INFREQUENT WEIGHTED ITEM SET MINING USING NODE SET BASED ALGORITHM

G.Amlu*1 S.Chandralekha#2 and PraveenKumar*1

* B.Tech, Information Technology, Anand Institute of Higher Technology, Chennai, India
# B.Tech, Information Technology, Anand Institute of Higher Technology, Chennai, India

Abstract--- Mining infrequent weighted itemsets that composed of two algorithms. The algorithms are infrequent weighted Item set Miner (IWI Miner) and Minimal Infrequent Weighted Item set Miner (MIWI). These algorithms are primarily based on Fp-Growth algorithm. Normally Fp-growth algorithms are used to mine frequent itemsets. In order to mine Infrequent Itemsets we have to make some changes in the FP – Growth. The changes are construction of Fp-tree and different pruning strategy that makes two algorithms IWI and MIWI. Normally the frequent itemsets are mined based some threshold values. Here we are also using a threshold that is called IWI support threshold that mines infrequent items. The IWI and MIWI both are based on Fp-Growth. But Fp-Growth consumes more memory and it takes more time to mine relevant itemsets whether it may be frequent or infrequent. So that we are using a concept called Node set based algorithm which consumes considerable amount of memory and reduces the mining time. Based on Node set we present an efficient algorithm called FIN which has high performance and fast mining process.

Index Terms— Frequent and Infrequent Item set mining, Node set, Pruning, Support.

I. INTRODUCTION

Data mining is a process of analysing data from different perspective or angles and summarizing it in order to produce the Information. In order to increase the revenue using that information. The information can also be used for increasing the cuts cost and revenue or cut cost or both. Data mining can also be defines in other terms such that it is a process of finding correlation or pattern among dozens of fields in large relational data bases. Data mining can be used in the fields like retail business in which it is used for analysing retail point of sale transaction data can yield information on which products are selling and when Data mining is used for predicting the customer’s behaviour. So the retailer or manufacturer could determine which items are most susceptible to promotional efforts. Data mining is also known as knowledge discovery. There are generally four types of analysing of data. They are classes, clusters, association, and sequential pattern. Class means stored data is used to locate data in predetermined groups. Clusters means data items are grouped according to logical relationships or consumer preference. Association means data can be mined to identify associations. Sequential pattern mining is used for expecting behaviour pattern and trends. Data mining can be used in variety of applications are sales and marketing, customer retention, risk assessment and fraud detection. In various applications it implements various kinds of techniques. They are link analysis, predictive modelling, database segmentation and deviation detection.

Data mining are very important today. Because changes in business environment. Customer becoming more demanding and markets are saturated. The corporate world is a cut-throat word. It is used to take decisions rapidly. The decisions are made with maximum knowledge. It motivates by adding values to the data holding, and used to take long term and short term decisions. Item set mining is an exploratory data mining technique used identifying correlation among the data. The first attempt in item set mining is to mine the frequent itemsets. A specific threshold is maintained in order to mine the frequent items that are having that amount of frequency. The frequent items are used in number of real time applications. The frequent item set mining is used in market basket analysis, medical image processing, biological data analysis. Many traditional approaches ignore the interest or influence of each item or transaction within the analysed data. So that weight is introduced for treating items or transactions differently. The weight [3] [6] is associated with each item and it indicates the local significance of each item within each transaction.

There are two measures available in association rule mining. They are objective and subjective measures. Objective measures are the most commonly used measures. Support and confidence are the objective measures. Here support means each item’s total number of
occurrence in a transaction set. Mostly the frequent items are mined based on this support count values. Item that has support count above the specified threshold is mined as frequent items. Confidence is another objective measure of association rule mining. That indicates the possible combination item set in each transaction. Consider that X ==> Y, if C % of transaction in T that contains X also contains Y. rules that have a ‘c’ greater than a user specified confidence is said to have minimum confidence. Likewise rules that have a’s’ greater than a user specified support is said to have minimum support. Here’s’ represents number of transaction that contains particular item. The formula for support is represented below

\[
\text{Support of each item} = \frac{\text{occurrence}}{\text{Total support}}
\]

