Mining Local Periodic Patterns in Temporal Databases

What is local periodic pattern mining?

Local periodic pattern mining aims to discover all interesting patterns in a temporal database that have **periodicity** no greater than the user-specified **maximum periodicity** (**maxPer**) constraint, **time interval of occurence** no greater than user-specified **maximum period of spillovers** (**maxSoPer**) constraint and **minDur** is no less than **minimum duration (minDur)**. The **minDur** controls the minimum duration that a pattern is reocurring.

Research paper: Fournier Viger, Philippe & Yang, Peng & Rage, Uday & Ventura, Sebastian & Luna, José María. (2020). Mining Local Periodic Patterns in a Discrete Sequence. Information Sciences. 544. 10.1016/j.ins.2020.09.044.

What is a temporal database?

A temporal database is a collection of transactions at a particular timestamp, where each transaction contains a timestamp and a set of items.
A hypothetical temporal database containing the items ***a, b, c, d, e, f, and g*** as shown below

TS	Transactions
1	a b c g
2	b c d e
3	a b c d
4	a c d f
5	a b c d g
6	c d e f
7	a b c d
8	a e f
9	a b c d
10	b c d e

Note: Duplicate items must not exist in a transaction.

Acceptable format of temporal databases in PAMI

Each row in a temporal database must contain timestamp and items.

1 a b c g

2 b c d e

3 a b c d

4 a c d f

5 a b c d g

6 c d e f

7 a b c d

8 a e f

9 a b c d

10 b c d e

Understanding the statisctics of database

To understand about the database. The below code will give the detail about the transactional database.

- Total number of transactions (Database size)
- Total number of unique items in database
- Minimum lenth of transaction that existed in database
- Average length of all transactions that exists in database
- Maximum length of transaction that existed in database
- Minimum periodicity exists in database
- Average periodicity exists in database
- Maximum periodicity exists in database
- Standard deviation of transaction length
- Variance in transaction length
- Sparsity of database

The sample code

In [ ]:

```
import PAMI.extras.dbStats.temporalDatabaseStats as stats

obj = stats.temporalDatabaseStats('sampleInputFile.txt', ' ')
obj.run()
obj.printStats()
```

What is the input to local periodic pattern mining algorithms

Algorithms to mine the local periodic patterns requires temporal database, maxPer, maxSoPer and minDur (specified by user).

- Temporal database is accepted following formats:

  > - String : E.g., 'temporalDatabase.txt'
  > - URL : E.g., https://u-aizu.ac.jp/~udayrage/datasets/transactionalDatabases/transactional_T10
  > - DataFrame. Please note that dataframe must contain the header titled 'TS' and 'Transactions'

- maxPer should be mentioned in

  > - **count (beween 0 to length of database)**
  > - [0, 1]

- maxSoPer should be mentioned in

  > - **count (beween 0 to length of database)**
  > - [0, 1]

- minDur should be mentioned in

  > - **count (beween 0 to length of database)**
  > - [0, 1]

- seperator
  default seperator is '\t' (tab space)

How to run the local periodic pattern algorithm in terminal

- Download the PAMI source code from github.
- Unzip the PAMI source code folder and enter into local periodic pattern folder.
- Enter into localPeriodicPattern folder
- Enter into a **basic** folder of your choice and execute the following command on terminal.

syntax: python3 algorithmName.py `<path to the input file>` `<path to the output file>` `<maxPer>` `<maxSoPer>` `<minDur>` `<seperator>`

Sample command to execute the LPPGrowth algorithm in localPeriodicPattern/basic folder

python3 `LPPGrowth.py` `inputFile.txt` `outputFile.txt` 3 4 2 `' '`

How to implement the LPPGrowth algorithm by importing PAMI package

Import the PAMI package executing: **pip3 install PAMI**

- Install the PAMI package from the PYPI repository by executing the following command: **pip3 install PAMI**
- Run the below sample code by making necessary changes

In [4]:

```
import PAMI.localPeriodicPattern.basic.LPPGrowth as alg

iFile = 'sampleInputFile.txt'  #specify the input transactional database

maxPer = 3  #specify the maxPer value <br>
maxSoPer = 5  #specify the maxSoPer value <br>
minDur = 5  #specify the minDur value <br>
seperator = ' ' #specify the seperator. Default seperator is tab space. <
oFile = 'localPeriodicPatterns.txt'   #specify the output file name<br>

obj = alg.LPPGrowth(iFile, maxPer, maxSoPer, minDur, seperator) #initiali
obj.startMine()                        #start the mining process <br>
obj.savePatterns(oFile)                #store the patterns in file <br>
df = obj.getPatternsAsDataFrame()      #Get the patterns discovered into a
#obj.printStats()                       #Print the statistics of mining pr
```

The localPeriodicPatterns.txt file contains the following patterns (*format:* pattern:support):!cat localPeriodicPatterns.txt

In [5]:

```
!cat localPeriodicPatterns.txt
```

```
d : {(2, 18)}
('d', 'c') : {(2, 10)}
('d', 'c', 'b') : {(2, 10)}
('d', 'c', 'b', 'a') : {(3, 10)}
('d', 'c', 'a') : {(3, 10)}
('d', 'b') : {(2, 10)}
('d', 'b', 'a') : {(3, 10)}
('d', 'a') : {(3, 10)}
c : {(1, 10)}
('c', 'b') : {(1, 10)}
('c', 'b', 'a') : {(1, 10)}
('c', 'a') : {(1, 10)}
b : {(1, 10)}
('b', 'a') : {(1, 10)}
a : {(1, 9)}
```

The dataframe containing the patterns is shown below:

In [6]:

```
df
```

Out[6]:

	Patterns	PTL
0	d	{(2, 18)}
1	(d, c)	{(2, 10)}
2	(d, c, b)	{(2, 10)}
3	(d, c, b, a)	{(3, 10)}
4	(d, c, a)	{(3, 10)}
5	(d, b)	{(2, 10)}
6	(d, b, a)	{(3, 10)}
7	(d, a)	{(3, 10)}
8	c	{(1, 10)}
9	(c, b)	{(1, 10)}
10	(c, b, a)	{(1, 10)}
11	(c, a)	{(1, 10)}
12	b	{(1, 10)}
13	(b, a)	{(1, 10)}
14	a	{(1, 9)}