Mining Association rules with Dynamic and Collective Support Thresholds

C S Kanimozhi Selvi and A Tamilarasi

Abstract— **Mining association rules is an important task in data mining. It discovers the hidden, interesting relationships (associations) between items in the database based on the user-specified support and confidence thresholds. In order to find relevant associations one has to specify an appropriate support threshold. The support threshold plays an important role in deciding the number of appropriate rules found. The rare associations will not appear if a high threshold is set. Some uninteresting associations may appear if a low threshold is set. This paper proposes an approach to obtain the appropriate support thresholds at each level of the level-wise mining approach. It sets the support threshold by analyzing the frequency of items and their associations in the database at each level. Experimental results show that this approach produces the interesting rules without specifying the user specified support threshold.**

Index Terms— **Association rules, Collective Support, Dynamic Support, Frequent itemset**

I. INTRODUCTION

Data mining is widely used in various application areas such as banking, marketing and retail industry and so on. Association rule mining is one technique used in data mining to discover hidden associations that occur between various data items [1].

An association rule [1] is an expression of the form X à Y, where X, Y are itemsets. It reveals the relationship between the itemsets X and Y. The portion of transactions containing X also containing Y, i.e., P(Y|X) = $P(X \cup Y)/P(X)$ is called the confidence (conf) of the rule. The support (sup) of the rule is the portion of the transactions that contain all items both in X and Y, i.e., sup(X à Y) = $P(XUY)$. To generate an interesting association rule, the support and confidence of the rule should satisfy a user-specified minimum support called MINSUP and minimum confidence called MINCON, respectively.

Mining association rules consists of two steps [1]:
1) Finding all the frequent itemsets having adequate support.
2) Producing association rules from these frequent itemsets.
The outcome depends upon the results of step1.

According to step 1, the itemsets whose support value is greater than the MINSUP are labeled as frequent itemsets. The result of this step plays an important role in producing the association rules. Consequently, the concentration has been given only on the first step by many researchers. However, without specific knowledge, users will have difficulties in setting the MINSUP to obtain their required results. If MINSUP specified by the user is not appropriate, the user may find many meaningless rules or may miss some interesting rules. Hence, the user has to try many possibilities for specifying the MINSUP in order to find the appropriate one.

To overcome these problems, a technique is required to generate the rules of high confidence without having user specified support constraint. To meet out this task, the paper proposes an approach to specify support for frequent itemset generation without consulting the users. This approach finds an initial MINSUP value by analyzing the itemsets and their frequency. It also proposes a collective support threshold on the subsequent levels based on the previous level support and the items considered in the current level.

The rest of the paper is organized as follows: Section II revisits the problem of association rule mining and explores the need for dynamic and collective support thresholds for the generation of association rules. Section III provides a detailed insight into the modified association rule framework and explains the support specification and mining process for this model. Section IV reports the experimental results on the IBM synthetic data upon rule set size. Finally, the conclusions are pointed out in Section V.

II. RELATED WORK

Many researchers considered association rule mining as an interesting research area and studied widely [1-11]. Most of these studies address the issue of finding the association rules that satisfy user-specified minimum support and minimum confidence constraints. Approaches like Apriori [1,12] and FP-Growth [13] employ the uniform minimum support at all levels. These approaches assume that all items in the data are of the same kind and have similar frequencies in the database but this assumption is not applicable for real-life applications. In many applications, some items appear very frequently in the database, while others hardly ever appear. One cannot claim that the frequent itemsets are alone interesting, but the rare items would also matter. To identify the frequent and rare items, an appropriate minimum support has to be specified. Otherwise the user would face two problems [4].

Manuscript received May 4, 2009.

C S Kanimozhiselvi, Assistant Professor, Department of Computer Science and Engineering, Kongu Engineering College, Perundurai 638 052 Erode, Tamilnadu, India Phone: 91-09842168224;

A Tamilarasi, Professor, Department of Computer Science and Engineering, Kongu Engineering College, Perundurai 638 052 Erode, Tamilnadu, India.