II. EXISTING SYSTEM

In recent years the attention of the research community has been focused on the infrequent weighted item set mining problem. That means discovering itemsets whose occurrence frequency is less than or equal to maximum threshold in the analyzed data. Infrequent item set mining is an intermediary step in many applications. Frequent item set are the itemsets that also having their subsets to be frequent. But infrequent means the item set should not keep any infrequent subsets. This is called minimal [7] infrequent itemsets [1]. This type of infrequent mining can be used in fraud detection, risk assessment [4] [7]. The FP-growth algorithm is used in mining of infrequent itemsets. But it is an algorithm that is used for mine frequent itemsets. We cannot apply the FP-growth directly to mine infrequent itemsets. Some modifications need to be made in the FP-growth [10] in order to make frequent itemset mining. These modifications made two algorithms that are called infrequent item set miner (IWI) and Minimal Infrequent item set Miner (MIWI). These modified algorithms are used to get infrequent itemsets. The modifications made in the FP-growth algorithm are generating tree based on itemsets in each transaction and applying IWI support function that may be IWI-support-min or IWI-support-max thresholds. The IWI support threshold is used to prune the tree that was constructed in each transaction. Pruning means cutting down unwanted portions from the tree. Before that we need to apply equivalence property. The weighted transaction equivalence creates an association between a weighted transaction data set T, composed of transaction with arbitrarily weighted items within each transaction and TE is the equivalent data set in which each transaction is exclusively composed of equally weighted items. The equivalence property transformation is used for compact representation of original data set by means of FP-tree index.

<table>
<thead>
<tr>
<th>TID</th>
<th>Original Transaction</th>
<th>Equivalent weighted transaction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Minimum weighting Function</td>
</tr>
<tr>
<td>0</td>
<td>100 57 41</td>
<td></td>
</tr>
<tr>
<td>1.a</td>
<td>0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>1.b</td>
<td>* 57 57 57</td>
<td></td>
</tr>
<tr>
<td>1.c</td>
<td>* 14 * 14</td>
<td></td>
</tr>
<tr>
<td>1.d</td>
<td>* 29 * *</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TID</th>
<th>Original transaction</th>
<th>Equivalent weighted transaction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Maximum weighting Function</td>
</tr>
<tr>
<td>0</td>
<td>100 57 41</td>
<td></td>
</tr>
<tr>
<td>1.a</td>
<td>100 100 100 100</td>
<td></td>
</tr>
<tr>
<td>1.b</td>
<td>-29 * -29 -29</td>
<td></td>
</tr>
<tr>
<td>1.c</td>
<td>-14 * -14 *</td>
<td></td>
</tr>
<tr>
<td>1.d</td>
<td>-57 * * *</td>
<td></td>
</tr>
</tbody>
</table>

‘*’ is used to indicate there is no transaction of particular item. The equivalent data sets only includes equally weighted [3] [6] items. In both the functions the transaction 1 itemsets are reported as equivalent itemsets in the above two table. When using minimum weighting function, the procedure first takes the lowest among the weights occurring in original transaction. And it is represented as \( w_{\text{ref}} \). Using iterative procedure for each transaction an equivalent transaction itemsets with equivalent weights is produced. In maximum weighting function the highest values are taken in to first consideration. And subtraction is used iteratively instead of weight addition for each equivalent transaction within one original transaction. The above procedure is repeated until the ‘S’ is empty. The IWI-support of a weighted itemsets in a weighted transactional data set corresponds to the one evaluated on the equivalent data set. The important thing is IWI-support-max (I, TE) = IWI-support-max (I, T) [1]. The next step is to generating tree for each transaction. The main difference between IWI
miner and MIWI miner is stated in upcoming sentences. The IWI miner outputs the itemsets that are infrequent based on threshold. The MIWI is used to mine smallest infrequent itemsets.

The FP-growth algorithm [8] is modified in order to mine infrequent items. In IWI miner we are checking the support value that is prefixed with each item. We are checking the support value from root to the leaf of the tree. If the support value of the leaf node is greater than the IWI support threshold means the path that contains the leaf node is prunes from the tree. Remaining nodes represents the infrequent items. Before inserting itemsets into tree we arranged the itemsets in descending order based on the support values. Then it is under going to the mining process and conditional pattern base. In MIWI miner algorithm we are not going to consider a path if the leaf node in the path containing support value greater than the specified level (maximum threshold). So it will yield smallest infrequent itemsets. But in IWI we can consider the other nodes in the paths.

