International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org
Volume 3, Issue 4, April 2014 ISSN 2319 - 4847

DFP-Growth: An Efficient Algorithm for Frequent Patterns in Dynamic Data Mining

SUPRIYA SINGH

United College Of Engineering& Research
Gautam Buddha Technical University, Lucknow, INDIA

ABSTRACT

Mining frequent patterns in a large database is still an important and relevant topic in data mining. Nowadays, FP-Growth is one of the famous and benchmarked algorithms to mine the frequent patterns from FP-Tree data structure. However, the major drawback in FP-Growth is, the FP-Tree must be rebuilt all over again once the original database is changed. Therefore, in this paper we introduce an efficient algorithm called Dynamic Frequent Pattern Growth (DFP-Growth) for dynamic data mining.

Keywords: Data Mining, Dynamic Approach, Knowledge Discovery, Association Mining, Frequent Itemsets.

1. INTRODUCTION

Dynamic data mining is a procedure that is activated at certain intervals during the optimization process in order to make use of information obtained during that process, with the goal of speeding the search for optimal solutions. The data is not modified or updated, which is in contrast to the dynamic case, in which we may update the data by incorporating new features or factors as a result of information gained during the optimization process. To accommodate data that changes over time, successive snapshots or samples are taken using updated forecasts or other information, and the analysis proceeds in the form of a moving data window design. Dynamic Data Mining (DDM) combines modern data mining techniques with modern time series analysis techniques. Standard time series analysis deals with the type of temporal sequences of data points and forecasting of future data points that can be used for forecasting data. But it is usually not able to take into account or handle in the best way large amounts of data and large numbers of input variables. Standard data mining, on the other hand, incorporates many methods to handle large numbers of input variables, but it is typically not suitable for time series data. The Dynamic Data Mining technology developed by Adaption combines the best of both worlds: it uses state of the art nonlinear time series analysis and prediction techniques with state of the art data mining techniques for handling and analyzing large amounts of input data [11].

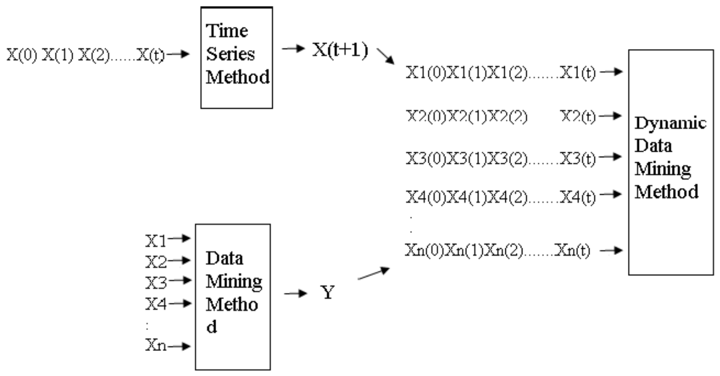

X(0) X(1) X(2)......X(t)
Time Series Method
X(t+1)
X1(0)X1(1)X1(2).......X1(t)
X2(0)X2(1)X2(2) X2(t)
X3(0)X3(1)X3(2).......X3(t)
X4(0)X4(1)X4(2).......X4(t)
Xn(0)Xn(1)Xn(2).......Xn(t)
Dynamic Data Mining Method
X1
X2
X3
X4
Xn
Data Mining Metho d
Y

Figure 1. Dynamic Data Mining Procedure

DDM technology leads to high forecasting accuracy, as shown in multiple business cases. Additionally, an important benefit of Dynamic Data Mining technology is provided by its analysis capabilities. These consist of methods to analyze the patterns in the data and the strengths and weaknesses of the current forecasts. They allow the user to "look inside the black box" to learn more about the data and the forecasting difficulties which a customer faces. It is important to note that DDM does not consist of one single algorithm or one single step of data processing rather, it consists of several components, each of which is important in obtaining good prediction results, and it is the combination of multiple processing components that gives DDM its power.

2. Dynamic Association Rule Mining Approach

For $\sum_{i}^{n} F_{i=1} * \delta_i < 0$, the two options described above could be combined into a single decision rule that says discard S if

$$\sum_{i=k}^{n} F_i * (\text{MINSUP} + \delta_i) \Big/ \sum_{i=k}^{n} F_i \geq \text{MINSUP}/\alpha$$

where $1 \leq \alpha < \infty$, and $k \geq 1$.

