

Discovering Partial Periodic Frequent Pattern in Big Data Using GPFgrowth Algorithm

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

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

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

Step 1: Import the GPFgrowth algorithm

```
In [1]: from PAMI.partialPeriodicFrequentPattern.basic import GPFgrowth as alg
```

Step 2: Specify the following input parameters

```
In [2]: inputFile = 'temporal_T10I4D100K.csv'
maxPerCount=500
minPRcount=0.5
minimumSupportCount=100 #Users can also specify this constraint between 0 to 1.

seperator='¥t'
```

Step 3: Execute the GPFgrowth algorithm

```
In [3]: obj = alg.GPFgrowth(iFile=inputFile, minSup=minimumSupportCount, maxPer=maxPerCount, r
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: 20688

Step 5.2: Runtime consumed by the mining algorithm

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

Runtime: 54.974111795425415

```
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): 696582144

Memory (USS): 658100224