Both the IWI miner and MIWI miner are having same steps. Some final procedures are different. They are having two step procedures. First it will take transactions and their weighted data sets. Then it will get threshold input and generates equivalent transaction weighted itemsets. The next step is to mining the infrequent itemsets. For that we are passing generated tree and threshold and prefix of each node in the tree. The output will be mined infrequent weighted itemsets. The MIWI and IWI miner both are using IWI support function it may be the cost functions like IWI-support-min or IWI-support-max. Based on this cost function the IWI support threshold will be determined and used in mining process.

III. PROPOSED SYSTEM

Infrequent weighted Item set Mining using Fp-growth algorithm [2] has many disadvantages. Even though the IWI miner and MIWI miner [1] are used to mine infrequent weighted items the basic for these algorithms are Fp-growth [5]. The Fp-growth algorithm has following disadvantages. It needs two passes to perform the mining process. In first pass it needs to scan the data base. And finds support for each items in the transaction. Then tree is constructed and the common prefixes are shared. In second pass it will scan each transaction one by one and counter is incremented if same items are in the transaction. Based on threshold we are mining the infrequent items. When reading the transaction and their relevant items it uses fixed order. When paths are overlapped means it will share the paths and counter value is incremented. There are two cases. First case is it is having smaller size than data which is uncompressed. If many transactions sharing same items means it will use same path again and again. It is called best case. The second case is worst case. If every transaction items having different items means the corresponding Fp-tree will also get increased. So it requires more storage space for nodes and pointers. There is no assurance for smaller tree if items are ordered by decreasing support. Fp-tree is very costly to construct as it requires more memory and time when size and number of transaction increases.

Instead of Fp-tree approach we proposed IWI miner and MIWI miner algorithms using Node set based [2] algorithm. Normally node set based algorithm is used in networks to find neighbouring nodes in that networks. Continuously the details of each node in the particular path are pinged to each node in the network. So it will reduce the time for checking each node in every transaction. This is the advantage of node set based algorithm. The effective FIN algorithm is used to mine infrequent items. So IWI miner and MIWI miner [1] are formed based on the FIN algorithm. It is same as Fp-growth. In order to make mining process very efficient we are suggesting FIN [2] algorithm. When comparing with Fp-growth, FIN has high performance on both running time and memory usage.

III. A. BASIC DEFINITIONS

a) Pre-Order-Coding tree (POC tree):

It consists [2] of one root node that is called null node and children nodes that contains the set of prefix sub trees. There are five fields in the item prefix of each node sub tree [2]. They are item name, count, children list, pre-order list, item name register contains which items this node represents, and count register contains [2] the number of transaction that is presented by the portion of the path reaching this particular node. Children register represents the children of the node and pre-order register indicates pre-order rank of the node. The POC [2] tree is used until the node sets for
infrequent 2 items are generated. Then it will be useless. So it is deleted.

Frame work of FIN

![Architecture diagram of node set based Infrequent weighted Item set Mining](image)

#### Algorithm for POC tree construction

**Input:** data base and threshold value.

**Output:** POC tree

1. First set of items generation:
   - Scan the data base and based on the threshold sort the support values of item sets in descending order.
2. Initiate the root node as null.
3. For each transaction
4. Sort the items in previously sorted order
5. do
6. Call insert tree ((P | p), tree)
7. If child of tree is n-node
8. Then n. item-name = p. item-name.
9. n-count ++
10. Else create new node n and add it to children list
11. While P is non-empty
12. Pre-order traversal and generate pre-order code by scanning each node.

b). Node set:

Before knowing about node set we have to know about N-info. [2] It is the pair of pre-order and count. In a POC tree node set of item ‘I’ is the sequence of N-info of nodes that is registering ‘I’. The sum of count registers of N-info in the nodes that registers ‘I’ is equal to the support of ‘I’.

c). Super set Equivalence:

The support of P is equal to the support of P U {i} indicates [2] that any transaction that contains P also contains i.