$\alpha = 1$ Discard S from the set of a large itemsets (it becomes a small itemset with no history record)
$\alpha \rightarrow \infty$ Keep it for future calculations (it becomes a small itemset with a history record)

The value of α determines how much history information would be carried. This history information along with the calculated values of locality can be used to

1. Determine the significance or the importance of the generated emerged-large itemsets.
2. Determine the significance or the importance of the generated declined-large itemsets.
3. Generate large itemsets with less SUPPORT values without having to rerun the mining procedure again.

The choice of which value of α to choose is the essence of our approach. If the value of α is chosen to be near the value of 1, we will have less declined-large itemsets and more emerged-large itemsets, and those emerged-large itemsets are more to be occurred near the latest interval episodes. For those cases where the value of α is chosen to be far from the value of 1, we will have more declined-large itemsets and less emerged-large itemsets, and those emerged-large itemsets are more to be large itemsets in the Apriori-like approach.
In this section, we introduce the notions of declined-large itemset, emerged-large itemset, and locality.
Let S be a large itemset a emerged-large itemset in a transaction subset T_l, $l \geq 1$. S is called a declined-large itemset in transaction subset T_n , $n > l$, if

$$\text{MINSUP} > \sum_{i=1}^{m} F_i^{*} (\text{MINSUP} + \delta_j) \Big/ \sum_{i=k}^{m} F_j \geq \text{MINSUP}/\alpha$$

for all $l < m \leq n$, where $1 \leq k \leq m$, and $1 \leq \alpha < \infty$,

S is called a emerged-large itemset in transaction subset T_n , $n > 1$, if S was a small itemset in transaction subset T_{n-1} and , or S was a declined-large itemset in transaction subset T_{n-1}, $n > 1$, and $\sum_{i=k}^{n} F_i * \delta_i \geq 0$, $k \geq 1$.
For an itemset S and a transaction subset T_n , locality(S) is defined as the ratio of the total size of those transaction subsets where S is either a large itemset or a emerged-large itemset to the total size of transaction subsets T_i, $1 \leq i \leq n$.

$$\sum_{\substack{\forall i \text{ s.t.S is a large or a} \\ \text{emerged -large itemset}}} F_i \Big/ \sum_{i=1}^{n} F_i$$

Clearly, the locality(S) =1 for all large itemsets S. The dynamic data mining approach generates three sets of itemsets,
1. large itemsets, that satisfy the rule $\sum_{i=1}^{n} F_i * \delta_i \geq 0$, where n is the number of intervals carried out by the dynamic data mining approach
2. declined-large itemsets, that were large at previous intervals and still maintaining the rule MINSUP > $\sum_{i=1}^{m} F_i^{*} (\text{MINSUP} + \delta_j)/\sum_{i=k}^{m} F_j \geq \text{MINSUP}/\alpha$, for some value α.
3. emerged-large itemsets, that were
a) either small itemsets and at a transaction subset Tk they satisfied the rule$F_i * \delta_i \geq 0$, and still satisfy the rule $\sum_{i=1}^{n} F_i * \delta_i \geq 0$,
b) Or they were declined-large itemsets, and at a transaction subset Tm they satisfied the rule $\sum_{i=1}^{n} F_i * \delta_i \geq 0$., and still satisfy the rule $\sum_{i=1}^{n} F_i * \delta_i \geq 0$.

3.Existing Algorithm

We propose an approach that dynamically updates knowledge obtained from the previous data mining process. Transactions over a long duration are divided into a set of consecutive episodes.

3.1 FP-Growth Algorithm

FP-growth (frequent pattern growth) uses an extended prefix-tree (FP-tree) structure to store the database in a compressed form. FP-growth adopts a divide-and-conquer approach to decompose both the mining tasks and the databases. It uses a pattern fragment growth method to avoid the costly process of candidate generation and testing used by Apriori. Compress a large database into a compact, Frequent-Pattern tree (FP-tree) structure highly condensed, but complete for frequent pattern mining, avoid costly database scans, Develop an efficient, FP-tree-based frequent pattern mining method, A divide-and-conquer methodology: decompose mining tasks into smaller ones, Avoid candidate generation: sub-database test only [34].