9) Generation of fewer rules upon specifying the high minimum support.
10) Generation of too many rules sometimes uninteresting upon specifying low minimum support. This may lead to the development of stronger rule pruning techniques.

The paper [4] argues that the single MINSUP for the whole database is inadequate, because it cannot capture the inherent nature and / or frequency differences of the items in the database. Therefore it suggests a model with multiple minimum supports for items at each level. Even if the user specifies multiple minimum supports, the rules generated may not be more appropriate and interesting.

A better solution lies in developing support constraints, which specify minimum support required for the itemsets, so that only the necessary itemsets are generated. If more than one support constraint is satisfied by an itemset, then the one with minimum support value should be adopted. This has been studied in the paper [6]. This approach also has some problems:

1) This approach deals with only the specific problems but not in general. Moreover it assumes that the user has adequate domain knowledge.
2) Using the approach, one can analyze the nature of the items but can't able to know frequency of the items until all the items in the database are scanned and candidate items are generated.

Correlation based framework [10] without the use of support thresholds has also been studied in the past. It uses contingency table to find positively correlated items. Although the correlation framework discovers strongly correlated items, the support threshold is still important. Without the support threshold, the computation cost will be too high and many ineffectual itemsets would be generated [10]. As such, users still face the problem of appropriate support specification. In [11], the authors avoided the use of support measure to find the interesting associations. This approach is not suitable for applications that adhere to the traditional asymmetric confidence measure as pointed out by [11].

Although substantial effort has been made to lighten these problems, such as adding the lift or conviction measure, which derives the minimum support dynamically from the item support, it also leaves out some interesting rules composed of more than two items because the specification is derived from frequent 2-itemsets [16]. Another effort is made by [14, 15] for finding the N-most frequent itemsets. It allows the user to control the result by specifying the value N, thus leading to user intervention. This paper aims at making the user free from specifying any constraints including support constraints. It proposes an approach to calculate the minimum support threshold dynamically by scanning the records in the database so that all frequent, interesting and meaningful rules would be generated. This minimizes the generation of large number of rules and reduces the need for rule pruning techniques. Experiment results on synthetic data show that the proposed technique is effective.

III. ASSOCIATION RULE MINING FRAMEWORK

Consider a given transaction database T = {r1, r2, …, r*n*}, where each record r*i*, $1 \le i \le n$ is a set of items from a set *I* of items, i.e., r*i* $\subset I$. The basic association rule model as follows:
Let I = {i_1, i_2 …, i_m} be a set of items. Let R be a set of records in the database, where each record r is a set of items such that $r \subseteq I$. An association rule is an expression of the form, X à Y, where $X \subset I, Y \subset I$, and $X \cap Y = f$. The rule X à Y holds in the set of records R with confidence *c* if *c*% of records in *R that* supports *X* also supports *Y*. The rule has *support s* in *R if s*% of the records in *R* contains *X U Y*.

Given a set of records *R*, the problem of mining association rules is to discover all association rules that have support and confidence greater than the user-specified minimum support (*MINSUP*) and minimum confidence (*MINCON*). An association rule mining algorithm works in two steps [3, 4]:

1. Generate frequent itemsets that satisfy MINSUP.
2. Generate interesting association rules that satisfy MINCON using the frequent itemsets.

A. Proposed Model

In this model, we propose two minimum support counts namely Dynamic Minimum Support Count (DMS) and Collective Minimum Support Count (CMS) for the itemset generation at each level. Initially, DMS is calculated while scanning the items in the database. CMS is calculated during the itemset generation. DMS reflects the frequency of items in the database. CMS reflects the intrinsic nature of items in the database by carrying over the existing support to the next level. This model is based on multiple minimum supports model. In each pass, a different minimum support value is used (i.e) the DMS and CMS values are calculated in each pass. Initially, the DMS is used for itemset generation and in the subsequent passes CMS values are used to find the frequent itemsets.

