

Discovering Fuzzy Correlated Patterns in Big Data Using FCPGrowth Algorithm

In this tutorial, we will discuss two approaches to find Fuzzy Correlated Patterns in big data using FCPGrowth algorithm.

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

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

Step 1: Import the FCPGrowth algorithm

```
In [1]: from PAMI.fuzzyCorrelatedPattern.basic import FCPGrowth as alg
```

Step 2: Specify the following input parameters

```
In [2]: inputFile = 'T10_utility.txt'

minimumSupportCount=1200 #Users can also specify this constraint between 0 to 1.
ratioExample=0.8
seperator='¥t'
```

Step 3: Execute the FCPGrowth algorithm

```
In [3]: obj = alg.FCPGrowth(iFile=inputFile, minSup=minimumSupportCount, ratio=ratioExample, s
obj.startMine() #Start the mining process
```

Fuzzy Correlated Patterns Successfully generated using FCPGrowth algorithms

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: 318

Step 5.2: Runtime consumed by the mining algorithm

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

Runtime: 260.4294843673706

```
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): 475705344

Memory (USS): 436994048