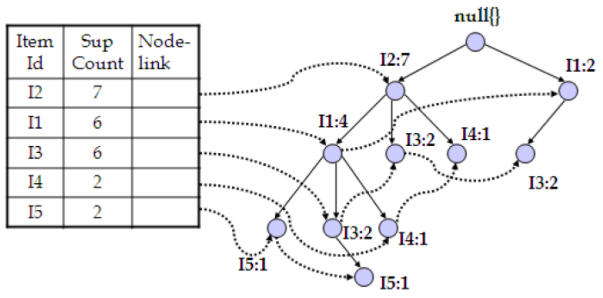
a. Consider the same previous example of a database, D, consisting of 9 transactions.
b. Suppose min. support count required is 2 (i.e. min_sup = 2/9 = 22 %)
c. The first scan of database is same as Apriori, which derives the set of 1-itemsets & their support counts.
d. The set of frequent items is sorted in the order of descending support count.

The resulting set is denoted as L = {I2:7, I1:6, I3:6, I4:2, I5:2}.

3.1.1 FP-Growth Method: Construction of FP-Tree

a. First, create the root of the tree, labeled with "null".
b. Scan the database D a second time. (First time we scanned it to create 1-itemset and then L).
c. The items in each transaction are processed in L order (i.e. sorted order).
d. A branch is created for each transaction with items having their support count separated by colon.
e. Whenever the same node is encountered in another transaction, we just increment the support count of the common node or Prefix.
f. To facilitate tree traversal, an item header table is built so that each item points to its occurrences in the tree via a chain of node-links.
g. Now, The problem of mining frequent patterns in database is transformed to that of mining the FP-Tree [33].

FP-Growth Method: Construction of FP-Tree



Item Id	Sup Count	Node-link
I2	7	
I1	6	
I3	6	
I4	2	
I5	2	

Figure 2 An FP-Tree that registers compressed, frequent pattern information

Mining the FP-Tree by Creating Conditional (sub) pattern bases

Steps:
1. Start from each frequent length-1 pattern (as an initial suffix pattern).
2. Construct its conditional pattern base which consists of the set of prefix paths in the FP-Tree co-occurring with suffix pattern.
3. Then, Construct its conditional FP-Tree & perform mining on such a tree.
4. The pattern growth is achieved by concatenation of the suffix pattern with the frequent patterns generated from a conditional FP-Tree.
5. The union of all frequent patterns (generated by step 4) gives the required frequent itemset.

Table 1: Mining the FP-Tree by creating conditional (sub) pattern bases

Item	Conditional pattern base	Conditional FP-Tree	Frequent pattern generated
I5	{(I2 I1: 1),(I2 I1 I3: 1)}	<I2:2 , I1:2>	I2 I5:2, I1 I5:2, I2 I1 I5: 2
I4	{(I2 I1: 1),(I2: 1)}	<I2: 2>	I2 I4: 2
I3	{(I2 I1: 1),(I2: 2), (I1: 2)}	<I2: 4, I1: 2>,<I1:2>	I2 I3:4, I1, I3: 2 , I2 I1 I3: 2
I2	{(I2: 4)}	<I2: 4>	I2 I1: 4

Now, following the above mentioned steps:
1. Let's start from I5. The I5 is involved in 2 branches namely {I2 I1 I5: 1} and {I2 I1 I3 I5: 1}.
2. Therefore considering I5 as suffix, its 2 corresponding prefix paths would be {I2 I1: 1} and {I2 I1 I3: 1}, which forms its conditional pattern base.
3. Out of these, Only I1 & I2 is selected in the conditional FP-Tree because I3 is not satisfying the minimum support count.

For I1 , support count in conditional pattern base = 1 + 1 = 2
For I2 , support count in conditional pattern base = 1 + 1 = 2
For I3, support count in conditional pattern base = 1
Thus support count for I3 is less than required min_sup which is 2 here.
4. Now, we have conditional FP-Tree with us.
5. All frequent patterns corresponding to suffix I5 are generated by considering all possible combinations of I5 and conditional FP-Tree.
6. The same procedure is applied to suffixes I4, I3 and I1.
7. Note: I2 is not taken into consideration for suffix because it doesn't have any prefix at all.

4. Proposed Algorithm

4.1 Dynamic Fp-Growth Algorithm

$f_1 T_n$ is the set of large and emerged large item set
$f_{1*} T_n$ is the set of decline large item set
Begin
$\Delta = \Delta + F_i$, Where Δ is the count of transaction, F_i the large- emerged large item set
$f_1(T_n) = \sum_{i=1}^{n} F_i * \delta_i \geq 0$, Where n is the number of intervals

$$f_1{}^*(T_n) = \text{MINSUP} > \sum_{i=}^{m} F_i{}^* (\text{MINSUP} + \delta_j) \Big/ \sum_{i=k}^{m} F_j \geq \text{MINSUP}/\alpha$$

Begin

Apply Fp-growth of every Transaction set

$$f_k(T_n) = \{(x, cl_x), x \epsilon c, cl_x = F_n \mid \sum_{i=1}^{n} F_i * \delta_i \geq 0\}$$

$$f_k{}^*(T_n) = \{(x, cl_x) \mid \text{MINSUP} > \sum_{j=1}^{m} F_j{}^* (\text{MINSUP} + \delta_i) \Big/ \sum_{j=k}^{m} F_j \geq \text{MINSUP}/\alpha\}$$

return $f_k T_n$ and $f_{k*}(T_n)$ and
end

FP-tree construction Algorithm

```
Create a root node of FP-Tree label it as null
do for every transaction t
if t is not empty
insert(t,root)
link the new nodes to other nodes with similar labels links originating from
header list.
end do
return FP-Tree
insert(t,any_node)
while t is not empty
any_node has a child node with label head_t
increment the link count between any_node and head_t by1
crate a new node of any_node with label head_t with link count 1
insert (body_t,head_t)
do
```

Steps of Dynamic FP-Growth Algorithm

Step 1: $f_1 T_n$ is the large and emerged item set
Step 2: $f_{1*} T_n$ is the declined item set which is less than minimum support
Step3: To accumulating
$\Delta = \Delta + F_i$

Beg in i=0 to i=n

Step 4: Find out minimum support
Step 5: Find out the count value of large item set
Step 6: Than calculate the support value
Step7: Using this support value we calculate emerged item set, large item set, and declined item set
Step 8: do until MINSUP>σ*MINSUP/DEL
Step 9: Now apply FP-growth algorithm

Step 10: For all transaction t belongs to Tn
Step 11: For all transaction c belongs to Cn
Step12: Then find out all frequent items set from large item set.

5. CONCLUSION AND FUTURE WORK

In this paper, we have introduced a Dynamic Association Rule Mining approach. We also proposed dynamic FP-growth algorithm. The proposed approach performs periodically the data mining process on data updates during a current episode and uses that knowledge captured in the previous episode to produce data mining rules. In our approach, we dynamically update knowledge obtained from the previous data mining process. Transactions domain is treated as a set of consecutive episodes. In our approach, information gained during a current episode depends on the current set of transactions and that discovered information during the previous episode.
As a future work, the Dynamic Association Rule Mining approach will be tested with different datasets that cover a large spectrum of different data mining applications, such as, web site access analysis for improvements in e-commerce advertising, fraud detection, screening and investigation, retail site or product analysis, and customer segmentation.

REFERENCES

[1] Ao, F., Yan, Y., Huang, J., Huang, K. “Mining maximal frequent item sets in data streams based on FP trees” Springer Verlag Berlin Heidelberg pp. 479-489, (2007).
[2] C., Han, J., Pei, J., Yan, X., Yu, P. ”Mining frequent patterns in data stream at multiple time granularities” In Next Generation Data Mining pp. 105-124, (2003).
[3] Clifton, Christopher "Encyclopædia Britannica: Definition of Data Mining" Retrieved (2010).
[4] Ben-David, S., Gehrke, J., Kifer, D. “Detecting change in data streams” Paper presented at the 30th VLDB Conference, Toronto, Canada (2004).
[5] http://www.adapticon.com/index.php?page=dynamic_data_mining.
[6] Joseph Kielman “The real-time nature and value of homeland security information” CIKM '06 Proceedings of the 15th ACM international conference on Information and knowledge management, (2006).
[7] Han Jiawei, Pei, Jian, Yin, Yiwen and Mao, Runying “Mining frequent patterns without candidate generation” Data Mining and Knowledge Discovery 8:53-87, (2004).