Let there are n items in the database and sup_p be the support of each item in the database and p represents the current pass. MAXS_p and MINS_p denote the Maximum Support and Minimum Support respectively. The total support of items considered in each pass is TOTOCC_p.

$$TOTOCC_p = \sum_{i=1}^{n} \mathrm{sup}_p$$

$$DMS_p = \frac{1}{2}\left[\frac{TOTOCC_p}{n} + \frac{MINS_p + MAXS_p}{2}\right]$$

$$CMS_p = \frac{1}{4}\left[DMS_p + DMS_{p-1}\right]$$

The calculation for the DMS values has been the same at all level. Here DMS_p represents the value at the current level whereas the DMS_{p-1} represents the value at the previous level.

B. Frequent Itemset Generation

The proposed algorithm extends the Apriori algorithm for finding large itemsets. We call the new algorithm, DCS (Dynamic Collective Support) Apriori. The new algorithm is also based on level wise search. It generates all large itemsets by making multiple passes over the data. In each pass p, it counts the supports of itemsets and finds the $MINS_p$ and $MAXS_p$ values, $TOTOCC_p$ and thus the DMS_p value. Initially, the itemsets that satisfy the DMS_p value are retrieved. The DMS_p value is calculated based on the candidates generated in the previous pass.

Let L_k denote the list of k-itemsets. Each itemset c is of the following form, $\{c[1], c[2], \ldots, c[k]\}$, which consists of items, $c[1], c[2], \ldots, c[k]$. The algorithm is given below:

```
Algorithm DCSApriori
Input : Set of records R in Transaction database T
Output : Frequent itemsets
$L_1$= find frequent 1 itemsets(R)
for (k = 2; $L_{k-1} \neq Ø$, k++) do
   $C_k$=candidate-gen ($L_{k-1}$)
end
for each Record r in Є R do
   $C_t$ = subset ($C_k$, r);
   for each candidate c Є Ct do c.count++;
   end
end
   $CMS_p$ = sup_calc($C_k$)
   $L_k$ = {c Є $C_k$ | c.count ≥( $CMS_p$)}
   return $U_k$ $L_k$;

Procedure Candidate-gen($L_{k-1)}$
for each itemset $l_1$ Є $L_{k-1}$
   for each itemset $l_2$ Є $L_{k-1}$
       perform join operation l1   $l_2$
        if has_infrequent_subset(c, $L_{k-1)}$
             prune c;
       else
            add c to $C_{k;}$
       end if
   end
 end
return $C_{k;}$

Procedure has_infrequent_subset(c, $L_{k-1)}$
for each (k-1) subset s of c
   if s is in $L_{k-1}$
    return false;
  else
   return true;
   end if
end

Procedure sup_calc($C_k$)
$MINS_p$ = c.count | c.count is minimum for all $C_k$
                            and c.count > 1
$MAXS_p$ = c.count | c.count is maximum for all $C_k$
$TOTOCC_p$= sum(c.count) for all $C_k$
$DMS_p$ = Average( (Average ($MAXS_p$,
$MINS_p$)+Average($TOTOCC_p$))
If (p= 1) then
$CMS_{p =}$ $DMS_p$
Else
$CMS_{p =}$ ($DMS_{p-1}$ + $DMS_p$) / 4
End if
return $CMS_p$
```

Example: Consider the following dataset.

TABLE I: TRANSACTION DATABASE

R1 : I_1,I_2,I_5
R2 : I_1,I_3
R3 : I_3,I_4
R4 : I_1, I_2, I_3
R5: $I_1,I_2,I_3,I_{5,}I_6$
R6: I_2,I_5,I_6
R7 : I_2,I_5,I_6,I_7

Initially the database is scanned and the support counts of itemsets are found. If there are n items and the minimum support and maximum support are MINS and MAXS respectively. The sum of support of all items is known as TOTOCC. In the first pass, the algorithm retrieves the itemsets whose support count is greater than the DMS value as frequent itemsets. In the subsequent passes, the algorithm uses the CMS value to prune the uninteresting items.