**FIN algorithm:**

**Input:** data base and maximum threshold.

**Output:** F-set of all infrequent items.

1. F= null
2. Call POC construction algorithm
3. F2=null
4. Scan POC in pre-order traversal
5. Do
6. N = currently visited node
7. i_y= item registered in node N
8. For each ancestor of N, N_a, do
9. i_x = the item registered in N_a;
10. If i_x i_y ∈ F2, then
11. i_x i_y.support= i_x i_y.support + N.account;
12. Else
13. i_x i_y.support = N.account;
14. F2= F2 ∪ {i_x i_y};
15. End if
16. End for
17. For each itemset, P, in F2 do
18. If P. support > e × DB, then
19. F2= F2 - {P};
20. Else
21. P. Node set = null;
22. End if
23. End for
24. Scan the POC-tree by the pre-order traversal do
25. N_d currently visiting Node;
26. i_y the item registered in N_d;
27. For each ancestor of N_d, do
28. i_x the item registered in N_{da};
29. If i_x i_y ∈ F2, then
30. i_x i_y.Node set = i_x i_y. Node set U N_d. N-info;
31. End if
32. End for
33. \( F = F \cup F_1; \)
34. For each frequent itemset, \( i_k \) in \( F_2 \) do
35. Create the root of a tree, \( R_{at} \), and label it by \( i_k \);
36. Constructing-Pattern-Tree \( (R_{at}, \{i \mid i \in F_1, i > i_k\}, \text{null}); \)
37. End for
38. Return \( F; \)

Procedure for Constructing-Pattern-Tree \((N_d, \text{Cad-set, FIS-parent})\)
1. \( N_d, \text{equivalent-items} = \text{null}; \)
2. \( N_d, \text{child nodes} = \text{null}; \)
3. Next-Cad-set = \text{null};
4. For each \( i \in \text{Cad-set} \) do
5. \( X = N_d, \text{itemset}; \)
6. \( Y = \{i\} \cup (X - X \{i\}); \)
7. \( P = \{i\} \cup X; \)
8. \( P, \text{Node set} = X, \text{Node set} \cap Y, \text{Node set}; \)
9. If \( P, \text{support} = X, \text{support} \) then
10. \( \text{equivalent-items} = \text{equivalent-items} \cup \{i\}; \)
11. Else if \( P, \text{support} \leq |DB| \times \varepsilon, \) then
12. Create node \( N_{di} \);
13. \( N_{di}, \text{label} = i; \)
14. \( N_{di}, \text{itemset} = P; \)
15. \( N_{di}, \text{child nodes} = N_{di}, \text{child nodes} \cup \{N_{di}\}; \)
16. Next-Cad-set = Next-Cad-set \cup \{i\};
17. End if
18. End if
19. End if
20. If \( N_d, \text{equivalent-items} \neq \text{null} \) then
21. \( SS = \text{the set of all subsets of } N_d, \text{equivalent-items}; \)
22. \( P, \text{set} = \{A \mid A = N_d, \text{label} \cup A', A' \in SS\}; \)
23. If \( \text{FIS-parent} = \text{null}, \) then
24. FIT-\( N_d = P, \text{set}; \)
25. Else
26. FIT-\( N_d = \{P' \mid P' = P_1 \cup P_2, (P_1 = \text{null} \land P_1 \in P, \text{set}) \land (P_2 = \text{null} \land P_1 \in \text{FIS-parent}); \}
27. End if
28. \( F = F \cup \text{FIT-}N_d; \)
29. End if
30. If \( N_d, \text{child nodes} = \text{null} \) then
31. For each \( N_{di} \in N_d, \text{child nodes} \) do
32. Constructing-Pattern-Tree \((N_{di}, \{j \mid j \in \text{Next-Cad-set, j > i}\}, \text{FIT-}N_d); \)
33. End for

IV. CONCLUSION

The proposed approach in this paper explains how to mine the infrequent items from the data base. Here the mining process represents only IWI mining. In order to mine MIWI we won’t travel to path that contains the item which support is greater than the threshold. This is the thing that is considered in MIWI. But here we specify only IWI mining. But same steps need to be followed except the final mining process.

V. REFERENCES

[1] Luca Cagliero and Paolo Garza,” Infrequent Weighted Itemset Mining Using Frequent Pattern Growth”, IEEE transaction on knowledge and data engineering, VOL. 26, NO. 4, April 2014, 9
G. Amlu pursuing bachelor’s degree in Information Technology at Anand Institute of Higher Technology, Chennai. She is interested in Data Mining domain and she would like to continue her research on data mining.

S. Chandralekha pursuing bachelor’s degree in Information Technology at Anand Institute of Higher Technology, Chennai. She is interested to work with data mining projects and undertaking project on data mining.

R. Praveen Kumar working as an Assistant Professor at Anand Institute of Higher Technology, Chennai. He is interested on wireless network domain. He has completed his P.G in Government College of Technology, Coimbatore.