

Discovering Frequent Patterns in Big Data Using Frequent-Pattern Growth Algorithm

In this tutorial, we will discuss two approaches to find frequent patterns in big data using FP-growth algorithm.

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

Basic approach: Executing FP-growth on a single dataset at a particular minimum support value

Step 1: Import the FP-growth algorithm

```
In [1]: from PAMI.frequentPattern.basic import FPGrowth as alg
```

Step 2: Specify the following input parameters

```
In [2]: inputFile = 'transactional_T10I4D100K.csv'

minimumSupportCount=100 #Users can also specify this constraint between 0 to 1.

seperator='¥t'
```

Step 3: Execute the Apriori algorithm

```
In [3]: obj = alg.FPGrowth(iFile=inputFile, minSup=minimumSupportCount, sep=seperator) #
obj.startMine() #Start the mining process
```

Frequent patterns were generated successfully using frequentPatternGrowth algorithm

Step 4: Storing the generated patterns

Step 4.1: Storing the generated patterns in a file

```
In [4]: obj.savePatterns(outFile='frequentPatternsMinSupCount100.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: 27532

Step 5.2: Runtime consumed by the mining algorithm

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

Runtime: 10.403892993927002

```
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): 517066752

Memory (USS): 478625792