RESEARCH ARTICLE OPEN ACCESS

A Study on Partition and Border Algorithms

T.Karpagam [1], Mrs. S. Rajathi [2]
M.phil scholar [1], Assistant Professor [2]
Department of Computer Science
M.V.Muthiah Government Arts College for Women, Dindigul
Tamil Nadu -India

ABSTRACT
Data mining is the process of extracting useful information from the huge amount of data stored in the database. Data mining tools and techniques help to predict business trends those can occur in near future. Data mining is the procedure of mining knowledge from data. The information or knowledge extracted can be used for any of the following applications such as Market analysis, Fraud detection, Customer retention, Production control, Science exploration. Database mining deals with the data stored in database management systems. Association rule mining is an important technique to discover hidden relationships among items in the transaction. The aim of this paper is to experimentally evaluate an association rule mining approaches, the partition and the border algorithm. The partition algorithm is divided into two phases. In the first phase, the database is divided into number of non overlapping partitions and then the frequent itemsets local to partition are generated for each partition. The database is scanned completely for the first time. Then in the second phase, local frequent itemsets from each partition are combined to generate global candidate itemsets. Again the database is scanned second time to generate global frequent itemsets. The border algorithm maintains support counters for all frequent sets and all border sets. And then get a promoted border (more precisely, when a border set is promoted to a frequent set), an additional pass over the database is made. If there is no promoted border, then the algorithm does not require even a single pass over the whole database. The partition algorithm produces frequent itemset whereas the border algorithm produces promoted border itemsets. The dataset used in this work is the vegetable dataset. The results of both algorithms are compared and analysed that in the border algorithm, the infrequent itemset becomes frequent itemset.

Keyword: - association rule mining, database, frequent itemset, partition

I. INTRODUCTION

1.1 *Data mining*

Data mining is the technique of automatic finding of hidden patterns and information elicitation from huge volume of raw data stored in data bases, data warehouses and other data repositories for making better business decisions, finding sales trends, in developing smarter marketing campaigns, and to predict customer loyalty.

The extraction of the hidden predictive information from large databases is a powerful new technology with great potential to analyze important information in the data warehouse. Data mining scours databases for hidden patterns, finding predictive information that experts may miss, as it goes beyond their expectations. When implemented on a high performance client/server or parallel processing computers, data mining tools can analyze massive databases to deliver answers to questions such as which clients are most likely to respond to the next promotional mailing. There is an increasing desire to use this new technology in the new application domain, and a growing perception that these large passive database can be made into useful actionable information [3].

1.2 KDD vs. Data Mining

Knowledge Discovery in Database (KDD) was formalized in 1989, with reference to the general concept of being broad and high level in the pursuit of seeking knowledge form data. The term data mining was then coined; this high-level application technique is used to present and analyze data for decision-markets.

Data mining is only one of the many steps involved in knowledge discovery in databases. The various steps in the knowledge discovery process include data selection, data cleaning and preprocessing, data transformation and reduction, data mining algorithm selection and finally the post processing and the interpretation of the discovered

knowledge. The KDD process tends to be highly iterative and interactive[4].

Data mining is a step in the KDD process concerned with the algorithmic means by which patterns or structures are enumerated form the data under acceptable computational efficiency limitations. The structures that are the outcome of the data mining process must meet certain conditions so that these can be considered as knowledge. These conditions are

- Validity
- Understandability
- Utility
- Novelty
- Interestingness

1.2.1 Stages of KDD

The stages of KDD, starting with the raw data and finishing with the extracted knowledge, are given below.

- **Selection:** this stage is concerned with selecting or segmenting the data that are relevant to some criteria.
- **Preprocessing:** Preprocessing is the data cleaning stage where unnecessary information is removed.
- **Transformation:** The data is not merely transferred across, but transformed in order to be suitable for the task of data mining. In this stage, the data is made usable and navigable.
- **Data mining:** This is concerned with the extraction of patterns from the data.
- **Interpretation and Evaluation:** The patterns obtained in the data mining stage are converted into knowledge, which in turn, is used to support decision-making

1.3 Association Rule Mining

Association rule mining is an interesting data mining technique that is used to find out interesting patterns or associations among the data items stored in database. Support and confidence are user supplied parameters and differ from user to user. Association rule mining is mainly used in market basket analysis or retail data analysis. In market basket analysis different buying habits of customers were identified and analyzed them to find associations among items those are purchased by customers. Items that are frequently purchased together by customers can be identified. Association analysis is used to help retailers to plan different types of marketing, item placement and inventory management strategies[2].

Terminology and concepts:

X∩Y= φ, Support and Confidence are two measures of rule interestingness.

The rule $X \rightarrow Y$ holds in the database D with support s, where is the percentage of transactions in d that contain X U Y. The rule has confidence c if c is the percentage of transactions in D containing X which also contains I.e.

- **Support($X \rightarrow Y$) = P(XUY)**
- **Confidence($X \rightarrow Y$)=P(Y|X)**

The rules that satisfy both the user specified minimum support and confidence are said to the Strong Assocition rules.

An itemset satisfies minimum support if the occurrence frequency of the itemset is greater than or equal to the product of minimum support and the total number of transations in the entire database. The number of transations required for the itemset to satisfy minimum support is refered as the minimum count. Strong association rules satisfy both minimum support and minimum confidence.

$$\textbf{Confidence}(X \rightarrow Y) = \textbf{P(Y|X)} = \frac{support-count(XU\text{ }Y)}{support-count(X)}$$

Association rules are generated as follows:

- For every frequent itemset x, generate all non empty subset of x
- For every non-empty subset s, of x generate the association rule

$s \rightarrow (x - s)$ **if** $\frac{support-count(XUY)}{support-count(X)}$ is greater or equal to minimum confidence

1.4 Partition Algorithm

Partition algorithm is based on Apriori algorithm, but it requires, only two complete scans over the database [1] as shown in figure 1.

- The database is divided into a number of non overlapping partitions and frequent itemsets local to partition are generated for

each partition. The database is scanned completely for the first time.

- Local frequent itemsets from each partition are combined to generate global candidate itemsets. Then the database is scanned second time to generate global frequent itemsets.

1.5 Border algorithm

The border algorithm is based on the notation of border sets. A set X is a border set if all its proper subsets are frequent sets (i.e., sets with at least minimum support), but it itself is not a frequent set. Thus, the collection of border sets defines the borderline between the frequent sets and non-frequent sets, in the lattice of attribute sets. The border algorithm works by constantly maintaining the count information for all frequent sets and all border sets in the current relation.

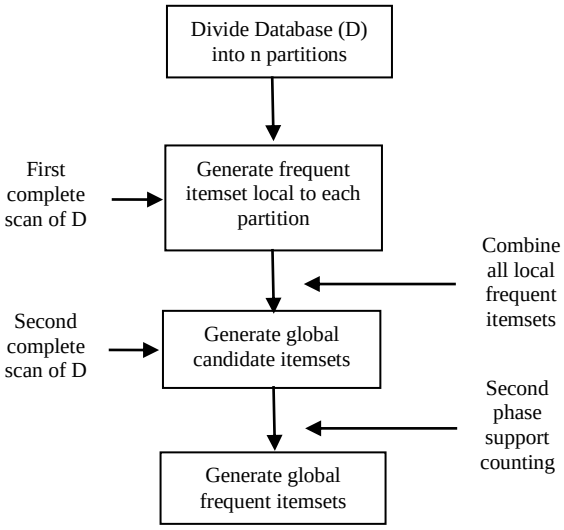

Fig. 1 Partitioning Approach for frequent itemsets mining

Set F contains the itemset of L_{old} becomes a frequent set in T_{whole}. Set B contains all the border sets whose support count reach the level σ and hence, are promoted borer sets. It=f there is no promoted border, and then F contains the entire frequent sets of T_{whole}. But if there is at least one border, then the algorithm generates candidate sets which are supersets of the promoted border sets. It makes one pass over the database to count the support of these candidate sets.

In this paper, several association algorithm has been studied and reviewed the partition and border algorithm in detailed and implemented it in various datasets.

II. RELATED WORK

Rakesh Agarwal et al.[9] initiated mining Association rules between sets of Items in Large Databases. They were given a large database of customer transactions. They presented an efficient algorithm that generated all significant association rules between items in the database. The algorithm incorporated buffer management and novel estimation and pruning techniques. They also presented results of applying that algorithm to sales data obtained from a large retailing company, which showed the effectiveness of the algorithm.

Rakesh Agarwal and Ramakrishnan Srikant [5] proposed the problem of mining sequential patterns over databases. They presented three algorithms to solve that problem, and empirically evaluate their performance using synthetic data. Two of the proposed algorithms Apriorisome and AprioriAll, have comparable performance, albeit AprioriSome performs a little when the minimum number of customers that must support a sequential pattern is low.

J. H. Chang et al.[7] proposed a data mining method for finding recent frequent itemsets adaptively over an online data strem. The effect of old transactions on the mining result of the data stem is diminished by decaying the old occurrences of each itemset as time goes by. Furthermore, several optimization techniques are devised to minimize processing time as well as main memory usage. Finally, the proposed method is analyzed by a series of experiments..

Dr. Kamal Ali Albashiri[8] described the problem of partitioning a compressed set enumeration tree data structure is used together with an associated ARM algorithm. The aim of scenario is to demonstrate that the Multi-Agent Data Mining (MADM) vision is capable of exploiting the benefits of parallel computing; particularly parallel query processing and parallel data accessing.

Neelima et al.[6] presented association rule can be defined as $\{X, Y\} => \{Z\}$. In retail stores if customer buys X,Y he is likely to by Z. It is defined

as to find out association rules that satisfy the predefined minimum support and confidence from a give database. If an itemset is said to be frequent, that itemset supports the minimum support and confidence for in this paper.

III. METHODOLOGY

A. Association rule

An association rule is an expression of the form $X \rightarrow Y$, where X and Y are subsets of A and $X \rightarrow Y$ holds with **confidence** τ, is τ% of transactions in D that support X also support y. The $X \rightarrow Y$ has **support** σ in the transaction set T if σ% of transactions in T support $X \cup Y$[2] .

Problem Decomposition

The problem of mining association rules can be decomposed into two sub problems:

- Find all set of items (itemset) whose support is greater than the user specified minimum support,σ. Such itemset are called frequent itemsets.
- Use the frequent itemsets to generate the desired rules.

$$\frac{S(\{A,B,C,D\})}{S(\{A,B\})} \geq \tau$$

Where s(X) is the support of X in T.

Frequent set

Let T be the transaction database and σ be the user specified minimum support. An itemset X∈ A is sai to be frequent itemset in T with respect to σ , if

$$S(X)_T \geq \sigma$$

Down ward Closure Property: Any subset of a frequent set is a frequent set.

Upward Closure Property: Any superset of an infrequent set is an infrequent set.[8]

B. Partition Algorithm

The partition algorithm is based on the observation that the frequent sets are normally very few in number compared to the set of all itemsets. As a result, if the set of transactions is partitioned to smaller segments such that each segment can be accommodated in the main memory, then the set of frequent sets of each of these partitions can be computed. It is assumed that these sets contain reasonably small number of itemsets. Hence, the whole database can be read once, to count the support of the set of all local frequent sets algorithm execute in two phase.

P=Partition_database (T); n=Number of partitions
//Phase I
for i=1 to n do begin
read _in_partition(T_i in P)
L^i =generate all frequent itemsets of T_i using apriopri method in main memory.
end
//merge phase
for (k=2;$L_k^i \neq \emptyset$ i=1,2,…….,n;k++)do begin
c_k^G =U i =1,2,…..,n Li^k
end
//phase II
for i=1 to n do begin
read_in_partition(T_i in P)
for all candidates c ∈ C^G compute s© Ti
end
$L^G = \{c \in C^G | s(c)T_i \geq \sigma\}$
Answer =L^G

C. Border Algorithm

The partition algorithm was proposed by Ronen Feldman in the year 1997. It maintains support counters for frequent sets [6] and all border sets. Thus, when we get a promoted border (when a border is promoted to a frequent set), an additional pass over the database is made. If there is no promoted border, then the algorithm does not require even a single pass over the whole database.

document should be in Times New Roman or Times font. Type 3 fonts must not be used. Other font types may be used if needed for special purposes.

read T_{new} and increment the support count of X for all X ∈ L_{old} U B_{old}
F:={X|X ∈ L_{old} and s(X) $T_{whole} \geq \sigma$
B:={X|X ∈ B_{old} and s(X) $T_{whole} \geq \sigma$
Let m be the size of the largest element in B.
Candidate-generation
For all itemsets l1 ∈ B_{k-1} U C_{k-1} do begin
For all itemsets l1 ∈ B_{k-1} U F_{k-1} U C_{k-1} do begin
if $l_1[1]=l_2[1]$^ $l_2[2]=l_2[2]$^…..^ $l_1[K-1]<l_2[k-1]$ then
c= $l_1[1],l_1[2]$….. $l_1[k-1],l_2[k-1]$
$C_k = C_k$ U {c}
end do
end do

prune C_k for all k:
all subsetsofk-1sizeshouldbepresent in $B_{k-1}UF_{k-1}UC_{k-1}$
k :=k+1
candidate c:=U C_k
read T_{whole} and count the support values of each itemset in C.
new_frequent _sets := F:={X|X∈Cand s(X) $T_{whole} \geq \sigma$}
L_{whole} := F U new_frequent_sets
B_{whole} :=(B_{old}\B) U {X∈C and s(X) $T_{whole} < \sigma$and all its subsets are in L_{whole}}

IV. IMPLEMENTATION

A. *Data set description*

The partition and Border algorithms were used for frequent itemset generation. After the development of these algorithms, it is tested by using the vegetables dataset.

B. *Vegetable data set Description*

The vegetable data set shown in table 1 contains six attributes. First attribute is transation ID, which is unique and the remaining attributes are in Binary values (0 and 1). The value 1 represents that the item is purchased. 0 represents that the item not purchased.

Table 1. Dataset

ID	Carrot	Beans	Potato	Tomato	radish
1	1	0	1	1	1
2	0	1	1	1	1
3	1	1	0	0	0
4	1	0	1	1	0
5	0	1	1	1	0

V. RESULT AND DISCUSSION

In this paper, the vegetable data set is used and applied in both the Partition Algorithm and Border algorithm and the frequent itemsets are generated. The results show that Border algorithm performs well compared with Partition algorithm.

Partition Algorithm

In this process, the Vegetables data set is partitioned into two tables such as T1 and T2.

The minimum support is σ ≥ 20%.

Table 2.Table T1

D	Carrot	Beans	Potato	Tomato	Radish
1	1	0	1	1	1
2	0	1	1	1	1

Here we take the one itemset at level1 and then two itemset take level2 continuously end for levelN

Level1: {{1},{2},{3},{4},{5}}

Level2:{{1,3},{1,4},{1,5},{2,3},{2,4},{2,5},{3,4},{3,5},{4,5}}

Level3:{{1,3,4},{1,3,5},{1,4,5},{2,3,4},{2,3,5},{2,4,5}, {3,4,5}}

Level4: {{1,3,4,5},{2,3,4,5}}

Partition 1 result

T1 Level= Level1 U Level2 U Level3 U Level4
T1Level=
{{1},{2},{3},{4},{5},{1,3},{1,4},{1,5},{2,3},{2,4},{2,5},{3,4},{3,5},{4,5},{1,3,4},{1,3,5},{1,4,5},{2,3,4},{2,3,5},{2,4,5},{3,4,5},{1,3,4,5},{2, 3, 4, 5}}

Table 3. Table T2

ID	Carrot	Beans	Potato	Tomato	Radish
3	1	1	0	0	0
4	1	0	1	1	0
5	0	1	1	1	0

Level 1: {{1},{2},{3},{4}}
Level 2: {{1,2},{1,3},{1,4},{2,3},{2,4},{3,4}}
Level 3: {{1,3,4}{2,3,4}}

Partition 2 result
T2 Level= Level1 U Level2 U Level3 U Level4
T2 Level= {{1},{2},{3}}

Finally the partition result(P1)
P1={1},{2},{3},{4},{5},{1,3},{1,4},{1,5},{2,3},{2,4},{2,5},{3,4},{3,5},{4,5},{1,3,4},{1,3,5},{1,4,5},{2,3,4},{2,3,5},{2,4,5},{3,4,5},{1,3,4,5},{2,3,4,5}}

Border algorithm

Here we use the above dataset called old dataset. The frequent itemset and border set are generated to get the following result.

Step1: In step one read the database to count the support of 1-itemsets. The frequent 1-itemsets and their support count is calculated.

Step2: Then using the Pruning step eliminates the itemsets which are not found to be frequent: i.e Border, the itemsets is less than or equal to the minimum support (the above two steps are using the each level)

- evel 1 determine the Frequent Itemset Generation.
- order 1 determine the InFrequent Itemset Generation.

Level 1: {1, 2, 3, 4} **Border 1:**{5}
Level 2: {{1,2},{1,3},{1,4},{2,3},{2,4},{3,4}}
Border 2: no frequent.
Level 3: {{1,2,3}, {2,3,4}} **Border 3:** {{1,3,4}, {2,3,4}}

Finally Candidate itemset is empty because the itemset are different and hence the algorithm stops, returning the set of frequent sets along with their respective support values as LEVEL OLD and BORDER OLD

Level old result
Level old= Level1 U Level2 U Level3 U Level4
{{1},{2},{3},{4},{1,2},{1,3},{1,4},{2,3},{2,4},{3,4}{1,2,3},{2,3,4}}

Border result
Border old = Border1 U Border2 U Border3 U Border4

Table 4. New data set

ID	Carrot	Beans	Potato	Tomato	Radish
1	1	0	1	1	0
2	0	1	1	1	0
3	1	1	0	0	0
4	1	0	1	1	0
5	0	1	1	1	0
6	1	0	1	1	1
7	1	1	1	1	1
8	0	1	1	1	1

Level new result
{{1},{2},{3},{4},{5},{1,2},{1,3},{1,4},{1,5},{2,3},{2,4},{2,5},{3,4},{3,5},{4,5},{1,3,4},{1,3,5},{1,4,5},{2,3,4},{2,3,5},{2,4,5},{3,4,5},{1,3,4,5},{2,3,4,5}}

Border old result
{{5},{1,2,3},{1,2,4}}
Level whole result
{{1},{2},{3},{4},{5},{1,3},{1,4},{1,5},{2,3},{2,4},{2,5},{3,4},{3,5},{4,5},{1,3,4},{1,3,5},{1,4,5},{2,3,4},{2,3,5},{2,4,5},{3,4,5},{1,3,4,5},{2,3,4,5}}
Border whole result
{{1,2,3},{1,2,4},{1,2,5}}

VI. CONCLUSION

In this paper, the most popular association rule mining algorithms, the partition and the border algorithms are used. The partition algorithm, partition the set of transactions into smaller segments and each segment can be accommodated in the main memory, then the set of frequent sets of each of these partitions is computed by transaction database. The border algorithm is used to generate frequent sets and border sets and also compute the promoted border. Both the algorithms was tested on different data sets and observed that the partition algorithm predicts the frequent itemset and the border algorithm predicts the promoted border itemset.

REFERENCES

[1] Akhilesh Tiwari, Rajendra K.Gupta and Dev Prakash Agarwal, "cluster based partition Approach for Mining Frequent Itemsets", Proceedings of the IJCSNS international Journal of Computer science and Network Securty,Vol.9 No.6,June 2009.

[2] David W. Cheung, S. D. Lee Benjamin Kao, "A general incremental Technique for maintain Discovered Association Rules", Processing of the Fifth International Conference on Database Systems for Advanced Application.

[3] R.Agarwal, T.Imielinski, A.Swami, " Database Mining: A Performance perspective", Journal of IEEE Transactions of Knowledge and Data Engineering, Volume 5, Issue 6, December 1993, Page 914-925.

[4] Gregory Piatetsky – Shapiro " An overview of Knowledge Discovery in Databases: Recent Progress and Challenges" Rough Sets, Fuzzy Sets and Knowledge Discovery, 1994, pp 1-10.

[5] Rakesh agarwal and Ramakrishnan srikant "mining sequential patterns", Proceedings of the International conference on Data Engineering(ICDE) march 1995.

[6] Neelima .S, Satyanarayana. N and Krishnamurthy .P, "A survery on approach es for mining frequent itemset ", IQSR –JCE, volume 16, issue 4, pp 31-34.

[7] J.H chang and W.S Lee, finding recent frequent itemsets adaptively over online data streams", proceedings of the 9th ACM SIGKDD international Conference on Knowledge Discovery and Data mining, PP 487-492, August 2003.

[8] Soumadip Ghosh, Sushanta Biswas, et al., " Mining frequent Itemsets Using Genetic algorithm", In the International Journal of Artificial Intelligence & Applications(IJAIA), vol 1, no.4, oct 2014

[9] Dr. kamal Ali Albashiri "data partitioning and association Rule Mining Using a Multi-Agent System", International journal of Engineering science and innovative technology, Vol. 2, issue 5, September 2013.

[10] Rakesh Agarwal, Taomasz Imielinski and Arun Swami, "mining Association Rules between sets of Items in large Databases", Proceedings of the ACM sigmod International Conference on Management of Data, Washington DC (USA), 1993

[11] V.vijayalakshmi and Dr. A Pethalakshmi, " Mining of frequent itemsets with an Enhanced Apriori Algorithm", In international journal of Computer Applications vol 81-No.4, Nov 2013