Mining Partial Periodic Patterns in Temporal Databases

What is partial periodic pattern mining?

Periodic-Frequent pattern mining aims to discover all interesting patterns in a temporal database that have **periodic support (frequency)** no less than the user-specified **minimum periodic support** (**minPS**) constraint and **inter arrival time** no greater than the specified **maximum inter-arrival time (maxIAT)**. The **periodic support** controls the count of periodic intervals of pattern occuring in the database. The **inter arrival time** controls the maximum interval time that pattern should reappear.

Ressearch paper: R. Uday Kiran, Haichuan Shang, Masashi Toyoda, and Masaru Kitsuregawa. 2017. Discovering Partial Periodic Itemsets in Temporal Databases. In Proceedings of the 29th International Conference on Scientific and Statistical Database Management (SSDBM '17). Association for Computing Machinery, New York, NY, USA, Article 30, 1–6. https://doi.org/10.1145/3085504.3085535

What is the 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 below sample code prints the statistical details of a database.

In [ ]:
```
import PAMI.extras.dbStats.temporalDatabaseStats as stats

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

What are the input parameters?

The input parameters to a partial periodic pattern mining algorithm are:

- **Temporal database**
  Acceptable formats:

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

- **minPS**
  specified in

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

- **maxIAT**
  specified in

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

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

How to store the output of a frequent pattern mining algorithm?

The patterns discovered by a frequent pattern mining algorithm can be saved into a file or a data frame.

How to run the partial periodic pattern algorithm in terminal

- Download the PAMI source code from github.
- Unzip the PAMI source code folder and enter into partial periodic pattern folder.
- Enter into partialPeriodicPattern folder
- You will find different types of folders like **basic, maximal, timeseries**
- Enter into a specific 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>` `<minPS>` `<maxPer>` `<seperator>`

Example: python3 `PPPGrowth.py` `/Users/Donwloads/inputFile.txt` `/Users/Downloads/outputFile.txt` `4` `3` `' '`

How to execute a partial periodic pattern mining algorithm in a Jupyter Notebook?

- 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 [ ]:
```
import PAMI.partialPeriodicPattern.basic.PPPGrowth as alg

iFile = 'sampleInputFile.txt'  #specify the input temporal database <br>
minPS = 5                       #specify the minPS value <br>
maxIAT = 2                      #specify the maxIAT value <br>
seperator = ' '                  #specify the seperator. Default seperato
oFile = 'partialPatterns.txt'   #specify the output file name<br>

obj = alg.PPPGrowth(iFile, minPS, maxIAT, seperator) #initialize the algo
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 pro
```

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

In [3]:
```
!cat partialPatterns.txt
```

```
b :6
b d :5
b d c :5
b c :6
a :6
a c :5
d :7
d c :7
c :8
```

The dataframe containing the patterns is shown below:

```
In [4]: df
```

Out[4]:

	Patterns	periodicSupport
0	b	6
1	b d	5
2	b d c	5
3	b c	6
4	a	6
5	a c	5
6	d	7
7	d c	7
8	c	8