Using the formula 1 and 2, the proposed algorithm finds out the frequent itemsets in the first pass as: I_1, I_2, $I_{3,}$ I_5. Similarly during the second pass the DMS and the CMS values are calculated and the itemsets $\{I_1, I_2\}$,$\{I_1, I_3\}$, $\{I_1, I_5\}$,{$\{I_2, I_3\}$, $\{I_2,I_6\}$,$\{I_3, I_6\}$ are generated. During the third pass the itemsets $\{I_2, I_3, I_5\}$, $\{I_1, I_2, I_3\}$, $\{I_1, I_2, I_5\}$, $\{I_1, I_3, I_5\}$ are generated as frequent itemsets.

C. Rule Generation

The proposed algorithm adopts the confidence based rule generation model of apriori [1] for rule generation. The confidence threshold can be used to find out the interesting rule set. The confidence of a rule is its support divided by the support of its antecedent. For example, the following rule { I_2, I_3} **à** { I_5} has confidence equivalent to support for { I_2, I_3, I_5} / support for { I_2, I_3}. Association rules are generated as follows. For each frequent itemset fl, all non empty subsets of fl are generated. For every non empty subset s of fl, the rule s **à** (fl-s) is formed, if support-count (fl) / support-count of (s) ≥ Minimum Confidence. After the generation of frequent items, the algorithm checks if it satisfies the MINCON threshold. If their confidence is larger than MINCON then they will be generated as interesting association rules. Otherwise, the rules will be discarded. The algorithm for rule generation is given below:

```
Algorithm: Rule_gen($L_k$, MINCON)
for each frequent itemset $fl \in L_k$ do
for each nonempty subset s of fl do
if c.count(fl)/c.count(s) ≥ MINCON then
output the rule $s \Longrightarrow (fl - s)$;
end
end
```

IV. EXPERIMENTAL RESULTS

In this section, the proposed DCSApriori algorithm obtains the frequent itemsets by finding various Collective Minimum Support (CMS) at each level. Then a minimum confidence is set by the user to find the interesting association rules. We use different minimum confidence thresholds at each run. The Apriori algorithm with the standard uniform support (US, set to 0.5 %,1%,2%,3% and 4) is also evaluated. The evaluation is examined based on the number of rules found. All experiments are performed on an Intel Pentium-IV with 256 MB RAM, running Windows 98.

We use two synthetic data sets generated from IBM synthetic data generator [1]: T10.I4.D100K and T40.I10.D100K. Characteristics of these two data sets are shown in Table II.

TABLE II PROPERTIES OF DATASETS

	T10I4D100K	T40I10D100K
Number of Transactions	100,000	100,000
Number of items	870	942
Minimum item frequency	0.001%	0.005%
Maximum item frequency	7.8%	28.738%
Average item frequency	0.11%	0.10%

As shown in Figs 1,2,3,4 and 5 DCSApriori generates not less rules or not more rules while varying the confidence levels. It retains the medium level where as apriori produces number of less confidence rules and at high confidence it produces fewer rules. This means that the ratio of spurious frequent itemsets increases when the MINCON becomes higher, and more uninteresting rules will be generated.
The advantage of DCSApriori is that it sets the appropriate minsup at each level based on the frequency and association of items. It explores more hidden but effective frequent itemsets and avoids the generation of spurious frequent itemsets and thus the low confidence rules.

