Discovering Maximal Partial Periodic Pattern in Big Data Using Max3PGrowth Algorithm

In this tutorial, we will discuss two approaches to find Maximal Partial Periodic Pattern in big data using Max3PGrowth algorithm.

1. **Basic approach:** Here, we present the steps to discover Maximal Partial Periodic Pattern using a single minimum support value
2. **Advanced approach:** Here, we generalize the basic approach by presenting the steps to discover Maximal Partial Periodic Pattern using multiple minimum support values.

Basic approach: Executing Max3PGrowth on a single dataset at a particular minimum support value

Step 1: Import the Max3PGrowth algorithm

In [1]:

```
from PAMI.partialPeriodicPattern.maximal import Max3PGrowth  as alg
```

Step 2: Specify the following input parameters

In [2]:

```
inputFile = 'temporal_T10I4D100K.csv'
periodCount=5000
periodicSupportCount=100
  #Users can also specify this constraint between 0 to 1.

seperator='¥t'
```

Step 3: Execute the Max3PGrowth algorithm

In [3]:

```
obj = alg.Max3PGrowth(iFile=inputFile,periodicSupport=periodicSupportCount, period=
obj.startMine()             #Start the mining process
```

```
Maximal Partial Periodic Frequent patterns were generated successfully using MAX-3PG
rowth algorithm
```

Step 4: Storing the generated patterns

Step 4.1: Storing the generated patterns in a file

In [4]:

```
obj.savePatterns(outFile='frequentPatternsMinSupCount1000.txt')
```

Step 4.2. Storing the generated patterns in a data frame

In [5]:

```
frequentPatternsDF= obj.getPatternsAsDataFrame()
```

Step 5: Getting the statistics

Step 5.1: Total number of discovered patterns

In [6]:

```
print('Total No of patterns: ' + str(len(frequentPatternsDF)))
```

```
Total No of patterns: 3948
```

Step 5.2: Runtime consumed by the mining algorithm

In [7]:

```
print('Runtime: ' + str(obj.getRuntime()))
```

```
Runtime: 12.587928771972656
```

In [8]:

```
##### Step 5.3: Total Memory consumed by the mining algorithm
```

In [9]:

```
print('Memory (RSS): ' + str(obj.getMemoryRSS()))
print('Memory (USS): ' + str(obj.getMemoryUSS()))
```

```
Memory (RSS): 576737280
Memory (USS): 537845760
```