 449, Santigo, Chile, 1994.
5. M.H.Margahny and A.A.Mitwaly, "*Fast Algorithm for Mining Association Rules*". Proceedings of AIML 05 Conference, CICC, Cairo, Egypt, 19-21 December 2005.
6. Jiawei Han and Yongjian Fu, *"Discovery of Multiple-Level Association Rules from Large Databases".* Proceedings of the 21st VLDB Conference Zurich, Swizerland, 1995.
7. R. S. Thakur, R. C. Jain and K. R. Pardasani, "*Fast Algorithm for mining multi-level association rules in large databases".* Asian Journal of International Management 1(1):19-26, 2007.
8. Yves Bastide, Rafik Taouil, Nicolas Pasquier, Gerd Stumme and Lotfi Lakhal, "*Mining Frequent Patterns with Counting Inference*". In proceeding of ACM SIGKDD, December 2000, pp68-75.
9. N.Rajkumar, M.R.Karthik, and S.N.Sivanandam, "*Fast algorithm for Mining Multilevel Association Rules*", 0-7803-7651-X/03/$17.00 © 2003 IEEE