Discovering Fuzzy Periodic Frequent Pattern in Big Data Using FPFPMiner Algorithm

In this tutorial, we will discuss two approaches to find Fuzzy Periodic Frequent Pattern in big data using FPFPMiner algorithm.

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

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

Step 1: Import the FPFPMiner algorithm

In [1]:
```
from PAMI.fuzzyPeriodicFrequentPattern.basic import FPFPMiner  as alg
```

Step 2: Specify the following input parameters

In [2]:
```
inputFile = 'T10_utility.txt'
periodCount=1000
minimumSupportCount=1000  #Users can also specify this constraint between 0 to 1.

seperator=' '
```

Step 3: Execute the FPFPMiner algorithm

In [3]:
```
obj = alg.FPFPMiner(iFile=inputFile, minSup=minimumSupportCount,period=periodCount,
obj.startMine()            #Start the mining process
```

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: 382
```

Step 5.2: Runtime consumed by the mining algorithm

In [7]:
```
print('Runtime: ' + str(obj.getRuntime()))
```

```
Runtime: 336.29632902145386
```

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): 427794432
Memory (USS): 388661248
```