An Improvement of FP-Growth Mining Algorithm Using Linked list

San San Maw
Faculty of Computing,
University of Computer
Studies,Mandalay
Mandalay, Myanmar
sansanmaw@ucsm.edu.mm

Abstract

Frequent pattern mining such as association rules, clustering, and classification is one of the most central areas in the data mining research. One of the foremost processes in association rule mining is the discovering of the frequent pattern. To draw on all substantial frequent patterns from the sizable amount of transaction data, various algorithms have been proposed. The proposed research aims to mine frequent patterns from the sizable amount of transaction database by using linked list. In this method, first scanning the database, the count of frequent 1-itemsets is searched using the hash map and for next itemsets, it is stored in the linked list, second scanning the database. The frequent 2-itemsets is generated using hash table and so on. So, the proposed research needs only two scans and this proposed method requires shorter processing time and smaller memory space.

Keywords: *frequent pattern mining, data mining, linked list, hash table*

I. INTRODUCTION

Data mining is to draw forth the applicable information from the sizable database. The association rule mining is one of the substantial matters in the field of data mining. The frequent pattern mining is the core process of association rule mining. The frequent pattern mining, which searches the relationship in a given data set, has been widely employed in various data mining techniques. The mined information should be wide-ranging that is hidden in the data and provides some facts and information that can further be used for management decision making and process control. Several algorithms have been developed for mining frequent patterns that are significant and can provide important information of planning and control.

In frequent pattern mining, it is necessary to consider a dataset, D $\{T_1, T_2, T_3, ..., T_n\}$ and so, it consists of "*n"* transactions. Each transaction T encloses a number of items of the itemsets $I = \{i_1, i_2, i_3, ..., i_m\}$. Each transaction (*TID*, *I*) is combined together with an identifier, called *TID*. The minimum support count, "*min-sup"*, the percentage of transactions in *D*. Assume *A* be a set of items. If $A \subseteq T$, a transaction *T* is said to contain *A*. In the transaction set *D* with support *s*, the rule

$A \Longrightarrow B$ holds it contains $A \cup B$.

Support $(A \Longrightarrow B) = P(A \cup B)$

The rule $A \Longrightarrow B$ has confidence "*c"* in the transaction set *D*, where "*c"* is the percentage of transaction in *D* containing *A* that also contain *B*.

Confidence $(A \Longrightarrow B) = P(B \mid A)$

Association rule mining is essential in data analysis method and data mining technology. R. Srikant proposed the Apriori algorithm, which employs iterative approach and the candidate itemsets generation. If the frequent k-itemsets exist, then it scans the k times fully database. This result is more time consuming and takes more memory space.

FP-Growth algorithm was proposed by Han et al. The set of frequent 1-itemset are collected by scanning the database. And then, FP-Tree, whose structure has only frequent 1-items as nodes, is constructed. And then it stored information about the frequent patterns. This method mines frequent patterns without using the generation of candidate itemsets and the database scans twice. A set of item-prefix subtrees has in the FP-Tree. Each node in the item-prefix subtree contains three fields - item node, count, node-link.

II. RELATED WORK

The various improvements in FP-Growth algorithm have been made by the researchers. Some algorithms are discussed.

In [2], frequent pattern mining using linked list(PML) was presented. Horizontal and vertical data layout are used. For frequent 1-itemset, horizontal data layout is employed. For frequent 2-itemsets and more, vertical data layout is used. Using intersection operations, transaction ids are speedy counted. This is the significant highlight of vertical data layout. When the frequent itemsets are large, the PML method runs faster than other methods (Apriori, FP-Growth and Eclat algorithms).

In [3], the frequent itemset mining using the N-list and subsume concepts (NSFI) was introduced by Vo, B., Le,

T., Coenen, F. and Hong, T.P. The procedure of creating the N-list associated with 1-itemsets is modified and N-list intersection algorithm is improved by using hash table. Moreover, the subsume index of frequent 1-itemset based on N-list concept is determined. This method suggested two theorems. In the light dataset, NSFI did not improve over PrePost method. In the compact dataset, NSFI method is speedier than Prepost method and dEclat method.

In [4], "an improvement of FP-Growth association rule mining algorithm based on adjacency table" was proposed by Yin, M., Wang, W., Liu, Y. and Jiang, D. The items in the adjacency table were stored using hash table. In this algorithm, only one scan is needed for the transaction database, to make the input/output jobs smaller to a certain degree. Particularly, this method proves to have the high performance in the dense transaction.

In the finding of Nadi, F., Hormozi, S.G., Foroozandeh, A. and Shahraki, M.H.N., the transactions elements of database are translated into a square matrix [1]. Then, this matrix is considered as the complete graph: one-to-one correspondence, and maximum complete subgraphs are pulled out as maximal frequent itemsets. This method has fit performance in the sizable database which has particularly small number of unique items compared to the complete number of transactions.

In [5], Zhang, R., Chen, W., Hsu, T.C., Yang, H. and Chung, Y.C presented "A combination of Apriori and graph computing techniques for frequent itemsets mining: ANG". In this method, Apriori method is very efficient when frequent small-itemsets are searched. When frequent large-itemsets are searched, the graph computing method is used. So, using the advantages of two methods, hybrid method was proposed. But, in this method, the accurate switching point is essential.

III. METHODOLOGY

A. The steps of frequent pattern mining using linked list

The proposed method only needs twice to scan the database.

Firstly, the frequency of 1-itemsets is counted scanning the database and removed this 1-itemsets that the support count(Sup-count) is less than minimum support count(min-sup).

Secondly, each of frequent 1-itemsets is stored in hash table as the key.

Thirdly, each of transaction is sorted decreasing order over frequent 1-itemsets and for the next itemsets, it is stored in the linked list.

Finally, the 1-itemsets related the key are counted with sup-count and removed this 1-itemsets whose frequency is less than min-sup and so on.

B. The algorithm for frequent itemsets generation

Algorithm: Find frequent itemsets using linked list

Input:

- *D*, a database of transactions;
- *min-sup*, the minimum support count threshold.

Output: frequent itemsets in *D*.

Method:

(1) Count the number of 1-itemsets from the transaction by scanning the database.

(2) Find the frequent 1-itemsets that satisfy min-sup.

(3) if (frequent 1-itemsets exists),

{ Set frequency = true.
Output frequent 1-itemsets.
Go to step 4.}
else
{ Set frequency = false.
Output "No frequent 1-itemsets".
Break.}

(4) Create Transaction Linked List hash table with items found in step-2 as hash table key.

(5) For each transaction of the database (//Scan D)

{-Sort the decreasing order with the items found in step-2.
Search the node related Transaction Linked List hash table key for the following itemsets in the database.
If exists, the frequency of this itemsets increases by one.
Else, create the node of this itemsets and the frequency sets one.
}

(6) Initialize k = 2.

(7) While (frequency)

{
If (k = = 2), then
{
Create 2-Itemsets Linked List hash table using the key in the Transaction Linked List hash table.
Count the frequency of 1-itemsets related the key using Transaction Linked List hash table.
Prune the 1-itemsets whose frequency is less than min-sup and update 2-Itemsets Linked List hash table
}
Else
{
Create k-Itemsets Linked List hash table using the key of k-1 Itemsets Linked List hash table.
Create the node of k-1 itemsets related the key using k-1 Itemsets Linked List hash table and

then count the frequency of this node using Transaction Linked List hash table.

Prune the node of k-1 itemsets that the frequency is less than min-sup and update k-Itemsets Linked List hash table.

}

-if (frequent k-itemsets exists), =

{

Set frequency = true.

Output frequent k-itemsets.

}

else

{

Set frequency = false.

Output “No frequent k-itemsets.”

}

Increases k;

}

As an example, the database shown in table 1 is explained and predefined minimum support count(min-sup) is 3.

Table I. Transaction Database

Transaction	Items
T1	a, d. f
T2	a, c, d, e
T3	b, d
T4	b, c, d
T5	b, c
T6	a, b, d
T7	b, d, e
T8	b, c, e, g
T9	c, d, f
T10	a, b, d

Firstly, the frequency of each item is counted with the support count(Sup-count) by scanning the database shown in table 2.

Table II. Frequency of each item

1-itemset	Sup-count
a	4
b	7
c	5
d	8
e	3
f	2
g	1

And then, the frequent 1-itemset are collected by pruning the 1-itemset whose frequency is less than minimum support count that are shown in table 3.

Table III. Frequent 1-itemset

Frequent 1-itemset	Sup-count
a	4
b	7
c	5
d	8
e	3

Each of frequent 1-itemset is stored as the key in the Transaction Linked List hash table shown in figure:1 and by scanning the database, each transaction of the database is sorted in decreasing order based on frequent 1-itemset shown in table 4 and the count of the next itemsets that are related with the hash table key are stored in the Transaction Linked List hash table.

Table IV. Sorted transaction

Transaction	Items
T1	d, a
T2	e, d, c, a
T3	d, b
T4	d, c, b
T5	c, b
T6	d, b, a
T7	e, d, b
T8	e, c, b
T9	d, c
T10	d, b, a

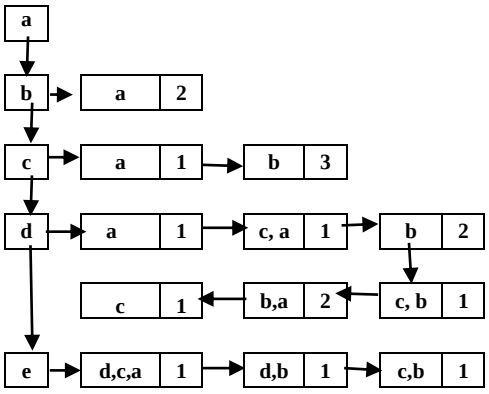

Figure 1. Transaction Linked List hash table

In addition, the 2-Itemsets Linked List is created using the key of the Transaction Linked List and then one

itemsets that are related the key are counted shown in figure:2.

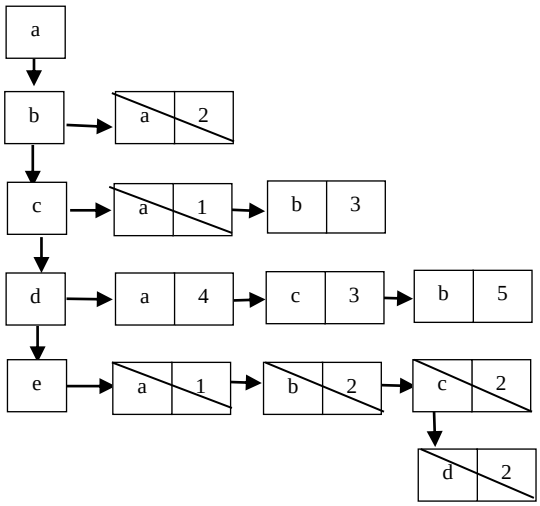

Figure 2. 2-Itemsets Linked List hash table

In accord with this min-sup count, it is necessary to get rid of infrequent items and update 2-Itemsets Linked List shown in figure:3.

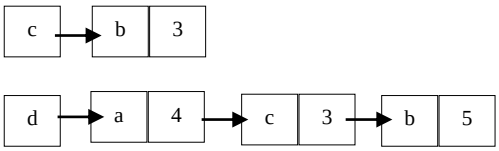

Figure 3. Updated 2-Itemsets Linked List hash table

Table V. Frequent 2-itemsets

Frequent 2-itemset	Sup-count
b, c	3
a, d	4
b, d	5
c, d	3

The frequent 2-itemsets are {b, c: 3, a, d: 4, b, d: 5, c, d: 3}.

And then, it is to create 3-Itemsets Linked List hash table shown in figure:4 using the key of 2-Itemsets Linked List and the two itemsets related the key are counted in the Transaction Linked List and pruned the itemsets whose frequency is less than min-sup count. Similarly, three itemsets are counted in the 4-Itemsets Linked List and pruned and so on.

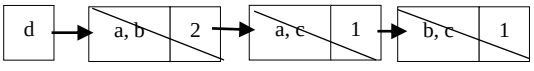

Figure 4. 3-Itemsets Linked List hash table

If there is no frequent itemsets, the process has stopped. In this example, frequent 3-itemsets does not occur because the frequency of next two itemsets is less than min-sup. So, the processing has stopped and frequent 1-itemsets and 2-iemsets are generated as the output.

IV. CONCLUSION

After having studied the mining process of frequent patterns algorithm, this paper proposes an improved FP-Growth method based on linked list. In the proposed algorithm, the database is scanned only twice and this tremendously reduces the input/output operations. The hash table is adopted in this algorithm for the speedy lookup. The proposed method has considerably reduced the running time and used up less memory space. The future work will be able to perform the comparison of the improved FP-Growth method employing linked list and FP-Growth.

ACKNOWLEDGMENT

First and foremost, I would like to thank Dr. Kay Thi Win and Dr. Ingyin Oo for giving me kind supports in doing my research. Then, I would also like to thank the organizers of the ICCA conference and the reviewers for providing me valuable and effective feedback comments in revising my paper.

REFERENCES

[1] Nadi, F., Hormozi, S.G., Foroozandeh, A. and Shahraki, M.H.N., 2014, October. A new method for mining maximal frequent itemsets based on graph theory. In *2014 4th International Conference on Computer and Knowledge Engineering (IEEE)* (pp. 183-188)..

[2] Sandpit, B.S. and Apurva, A.D., 2017, July. Pattern mining using Linked list (PML) mine the frequent patterns from transaction dataset using Linked list data structure. In *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp. 1-6). IEEE.

[3] Vo, B., Le, T., Coenen, F. and Hong, T.P., 2016. Mining frequent itemsets using the N-list and subsume concepts. *International Journal of Machine Learning and Cybernetics*, *7*(2), pp.253-265. Springer

[4] Yin, M., Wang, W., Liu, Y. and Jiang, D., 2018. An improvement of FP-Growth association rule mining algorithm based on adjacency table. In *MATEC Web of Conferences* (Vol. 189, p. 10012). EDP Sciences.

[5] Zhang, R., Chen, W., Hsu, T.C., Yang, H. and Chung, Y.C., 2019. ANG: a combination of Apriori and graph computing techniques for frequent itemsets mining. *The Journal of Supercomputing*, *75*(2), pp.646-661.