Q6

Define sequence pattern mining and understand the working of

i) GSP

ii) Prefix-Span algorithms.

Give a pseudocode and illustrate the same over a sample dataset of your choice.

Sequence Pattern Mining

Introduction:

Sequential Pattern Mining is the mining of frequently occurring ordered events or subsequences as pattern in sequence database.

A Sequence Database stores a number of records, where all records are sequences of ordered events, with or without concrete notions of time

Sequential Patterns are used for targeted marketing and customer retention

1. GSP

Introduction:

The Generalized Sequence Pattern algorithm was created from a simpler algorithm for mining sequences, but it has some extra bells and whistles added so it can be more flexible for different situations.

Pseudo Code:

F1 = the set of frequent 1-sequence

k=2,

do while Fk-1 != Null;

Generate candidate sets Ck (set of candidate k-sequences);

For all input sequences s in the database D

do

Increment count of all a in Ck if s supports a

End do

Fk = {a ∈ Ck such that its frequency exceeds the threshold}

k = k+1;

End do

Result = Set of all frequent sequences is the union of all Fk's

Trace:

Input Data

|  |  |  |
| --- | --- | --- |
| Transaction ID | Customer ID | Items |
| 1 | 1 | A |
| 2 | 1 | B |
| 3 | 1 | A |
| 4 | 2 | B |
| 5 | 2 | A |
| 6 | 2 | B |

First we prune Items with less support than threshold

Let threshold = 1

A – 3 > 1

B – 3 > 1

Now, get the sequences for each customer

|  |  |
| --- | --- |
| Customer ID | Sequence |
| 1 | ABA |
| 2 | BAB |

L1 = {A, B}

We generate C2,

C2 = {AA, BB, AB, BA}

Supports are,

AA – 1 – (A)B(A) in CID 1

BB – 1 – (B)A(B) in CID 2

AB – 2 – (AB)A and B(AB) in CID 1 and 2

BA – 2 – A(BA) and (BA)B in CID 1 and 2

All > 1 (Threshold)

L2 = {AA, AB, BA, BB}

We generate C3,

C3 = {AAA, AAB, ABA, ABB, BAA, BAB, BBA, BBB}

Supports are,

AAA – 0

AAB – 0

ABA – 1 >= 1

ABB – 0

BAA – 0

BAB – 1 >= 1

BBA – 0

BBB – 0

So, L3 = {ABA, BAB}

So, frequent sequences are {A, B, AA, AB, BA, BB, ABA, BAB}

1. Prefix Span Algorithm

Introduction:

A pattern-growth method based on projection is used in Prefix Span algorithm for mining sequential patterns. The basic idea behind this method is, rather than projecting sequence databases by evaluating the frequent occurrences of sub-sequences, the projection is made on frequent prefix. This helps to reduce the processing time which ultimately increases the algorithm efficiency.

Pseudo Code:

**Input:** A sequence database S, and the minimum support threshold min\_sup

**Output:** The complete set of sequential patterns

**Parameters:**

1. α: sequential pattern,
2. l: the length of α;
3. S|α: the α-projected database, if α ≠<>; otherwise; the sequence database S

Algorithm

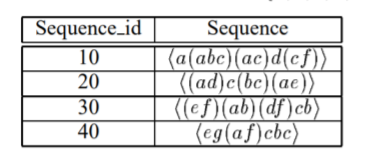
PrefixSpan(α, l, S|α)

1. Scan S|α once, find the set of frequent items b such that:
   1. b can be assembled to the last element of α to form a sequential pattern; or
   2. can be appended to α to form a sequential pattern.
2. For each frequent item b, append it to α to form a sequential pattern α’, and output α’;
3. For each α’, construct α’-projected database S|α’, and call PrefixSpan(α’, l+1, S|α’).

Trace:

Let minsup be 2,

Dataset,



Find length -1 sequential patterns,

Scan the database once to find all frequent items in sequences. Each of these frequent items is a length-1 sequential pattern. They are:

< a >: 4, < b >: 4, < c >: 4, < d >: 3, < e >: 3, < f >: 3

where < prefix >: count.

Divide search space

The complete set of sequential patterns can be partitioned into the following six subsets according to the six prefixes: (1) the ones having prefix < a > ; ... ; and (6) the ones having prefix < f >.

Find subsets of sequential patterns

The subsets of sequential patterns can be mined by constructing corresponding projected databases and mine each recursively. The projected databases as well as sequential patterns found in them are listed in Table 2, while the mining process is explained as follows, First, let us find sequential patterns having prefix < a >. Only the sequences containing < a > should be collected.ted. Moreover, in a sequence containing < a >, only the subsequence prefixed with the first occurrence of < a >, should be considered. For example, in sequence < (ef)(ab)(df)cb > only the subsequence < ( b)(df)cb > should be considered for mining sequential patterns having prefix < a >. Notice that (b) means that the last element in the prefix, which is a, together with b, form one element. As another example, only the subsequence < (abc)(ac)d(cf) > of sequence < a(abc)(ac)d(cf) > is considered. For our example,

Sequences in S containing < a > are projected wrt < a > to form the < a > - projected database, which consists of four postfix sequences : < (abc)(ac)d(cf) >, < ( d)c(bc)(ae) >, < ( b)(df)cb > and < ( f)cbc >. By scanning < a >-projected database once, all the lenght-2 sequential patterns having prefix < a > can be found.

They are: < aa >: 2, < ab >: 4, < (ab) >: 2, < ac >: 4, < ad >: 2, < af >: 2.

Recursively, all sequential having patterns prefix < a > can be partitioned into 6 subsets: (1) those having prefix < aa >, (2) those having prefix < ab >, . . ., and finally, (6) those having prefix < af >. These subsets can be mined by constructing respective projected databases and mining each recursively.

The < aa > -projected database consists of only one non-empty (postfix) subsequences having prefix < aa > : < ( bc)(ac)d(cf) >. Since there is no hope to generate any frequent subsequence from a single sequence, the processing of < aa >-projected database terminates.

The < ab > -projected database consists of three postfix sequences: < ( c)(ac)d(cf) >, < ( c)a > and < c >.

Recursively mining < ab > -projected database returns fours sequential patterns: < ( c) >, < ( c)a >, < a > and < c > (i.e. < a(bc) >, < a(bc)a >, < aba > and < abc >).

The < (ab) > projected sequence only consist of two sequence, < ( c)(ac)d(cf) > and < (df)c >, which leads to the finding of the following sequential patterns having prefix < (ab) > : < c >, < d >, < f > and < dc >.

The < ac > −, < ad > − and < af > − projected databases can be constructed and recursively mined similarly. The sequential patterns found are shown in figure.

Similarly, we can find sequential patterns having prefix < b >, < c >, < d >, < e > and < f > by constructing < b, > −, < c > −, < d > −, < e > − and < f > − projected databases and mining them respectively. The projected databases are shown in figure.

Projected Databases and Sequential Patterns,

