Prediction of Contextual Sequential Pattern Mining with Progressive Database

Uma N Dulhare#1, Kavitha Pachika#2, Prof. P. Premchand#3 and Prof. K. Sandyarani#4

#1, #3Department of CSE, University College of Engineering Osmania University, Hyderabad, A.P., India
#2, #4Department of CSE, S.P.M.V.V University, Tirupathi, A.P., India
E-mail: 1umarani_puligila@yahoo.co.in 3drpremchand_p@yahoo.com
4kavitha_pachika@gmail.com

Abstract
Sequential pattern mining is an important data-mining method for determining time-related behaviour in sequence databases. The information obtained from sequential pattern mining can be used in marketing, medical records, sales analysis, and so on. When sequential patterns are generated, the newly arriving patterns may not be identified as frequent sequential patterns due to the existence of old data and sequences. Users are normally more interested in the recent data than the old ones. To capture the dynamic nature of data, addition and deletion, Haung[7] proposed a progressive algorithm Pisa, which stands for Progressive mining of Sequential pAtterns, to progressively discover sequential patterns in defined time period of interest (POI). The POI is a sliding window continuously advancing as the time goes by. Pisa utilizes a progressive sequential tree to efficiently maintain the latest data sequences, discover the complete set of up-to-date sequential patterns, and delete obsolete data and patterns accordingly. An extension of this approach we proposed in this paper, users can select the frequently repeated patterns by allowing the Dynamic Period of Interest (DPOI). Here main focus is on sliding window that changes dynamically, so that the pattern can be extract according to situation.

Keywords: Sequential pattern mining, progressive database, dynamic period of interest

Introduction
The functionalities of data mining techniques include association rules mining, classification, clustering, mining time series, and sequential pattern mining, to name a few[2],[3],[4].Sequential pattern mining was first addressed in [1] as the problem: “Given a sequence database, where each sequence consists of a list of ordered item sets containing a set of different items, and a user defined minimum support threshold, sequential pattern mining is to find all subsequences whose occurrence frequencies are no less than the threshold from the set of sequences.”

The sequential pattern mining with a static database finds the sequential patterns in the database in which data do not change over time [5]. On the other hand, the sequential pattern mining with an incremental database corresponds to the mining process where there are new data arriving as time goes by (i.e., the sequences database is incremental) [6]. As for the sequential pattern mining with a progressive database, new data are added into the database and obsolete data are removed simultaneously. Therefore, one can find the most up-to-date sequential patterns without being influenced by obsolete data.

The existing algorithms cannot cope with sequential pattern mining with a progressive database efficiently. To remedy this problem, Haung [7] proposed an efficient algorithm Pisa, which stands for Progressive mining of Sequential patterns, corresponding to the mining in a progressive database.

Problem Description
POI is a sliding window, whose length is a user specified time interval, continuously advancing as the time goes by. The sequences having elements whose timestamps fall into this period, POI, contribute to the |Db| for current sequential patterns. On the other hand, the sequences having only elements with timestamps older than POI should be pruned away from the sequence database immediately and will not contribute to the |Db| thereafter.

PS-tree represents elements in the sequence, based on the sequence IDs and timestamps recorded in the nodes and the newly arriving data of the progressive database at each timestamp. PS-tree not only stores the elements and timestamps of sequences in each POI but also efficiently accumulates the occurrence frequency of every candidate sequential pattern at the same time.

Fig1 sample database
Consider the progressive database in Fig. 1, for example. S01; S02; . . . ; Sn represent different sequence IDs. A, B, C, and D are different items in the database and t1; t2; . . . ; tk represent timestamps. As the time advances, there will be more elements arriving into the progressive database. Every sequence contains a series of elements appearing at different timestamps. Each element consists of a single or multiple items.

For instance, sequence S01 has element A at timestamp t1, element B at timestamp t2, element C at timestamp t4, and element . At the bottom of Fig. 1, Db represents a subset of the database containing the elements from timestamp p to timestamp q. Let the minimum support threshold, min sup, be 0.5 and the POI be five timestamps in this example. There are five sequences having elements in this period. Therefore, the minimum frequency for a frequent sequential pattern is $|Db| \times \text{min sup} = 5 \times 0.5 = 2.5$.

We can find a frequent sequential pattern AB, whose occurrence frequency is 3 (in S01, S02, and S03) in the first POI. However, after this POI, AB is no longer a frequent sequential pattern in any POI of five timestamps. PS-tree not only contains the information of all sequences in a progressive database but also helps Pisa to generate frequent sequential patterns in each POI. The nodes in PS-tree can be divided into two different types. They are root node and common nodes. They are root node and common nodes. Root node is the root of PS-tree containing nothing but a list of common nodes as its children. Each common node stores two information, say node label and a sequence list. The label is the same as the element in a sequence. The sequence list stores a list of sequence IDs to represent the sequences containing this element. Each sequence ID in the sequence list is marked by a corresponding timestamp.

If we require frequent patterns in between timestamp 2 and 4 then apply algorithm Dpisa as Fig 3.

Let the minimum support threshold, min_sup be 0.5 and the in this example Dynamic POI is given by startTime 2 and endTime 4. There are three sequences having elements in this period. Therefore, the minimum frequency for a frequent sequential pattern is $|Db| \times \text{min sup} = 3 \times 0.5 = 1.5$. We can find frequent sequential pattern within that period as BC(2) pattern as Fig 4.

Algorithm Dpisa (support, POI)
Var PS ;//PS-tree
Var currentTime, startTime, endTime

Var eleset //used to store elements ele
While( there is new transaction)
  eleset=read all elements at currentTime;
  While( endTime<=startTime)
    if (sequences are given) then
      currentTime=startTime;
      Traverse(currentTime,PS)
      currentTime++;
    end.

Conclusions
Our proposed work mine frequent pattern in progressive sequential databases seasonally in market basket analysis. Every time sales analysis is not uniform so user can extract the pattern s on his own interest. The major constraint is it consider only recent items. To achieve this we modified progressive sequential algorithm by using startTime and endTime as dynamic period of interest our goal is according to requirement user can extract patterns dynamically.

References
[4] J. Han and M. Kamber, Data Mining: Concepts and
Prediction of Contextual Sequential Pattern Mining with Progressive Database

