BMS1

June 16, 2023

[1]:
```
import pandas as pd
import Topk_PPPGrowth as tp
```

[2]:
```
inputFile = 'Temporal_BMS_Webview1.csv'
seperator = '\t'
k = [150, 300, 450, 700, 850]
maxPer = 1000

totalResult = pd.DataFrame(columns=['algorithm', 'minSup', 'maxPer','patterns',␣
 ↪'runtime', 'memory'])
#initialize a data frame to store the results of PFECLAT algorithm
```

[3]:
```
algorithm = 'TOPK-3P'  #specify the algorithm name
for i in k:
    obj1 = tp.Topk_PPPGrowth(inputFile, k=i, periodicity=maxPer, sep=seperator)
    obj1.startMine()
    obj1.save('patterns.txt')
    #store the results in the data frame
    totalResult.loc[totalResult.shape[0]] = [algorithm, i, maxPer, len(obj1.
 ↪getPatterns()), obj1.getRuntime(), obj1.getMemoryRSS()]
```

```
150 150 1000
TopK partial periodic patterns were generated successfully
300 300 1000
TopK partial periodic patterns were generated successfully
450 450 1000
TopK partial periodic patterns were generated successfully
484 700 1000
TopK partial periodic patterns were generated successfully
484 850 1000
TopK partial periodic patterns were generated successfully
```

[4]:
```
print(totalResult)
```

```
  algorithm  minSup  maxPer  patterns   runtime     memory
0   TOPK-3P     150    1000       150  0.303891  143364096
1   TOPK-3P     300    1000       300  0.663416  144072704
2   TOPK-3P     450    1000       450  1.813450  144273408
```

```
3   TOPK-3P     700     1000       700    9.198055  144732160
4   TOPK-3P     850     1000       850   35.177038  144891904
```

[5]:

```python
def getTopPatterns(iFile, k):
    res = {}
    with open(iFile, 'r') as f:
        for line in f:
            line = line.split(':')
            res[line[0]] = line[1]
    res1 = {k:v for k, v in sorted(res.items(), key=lambda x:x[1],␣
 ↪reverse=True)}
    res1 = {k:v for k,v in list(res1.items())[:k]}
    return res1
```

[6]:

```python
import time
import os as _os
import os.path as _ospath
import psutil as _psutil
from PAMI.partialPeriodicPattern.basic import PPPGrowth as pf
startTime = time.time()
for i in [200, 400, 600, 800, 900]:
    obj = pf.PPPGrowth(inputFile, 30, 1000, '\t')
    obj.startMine()
    obj.save("patterns_bms1.txt")
    patterns = getTopPatterns("patterns_t10.txt", i)
    endTime = time.time()
    memoryUSS = float()
    process = _psutil.Process(_os.getpid())
    memoryUSS = process.memory_full_info().uss
    print("Total Number of patterns:", len(patterns))
    print("Total Memory Taken:", memoryUSS)
    print("Total Time Taken:", endTime - startTime)
```

```
Partial Periodic Patterns were generated successfully using 3PGrowth algorithm
Total Number of patterns: 200
Total Memory Taken: 159158272
Total Time Taken: 2.179365396499634
Partial Periodic Patterns were generated successfully using 3PGrowth algorithm
Total Number of patterns: 400
Total Memory Taken: 163737600
Total Time Taken: 4.317350625991821
Partial Periodic Patterns were generated successfully using 3PGrowth algorithm
Total Number of patterns: 600
Total Memory Taken: 164884480
Total Time Taken: 6.534010648727417
Partial Periodic Patterns were generated successfully using 3PGrowth algorithm
Total Number of patterns: 800
Total Memory Taken: 165019648
Total Time Taken: 8.696280241012573
Partial Periodic Patterns were generated successfully using 3PGrowth algorithm
Total Number of patterns: 900
Total Memory Taken: 165019648
Total Time Taken: 10.953281879425049
```

[ ]: