A Novel Approach to Mine Frequent Item sets of Process Models for Dyeing Process using Association Rule Mining

Harendra Singh
Deptt. Of Computer Science & Engg.
NRI Institute of Science and Technology, Bhopal RGPV Technical University Bhopal, INDIA
Sitendra Tamrakar
Asst Prof. Deptt. Of Computer Science & Engg
NRI Institute of Science and Technology, Bhopal RGPV Technical University Bhopal, INDIA

ABSTRACT:

A noval approach of process mining provides a new means to improve processes in a variety of application domains these process mining techniques help organizations to uncover their actual business processes in this presented paper. Dyeing process using apriori algorithm. but apriori algorithm has some drawback like more execution time, can not handle the large amount of data Now, we propose modified apriori algorithm that applied in dyeing process model to generate frequent pattern and remove the drawback of apriori algorithm in term of execution time, handle the large amount of data. We proposed work Evaluating and analyzing the usefulness and application of the association rule mining algorithms and as it was implemented to obtain simpler process models for the dyeing domain. In view of performance the proposed algorithm for frequent patterns discovery are, it reduces the size of the database after second pass and, the storage space and saves the computing time. this Proposed work has an excellent performance for various kinds of application marketing,medicine,e-commerce,web mining,bio informatics to create frequent patterns, outperforms currently available algorithms in dyeing processing systems, and is highly scalable to mining large databases.

Keyword: *frequent pattern mining dyeing process model, confidence, database, association rule*

I. INTRODUCTION

Data mining is one of the most dynamic emerging research in today's database technology and Artificial Intelligent research; the main aim is to discover valuable patterns from a large collection of data for users. In the transaction database, mining association rule is one of the important research techniques in data mining field. The original problem addressed by association rule mining was to find the correlation among sales of different items from the analysis of a large set of super market data. Right now, association rule mining research work is motivated by an extensive range of application areas, such as banking, manufacturing, health care, medicine, and telecommunications. There are two key issues that need to be addressed when applying association analysis. The first one is that discovering patterns from a large dataset can be computationally expensive, thus efficient algorithms are needed.. We propose a new dynamic algorithm for frequent pattern mining in which database represented in transposed form. And for counting the support we find out by longest common subsequence approach and after finding pattern longest common subsequence is stored or update in database so that next time instead of whole transaction we search from these filter transaction string.. The frequent pattern mining algorithms are applied in the dyeing process due to difficulties of doing the coloring process in an efficient way. Apriori algorithm can significantly reduce mining time by generating pattern candidates that had successfully brought many researchers' attention [1].These frequent patterns have a confidence for different treatments of the dyeing process. These confidences help the dyeing unit expert called dyer to predict better combination or association of treatments. This article also proposes to modify the APriori algorithm to the dyeing process of dyeing unit, which may have a major impact on the coloring process of dyeing industry to process their colors effectively without any dyeing problems, such as pales, dark spots on the colored yarn.

II. RELATED WORK

These process mining techniques help organizations to uncover their actual business processes. Process mining is not limited to process discovery. The frequent pattern mining plays an essential role in many data mining tasks and applications, such as mining association rules, correlations [5], sequential patterns [6], episodes [7], multidimensional patterns [8], max patterns and frequent closed patterns [9], partial periodicity [10], emerging patterns [11], classification [12] and clustering [13]. The numerous studies on the fast mining of frequent patterns can be classified into following categories.

2.1 candidate generation and test approaches,

candidate generation and test approaches such as Apriori and many subsequent studies, are directly based on an anti-monotone Apriori property [1] if a pattern with k items is not frequent, any of its super-patterns with (k + 1) or more items can never be frequent. A candidate generation and test approach iteratively generates a set of candidate patterns of length (k + 1) from a set of frequent patterns of length k ($k \geq 1$), and checks their corresponding occurrence frequencies in the database.

2.2 pattern-growth methods

The second category of methods, pattern-growth methods, such as FPGrowth [14] and Tree Projection have been proposed. A pattern-growth method uses the Apriori property. However, instead of generating candidate sets, it recursively partitions the database into sub-databases according to the frequent patterns found and searches for local frequent patterns to assemble longer global ones. However, these algorithms may still encounter some difficulties in different cases..

2.3 HMine to mine frequent patterns

In 2001 J. Pei [3] proposed an algorithm called HMine to mine frequent patterns efficiently on a sparse dataset. This algorithm utilizes H-Struct data structure, which has very limited and predictable space overhead, and runs very fast in memory setting, hence modified this algorithm with link structure and reverse order processing.

III. CLASSICAL ALGORITHM USED FOR DYING PROCESS

3.1 Sequential Algorithms

3.1.1 AIS Algorithm

The AIS algorithm was the first published algorithm developed to generate all large itemsets in a transaction database [Agrawal1993]. This technique is limited to only one item in the consequent. That is, the association rules are in the form of $X \Rightarrow I_j | \alpha$, where X is a set of items and I_j is a single item in the domain I, and α is the confidence of the rule.

3.1.2 Apriori

the Apriori algorithm developed by [Agrawal1994] is a great achievement in the history of mining association rules [Cheung1996c]. This technique uses the property that any subset of a large item set must be a large itemset.In the first pass, the item sets with only one item are counted. The discovered large item sets of the first pass are used to generate the candidate sets of the second pass using the apriori _gen() function. Once the candidate item sets are found, their supports are counted to discover the large item sets of size two by scanning the database. In the third pass, the large item sets of the second pass are considered as the candidate sets to discover large item sets of this pass. This iterative process terminates when no new large item sets are found. Each pass *i* of the algorithm scans the database once and determines large itemsets of size *i*. L_i denotes large item sets of size *i*, while C_i is candidates of size *i*. The apriori_gen() function as described in [Agrawal1994] has two steps. During the first step, L_{k-1} is joined with itself to obtain C_k. In the second step, apriori_gen() deletes all item sets from the join result, which have some (k-1)–subset that is not in L_{k-1}. Then, it returns the remaining large k-item sets.

Method: apriori_gen() [Agrawal1994]
Input: set of all large (k-1)-itemsets L_{k-1}
Output: A superset of the set of all large k-itemsets //Join step
I_i = Items iinsert into C_k
Select $p.I_1, p.I_2, \ldots\ldots, p.I_{k-1}, q.I_{k-1}$ From L_{k-1} is p, L_{k-1} is q
Where $p.I_1 = q.I_1$ and …… and $p.I_{k-2} = q.I_{k-2}$ and $p.I_{k-1} < q.I_{k-1}$. //pruning step
For all itemsets $c \in C_k$ do
for all (k-1)-subsets s of c do
If ($s \notin L_{k-1}$) then
delete c from C_k

Consider the example given in Table 4 to illustrate the apriori_gen(). Large item sets after the third pass are shown in the first column. Suppose a transaction contains {Apple, Bagel, Chicken, Eggs, DietCoke}. After joining L_3 with itself, C_4 will be {{Apple, Bagel, Chicken, DietCoke}, {Apple, Chicken, DietCoke, Eggs}. The prune step deletes the item set {Apple, Chicken, DietCoke, Eggs} because its subset with 3 items {Apple, DietCoke, Eggs} is not in L_3. The subset() function returns subsets of candidate sets that appear in a transaction. Counting support of candidates is a time-consuming step in the algorithm [Cengiz1997]. To reduce the number of candidates that need to be checked for a given transaction, candidate item sets C_k are stored in a hash tree. A node of the hash tree either contains a leaf node or a hash table (an internal node). The leaf nodes contain the candidate item sets in sorted order. The internal nodes of the tree have hash tables that link to child nodes. Item sets are inserted into the hash tree using a hash function. When an item set is inserted, it is required to start from the root and go down the tree until a leaf is reached. Furthermore, L_k is stored in a hash table to make the pruning step faster [Srikant1996b]
Algorithm 3 shows the Apriori technique. As mentioned earlier, the algorithm proceeds iteratively.

Large Item sets in the third pass (L_3)	Join (L_3, L_3)	Candidate sets of the fourth pass (C_4 after pruning)
{{Apple, Bagel, Chicken}, {Apple, Bagel, DietCoke}, {Apple, Chicken, DietCoke}, {Apple, Chicken, Eggs}, {Bagel, Chicken, DietCoke}}	{{Apple, Bagel, Chicken, DietCoke}, {Apple, Chicken, DietCoke Eggs}}	{{Apple, Bagel, Chicken, DietCoke}}

Table 5.1 :Finding Candidate Sets Using Apriori_gen()

Function count(C: a set of itemsets, D: database)
begin
for each transaction $T \in D = \bigcup D^i$ **do begin**
forall subsets $x \subseteq T$ **do**
if $x \in C$ **then**
x.count++;
end
Algorithm 3. Apriori [Agrawal1994]
Input:
I, D, s
Output:
L

Algorithm:

//Apriori Algorithm proposed by Agrawal R., Srikant, R. [**Agrawal1994**]//procedure LargeItemsets

1) C_1: = I; //Candidate 1-itemsets

2) Generate L_1 by traversing database and counting each occurrence of an attribute in a transaction;

3) **for** (k = 2; $L_{k-1} \neq \phi$; k++) **do begin**//Candidate Itemset generation

//New k-candidate itemsets are generated from (k-1)-large itemsets

4) C_k = apriori-gen(L_{k-1});//Counting support of C_k

5) Count (C_k, D)

6) $L_k = \{c \in C_k \mid c.count \geq minsup\}$

7) **end**

Initially, each item of the itemset is considered as a 1-item candidate itemset. Therefore, C_1 has four 1-item candidate sets which are {Bread}, {Butter}, {Eggs}, and {Milk}. L_1 consists of those 1-itemsets from C_1 with support greater than or equal to 0.4. C_2 is formed by joining L_1 with itself, and deleting any itemsets which have subsets not in L_1. This way, we obtain C_2 as {{Bread Butter}, {Bread Eggs}, {Butter Eggs}}. Counting support of C_2, L_2 is found to be {{Bread Butter}, {Butter Eggs}}. Using apriori_gen(), we do not get any candidate itemsets for the third round. This is because the conditions for joining L_2 with itself are not satisfied.

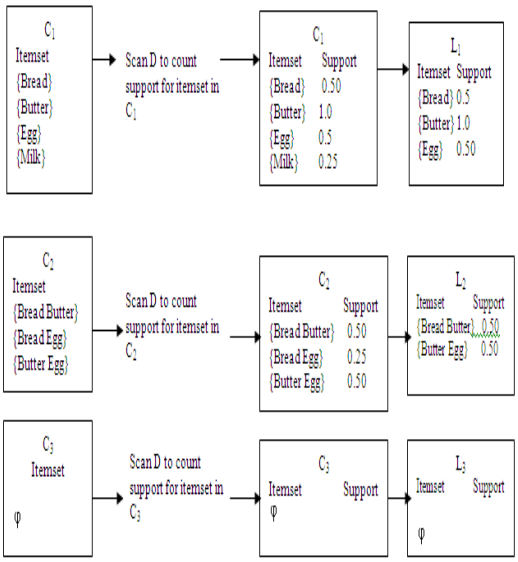

Figure 1 Discovering Large Itemsets using the Apriori Algorithm

4. PROPOSED SCHEME

Association rule mining is a popular and well researched area for discovering interesting relations between variables in large databases. We have to analyze the coloring process of dyeing unit using association rule mining algorithms using frequent patterns. Various algorithms are used for the coloring process of dyeing unit using association rules. For example. LRM, FP Growth Method., H-Mine and Aprori algorithm But these algorithm significantly reduces the size of candidate sets. However, it can suffer from three-nontrivial costs:

(1) Generating a huge number of candidate sets, and
(2) Repeatedly scanning the database and checking the candidates by pattern matching.
(3) It take more time for generate frequent item set.

Apriori algorithm, in spite of being simple and clear, has some limitation. It is costly to handle a huge number of candidate sets. . This is the inherent cost of candidate generation, no matter what implementation Technique is applied. It is tedious to repeatedly scan the database and check a large set of candidates by pattern matching, which is especially true for mining long patterns. Apriori Algorithm Scans the database too many times, When the database storing a large number of data services, the limited memory capacity, the system I/O load, considerable time scanning the database will be a very long time, so efficiency is very low. We have to proposed such that algorithm that it has a very limited and precisely predictable main memory cost and runs very quickly in

memory-based settings. it can be scaled up to very large databases using database partitioning. and to identify the better dyeing process of dyeing unit. In this paper, previous work is based on apriori algorithm in dyeing process of dyeing unit. So basic apriori algorithm are following

4.1 Limitation of Current State of Art.

Current security systems are facing challenges like The drawback of algorithm is

a) More time for execution
b) Less efficient on the larger size datasets
c) Slow and provide low accuracy

4.2 Algorithm Description

The Apriori algorithm had a major problem of multiple scans through the entire data. It required a lot of space and time. The modification in our paper suggests that we do not scan the whole database to count the support for every attribute. This is possible by keeping the count of minimum support and then comparing it with the support of every attribute. The support of an attribute is counted only till the time it reaches the minimum support value. Beyond that the support for an attribute need not be known. This provision is possible by using a variable named flag in the algorithm. As soon as flag changes its value, the loop is broken and the value for support is noted. In proposed algorithm, to calculate the support, we count the common transaction that contains in each element's of candidate set, with the help of the intersect query. In this approach, we have applied a constraints that we will consider only those transaction that contain at least k items,not less than k in process of support counting for candidate set of k length. This approach requires the proposed algorithm improvement mainly concentrated on (1) for reducing frequent itemset and (2) for reducing storage space as well as the computing time. In the case of large datasets like Wal-Mart datasets, the proposed algorithm is very much useful for reducing the frequent patterns and also reducing the database size for every subsequent passes. For example, in the improved algorithm, the number of occurrence of frequent k-itemsets when k-itemsets are generated from (k-1)-itemsets is computed. If k is greater than the size of the database D, there is no need to scan database D which is generated by (k-1)-itemsets according to the Aprior property and it can be remove automatically. Very less time as compared to all other approaches.

4.2.1 PROPOSED ALGORITHM

Algorithm 1: *Improved Algorithm*

Input: A transposed database D^T and the user defined minimum support threshold s.

Output: The complete set of frequent patterns

Step 1: Convert Database D into transpose form D^T

Step 2: Compute CT_1 candidate transaction sets of size-1 and finds the support count.

Step 3:Compute the large transaction sets (LT) of size-1.

(i.e., for all $CT1$ is greater than or equal to minimum support.)

LT_1 = {Large 1-transaction set (LT)};

For (k=2; LT_{k-1} = 0; k++) do

Begin

$CTk = Apriori\text{-}gen(LT_{k-1}, ct)$;

//new candidate transaction sets

End

Return $LT = \cup_k LT_k$;

Algorithm2:*Apriori-gen*($LTk-1$),Generate candidate sets

For all transactions p $\in LT_{k-1}$ do begin

For all transactions q $\in LT_{k-1}$ do begin

If p.transaction1 = q.transaction,..., p.transactionk-2=q.transactionk-2, p.transactionk-1< q.transactionk-1 then

Begin

ct=p ∞ q;

If $has\_infrequent\_subset(ct, LT_{k-1})$ then

delete ct;

Else
For all transaction set $t \in D^T$ do begin
If count(t) <k then delete t;
Else begin
$Ct=subset(CT_k, t)$;
End; End

For all candidate transactions $ct \in CT_i$ do begin
$CT.count = CT.count + 1$;
End; End;
$LTk = \{ct \in CT_k \mid CT.count \geq s\}$;
End; End;
End; End;
Return CT_k;

Algorithms3has_infrequent_subset(ct, LT_{k-1}) //checking the elements of candidate generation
For all (k-1)-sub transaction set of ct do
Begin
If $t \in LT_{k-1}$ then return true;
else return false;
End.

The main advantage of the proposed algorithm for frequent patterns discovery are, it reduces the size of the database after second pass and, the storage space and saves the computing time.

IV. PERFORMANCE ISSUE OF APRIORI ALGORITHM & MODIFIED APRIORI ALGORITHM

To evaluate the efficiency and effectiveness of the improved algorithm, we performed an extensive study of two algorithms: Apriori-like and improved algorithm, on both real time synthetic data sets with different ranges Now, we compare the association rules mining algorithms on the whole data set with 5000 data set Now we implement the association rules that Apriori algorithm is more efficient which takes less time, less memory and hence results in high efficiency The experimental results shows improvement in generation of candidate sets, results in reduced number of data base scan, and also the time and space consumption. we calculate support and confidence
Support= (XUY).count/n
Confidence= (XUY).count/X.count

Database Size	Apriori Algorithm Time(ms)	Modified apriori algorithm Time(ms)
200	400	300
400	550	410
600	630	520
800	690	540
1000	750	590

Table6.6 Support and Confidence value of modified Apriori algorithm

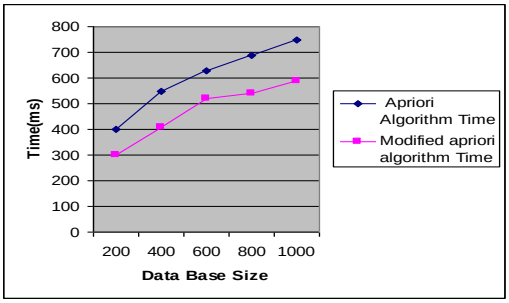

Fig 7.3: Support and Confidence value of modified Apriori algorithm

V. CONCLUSION

Frequent pattern mining is the first step for association rule mining. Association rule mining has found many applications other than market basket analysis, including applications in marketing, customer segmentation, medicine, e-commerce, classification, clustering, web mining, bioinformatics and finance. Various techniques have been found to mine frequent patterns.. It takes less memory by representing large database in compact tree-structure. But a word of caution here that association rules should not be used directly for prediction without further analysis or domain knowledge. They are, however, a helpful starting point for further exploration & understanding of data. Experimental results have shown advantages of Primitive Association Rule Mining over modified Apriori. There are a number of future research directions based on the work presented in this paper. Using constraints can further reduce the size of item sets generated and improve mining efficiency. This scheme was applied in retailer industry application, trying other industry is an interesting field for future work. This scheme use Maximal Apriori and FP-Tree. We can use other combination to improve this approach.

REFERENCE

[1.] A new approach for sensitive association rules hiding by Mohammad Naderi ehkordi, Kambiz Badie, Ahma Khade Zadeh in of International Journal Rapid Manufacturing 2009 - Vol. 1, No.2 pp. 128 – 149

[2.] Privacy Preserving Fuzzy Association Rules Hiding in Quantative Data by Manoj Gupta and R C Joshi in International Journal of Computer Theory and Engineering, Vol. 1, No. 4, October, 2009

[3.] Agarwal CC. and Yu PS., "Privacy-preserving data mining: Modeland Algorithms, (editors)CharuC.Aggarwal and Philip S. Yu, ISBN: 0-387- 70991-8, 2008

[4.] Vassilios S. Verykios,, Ahmed K. Elmagarmid , Elina Bertino, Yucel Saygin, Elena Dasseni. "Association Rule Hiding", IEEE Transactions on knowledge and data engineering, Vol.6, NO.4, April 2004

[5.] Shyue-Liang Wang, Yu-Huei Lee, Billis S., Jafari, A. "Hiding sensitive items in privacy preserving association rule mining", IEEE International Conference on Systems, Man and Cybernetics, Volume 4, 10-13 Oct. 2004 Page(s): 3239 – 3244

[6.] Brin, S., Motwani, R. and Silverstein, C. "Beyond market basket: Generalizing association rules to correlations", in the Proceedings of the 1997 ACM-SIGMOD International Conference on the Management of Data (SIGMOD'97), Tucson, AZ, 1997, pp. 265–276.

[7.] Agrawal, R. and Srikant, R. "Mining sequential patterns", in the Proceedings of the 1995 International Conference on the Data Engineering (ICDE'95), 1995. pp. 3–14.

[8.] Mannila, H., Toivonen, H. and Verkamo, A.I. Discovery of frequent episodes in event sequences. Data Mining and Knowledge Discovery, 1997, pp. 259–289.

[9.] Kamber, M., Han, J. and Chiang, J.Y. "Meta rule-guided mining of multidimensional association rules using data cubes", in the Proceedings of the 1997 International Conference on Knowledge Discovery and Data Mining (KDD'97), Newport Beach, CA, 1997, pp. 207–210.

[10.] Bayardo, R.J. "Efficiently mining long patterns from databases", in the Proceedings of the 1998 ACM- SIGMOD International Conference on the Management of Data (SIGMOD'98), Seattle, WA, 1998, pp. 85–93.

[11.] Han, J., Dong, G. and Yin, Y. Efficient mining of partial periodic patterns in time series database, in Proceedings of the 1999 International Conference on Data Engineering (ICDE'99), Sydney, Australia, 1999, pp. 106–115.

[12.] Dong, G. and Li, J. "Efficient mining of emerging patterns: discovering trends and differences", in Proceedings of the 1999 International Conference on Knowledge Discovery and Data Mining (KDD'99), San Diego, CA, 1999, pp. 43–52.

[13.] Liu, B., Hsu, W. and Ma, Y. "Integrating classification and association rule mining", in the Proceedings of the 1998 International Conference on Knowledge Discovery and Data Mining (KDD'98), 1998, New York, NY, pp. 80–86.

[14.] Agrawal, R., Gehrke, J., Gunopulos, D. and Raghavan, P. "Automatic subspace clustering of high dimensional data for data mining applications", in the Proceedings of the 1998 ACM-SIGMOD