Date :

Practical No. 8

Aim: To implement Fp Growth algorithm.

Theory:

Frequent pattern Growth Algorithm:

This algo. is an improvement to the Apriori method. A Fp is generated without the need for candidate generation. FP growth algo. represents the db in the form of tree called a Fp tree or Fp tree. This tree structure will maintain the association betn the itemsets. The db is fragmented using one frequent item. This fragmented part is called 'pattern fragment'. The itemset of these fragmented pattern are analyzed. Thus with this method, the search for frequent item sets is reduced comparatively.

FP tree:

FP tree is a tree like structure that is made with the initial itemset of the database. The purpose of FP tree is to mine the most frequent pattern. Each node of Fp tree represent an item of the interest itemset. The root node represents null while the lower nodes represent the itemsets. The association of the nodes with the lower nodes, that is the itemsets with the other itemsets are maintained while forming the tree.

Frequent pattern Algorithm steps:

1. The first step is to scan the db to find the occurences of the itemset in the database. This step is the same as the first step of Apriori. The count of 1-itemsets in the db is called support count or frequency of 1-itemset.
2. The second step is to construct the FP tree. For this, create the root of the tree. The root is represented by Null.

Date :

(3) The next step is to scan the database again and examine the transaction. Examine the first transaction and find out the itemset in it. The itemset with the max count is taken at top, the next itemset with lower count and so on. It means that the branch of the tree is constructed with transaction item sets in descending order of count.

(4) The next transaction in the database is examined. The item sets are ordered in descending order of count. If any itemset of this transaction is already present in another branch, then this transaction branch would share a common prefix to the root. This means that the common itemset is linked to the new node of another itemset in this transaction.

(5) Also, the count of the itemset is incremented as it occurs in the transaction. Both the common node and new node count is increased by 1 as they are created and linked acc to transaction.

(6) The next step is to mine the created FP tree. The lowest node is examined first along with the links to the lowest node. It represents the frequency pattern length 1. From this, traverse path in the FP tree. This path and paths are called a conditional pattern base.
Conditional pattern base is a sub-database consisting of prefix paths in the FP tree occuring with the lowest node

(7) construct a conditional FP tree which is formed by count of itemsets in the path.

(8) FP are generated from conditional FP tree.

Date :

Example :

Support threshold = 50%. Confidence = 60%.

Transaction	List of items
T1	I1, I2, I3
T2	I2, I3, I4
T3	I4, I5
T4	I1, I2, I4
T5	I1, I2, I3, I5
T6	I1, I2, I3, I4

Solution:

Support threshold = 50% $\Rightarrow$ 0.5 * 6 = 3

$\therefore$ min_sup = 3

1. Count each item :

item	count
I1	4
I2	5
I3	4
I4	4
I5	2

6

2. Sort the itemset in descending order :

item	count
I1	5
I2	4
I3	4
I4	4
I5	2

3. Build FP tree :

(1) Considering the root node null.

(2) The first scan of transaction T1 : I1, I2, I3 contain three Items

Date :

{I1:1}, {I2:1}, {I3:1} where I2 is linked as a child to root, I1 is linked to I2 and I3 is linked to I1.

(3) T2: I2, I3, I4 contain I2, I3 and I4 where I2 is linked to root, I3 is linked to I2 and I4 is linked to I3. But this branch would share 2 node I2 node as common as it is already used in T1.

(4) Increment the count of I2 by 1 and I3 is linked as a child to I2, I4 is linked as a child to I3. The count is {I2:2}, {I3:1}, {I4:1}

(5) T3: I4, I5. Similarly, a new branch with I5 is linked to I4 as a child is created.

(6) T4: I1, I2, I4. The sequence will be I2, I1, and I4. I2 is already linked to root node, hence it will be incremented by 1.

(7) T5: I1, I2, I3, I5. The sequence will be I2, I1, I3 & I5
Thus, {I2:4}, {I1:3}, {I3:2}, {I5:1}.

(8) T6: I1, I2, I3, I4. The sequence will be I2, I1, I3, I4.
Thus, {I2:5}, {I1:4}, {I3:3}, {I4:1}

4. Mining of Fp-tree is summarized below:

(1) The lowest node item I5 is not considered as it does not have a min support count, hence it is deleted.

(2) The next lower node is I4. I4 occurs in 2 branches {I2, I1, I3:I4:1}, {I2, I3, I4:1}. Therefore considering I4 as suffix the prefix paths will be {I2, I1, I3:1}, {I2, I3:1}. This forms the conditional pattern base.

(3) The conditional pattern base is considered a transaction db, an Fb tree is constructed. This will contain {I2:2, I3:2} I1 is not considered as it does not meet the main support count.

(4) This path will generate all combinations of Fp:
{I2, I4:2}, {I3, I4:2}, {I2, I3, I4:2}

(5) For I3, the prefix path would be {I2:I1:3}, {I2:1} this will generate a 2 node FP-tree {I2:4, I1:3} and FP generated {I2, I3:4}, {I1:I3:3}, {I2, I1, I3:3}.

Page No.

Date :

⑥ For I1, the prefix path would be : {I2:4} this will generate a single node FP-tree : {I2:4} and FP are generate {I2, I1:4}

Item	Conditional pattern Base	Conditional FP-tree	Frequent pattern generated
I4	{I2, I1, I3:1} {I2, I3:1}	{I2:2, I3:2}	{I2, I4:2}, {I3, I4:2}, {I2, I3, I4:2}
I3	{I2, I1:3}, {I2:1}	{I2:4, I1:3}	{I2, I3:4}, {I1, I3:3} {I2, I1, I3:3}
I1	{I2:4}	{I2:4}	{I2, I1:4}

Advantages:

① This algo needs to scan the db only twice when compared to Apriori which scans the transaction for each iteration

② The pairing of items is not done in this algo, and this makes it faster.

③ The db is stored in a compact version in memory.

④ It is efficient and scalable for mining both long and short frequent patterns.

Disadvantages:

① Fp trees is more cumbersome and difficult to build than Apriori.

② It may be expensive.

③ When the db is large, the algo. may not fit in the shared memory.

Date :

Viva Voce :→

① Define Fp-Growth algorithm ?

→ This algo. is an improvement to the Apriori method. A FP is generated without the need for candidate generation. Fp growth algo. represents the db in the form of tree called a Fp tree.

② How to construct Fp tree ?

→ To put it simply, an FP tree is a compressed representation of the input data. It is constructed by reading the dataset one transaction at one time and mapping each transaction onto a path in the FP-tree structure. As different transactions can have the same items, their paths may overlap.

③ Define frequent pattern ?

→ Fp mining (also Association Rule mining) is an analytical process that finds frequent patterns, association or causal structures from data set found in various kinds of databases such as Relational db, transactional db and other database repositories.