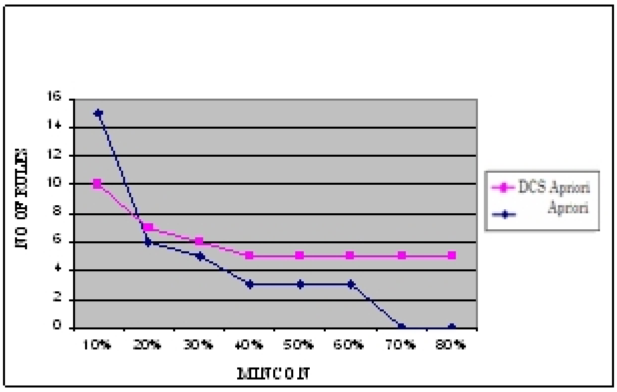

Fig 1. Apriori Vs DCSApriori with Support=0.5% for dataset T10I4D100K

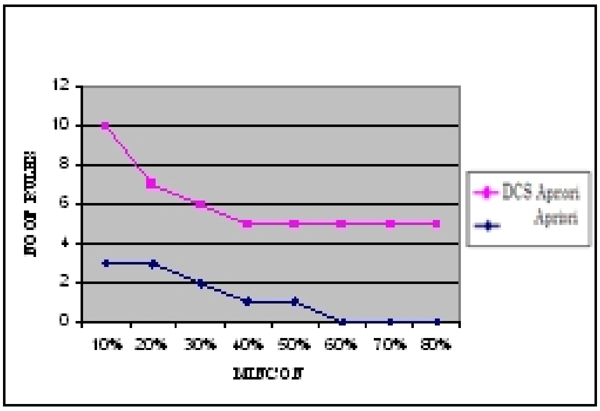

Fig 2 Apriori Vs DCSApriori with Support=1.0% for dataset T10I4D100K

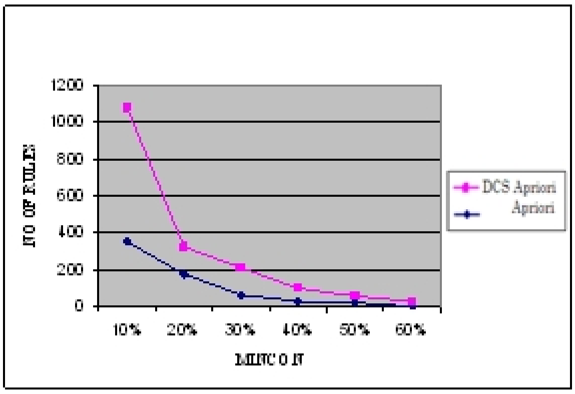

Fig 3 Apriori Vs DCSApriori with Support=1.0% for dataset T40I10D100K

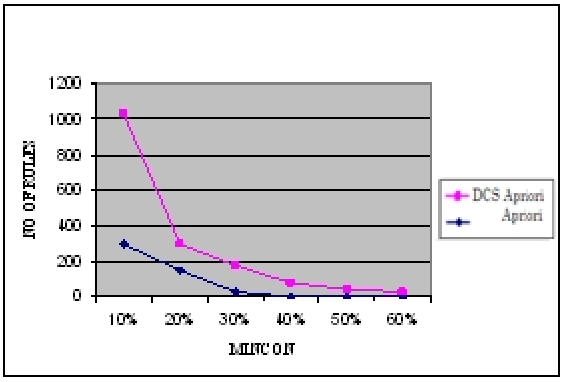

Fig 4 Apriori Vs DCSApriori with Support=3.0% for dataset T40I10D100K

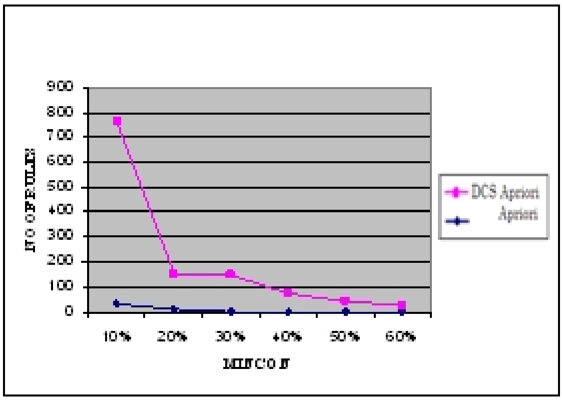

Fig 5 Apriori Vs DCSApriori with Support=4.0% for dataset T40I10D100K

V. CONCLUSION

In the proposed method the minimum support is calculated dynamically. Instead of using the user specified minimum support we use the calculated minimum support for each itemset generation and for rule generation. Thus it generates more relevant and meaningful rules. In fact the running time of the proposed method is not taken into account, it still leaves the user free from specifying minimum support and guarantees the generation of interesting association rules.

REFERENCES

[1] R. Agrawal, T. Imielinski, and A. Swami, “Mining association rules between sets of items in large databases,” Proceedings of 1993 ACM-SIGMOD International Conference on Management of Data, 1993, pp. 207-216.

[2] S. Brin, R. Motwani, J.D. Ullman, and S. Tsur, “Dynamic itemset counting and implication rules for marketbasket data,” Proceedings of 1997 ACM-SIGMOD International Conference on Management of Data, 1997, pp. 207-216.

[3] J. Han and Y. Fu, “Discovery of multiple-level association rules from large databases,” Proceedings of the21st International Conference on Very Large Data Base 1995, pp. 420-431.

[4] B. Liu, W. Hsu, and Y. Ma, “Mining association rules with multiple minimum supports,” Proceedings of 1999 ACM-SIGKDD International Conference on Knowledge Discovery and Data Mining 1999, pp. 337-341.

[5] M.C. Tseng and W.Y. Lin, “Mining generalized association rules with multiple minimum supports,” Proceedings of International Conference on Data Warehousing and Knowledge Discovery 2001, pp. 11-20.

[6] K. Wang, Y. He, and J. Han, “Mining frequent itemsets using support constraints,” Proceedings of the 26th International Conference on Very Large Data Bases 2000, pp. 43-52.

[7] M. Seno and G. Karypis, LPMiner: “An algorithm for finding frequent itemsets using length-decreasing support constraint,” Proceedings of the 1st IEEE International Conference on Data Mining 2001.

[8] J. Li and X. Zhang, “Efficient mining of high confidence association rules without support thresholds,” Proceedings of the 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases 1999.

[9] C.C. Aggarwal and P.S. Yu, “A new framework for itemset generation,” Proceedings of the 17th ACM Symposium on Principles of Database Systems 1998, pp. 18-24.

[10] S. Brin, R. Motwani, and C. Silverstein, “Beyond market baskets: generalizing association rules to correlations,” Proceedings of 1997 ACM-SIGMOD International Conference on Management of Data 1997, pp. 265-276

[11] E. Cohen, M. Datar, S. Fujiwara, A. Gionis, P. Indyk, R. Motwani, J.D. Ullman, and C. Yang, “Finding interesting associations without support pruning,” Proceedings of IEEE International Conference on Data Engineering 2000, pp. 489-499.

[12] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” Proceedings of the 20th International Conference on Very Large Data Bases 1994, pp. 487-499.

[13] J. Han, J. Pei, and Y. Yin., “Mining Frequent Patterns without Candidate Generation,” Proc. 2000 ACM-SIGMOD Int. Conf. on Management of Data (SIGMOD'00),Dallas, TX, May 2000.

[14] Ngan, S-C., Lam, T.,Wong, R.C-W. and Fu, A.W-C. (2005), “Mining N-most interesting itemsets without support threshold by the COFI-tree,” Vol. 1, No. 1, Int. J. Business Intelligence and Data Mining, pp.88–106.

[15] Y. Cheung and A. Fu. “Mining frequent itemsets without support threshold: with and without item constraints,” IEEE Trans. on Knowledge and Data Engineering, 16(9):1052–1069, 2004.

[16] Wen-Yang Lin and Ming-Cheng Tseng,, “Automated support specification for efficient mining of interesting association rules,” Journal of Information Science,Volume 32, Issue 3 (June 2006) : 238 - 250, 2006,ISSN:0165-5515