

T10I4D200K

August 11, 2023

1 Step 1: Import the “T10I4D200K” dataset, UPFPGrowth, and UPFPGrowth++ code

```
[1]: import UPFPGrowth as upfp
import UPFPGrowthPlus as upfpp
import pandas as pd
inputFile = 'uncertain_Temporal_T10I4D200K.csv'
result = pd.DataFrame(columns=['algorithm', 'minSup', 'maxPer', 'patterns',
↪ 'runtime', 'memory'])
```

2 Step 2: Finding the periodic-frequent patterns using UPFPGrowth++ by varying minSup and with fixed maxPer

```
[2]: algorithm = 'UPFPGrowth++'
minSup = [400, 500, 600, 700, 800, 900, 1000]
maxPer = 8000
for i in minSup:
    obj = upfpp.UPFPGrowthPlus(iFile=inputFile, minSup=i, maxPer=maxPer,
↪ sep='\t')
    obj.startMine()
    result.loc[result.shape[0]] = [algorithm, i, maxPer, len(obj.
↪ getPatterns()), obj.getRuntime(), obj.getMemoryUSS()]
    obj.printResults()
```

Periodic Frequent patterns were generated successfully using UPFPGrowth++ algorithm

Total number of Uncertain Periodic-Frequent Patterns: 609

Total number of generated false patterns: 202

Total Memory in USS: 1006804992

Total Memory in RSS 1065025536

Total ExecutionTime in ms: 64.34710168838501

Periodic Frequent patterns were generated successfully using UPFPGrowth++ algorithm

Total number of Uncertain Periodic-Frequent Patterns: 532

Total number of generated false patterns: 62

Total Memory in USS: 1006247936

```

Total Memory in RSS 1064468480
Total ExecutionTime in ms: 37.51975178718567
Periodic Frequent patterns were generated successfully using UPFPgrowth++
algorithm
Total number of Uncertain Periodic-Frequent Patterns: 478
Total number of generated false patterns: 23
Total Memory in USS: 991379456
Total Memory in RSS 1050189824
Total ExecutionTime in ms: 30.614928483963013
Periodic Frequent patterns were generated successfully using UPFPgrowth++
algorithm
Total number of Uncertain Periodic-Frequent Patterns: 435
Total number of generated false patterns: 4
Total Memory in USS: 977100800
Total Memory in RSS 1035911168
Total ExecutionTime in ms: 24.866675853729248
Periodic Frequent patterns were generated successfully using UPFPgrowth++
algorithm
Total number of Uncertain Periodic-Frequent Patterns: 398
Total number of generated false patterns: 1
Total Memory in USS: 962265088
Total Memory in RSS 1021075456
Total ExecutionTime in ms: 23.8896701335907
Periodic Frequent patterns were generated successfully using UPFPgrowth++
algorithm
Total number of Uncertain Periodic-Frequent Patterns: 363
Total number of generated false patterns: 1
Total Memory in USS: 946683904
Total Memory in RSS 1005436928
Total ExecutionTime in ms: 22.86089038848877
Periodic Frequent patterns were generated successfully using UPFPgrowth++
algorithm
Total number of Uncertain Periodic-Frequent Patterns: 337
Total number of generated false patterns: 0
Total Memory in USS: 933785600
Total Memory in RSS 992595968
Total ExecutionTime in ms: 21.281518936157227

```

3 Step 3: Print the results

```
[3]: print(result)
```

	algorithm	minSup	maxPer	patterns	runtime	memory
0	UPFPGrowth++	400	8000	609	64.347102	1006804992
1	UPFPGrowth++	500	8000	532	37.519752	1006247936
2	UPFPGrowth++	600	8000	478	30.614928	991379456
3	UPFPGrowth++	700	8000	435	24.866676	977100800

4	UPFPGrowth++	800	8000	398	23.889670	962265088
5	UPFPGrowth++	900	8000	363	22.860890	946683904
6	UPFPGrowth++	1000	8000	337	21.281519	933785600

4 Step 4: Executing UPFPGrowth

```
[4]: algorithm = 'UPFPGrowth'
minSup = [400, 500, 600, 700, 800, 900, 1000]
maxPer = 8000
for i in minSup:
    obj = upfp.UPFPGrowth(iFile=inputFile, minSup=i, maxPer=maxPer, sep='\t')
    obj.startMine()
    result.loc[result.shape[0]] = [algorithm, i, maxPer, len(obj.
    getPatterns()), obj.getRuntime(), obj.getMemoryUSS()]
    obj.printResults()
```

Periodic frequent patterns were generated successfully using UPFP algorithm
Total number of Uncertain Periodic-Frequent Patterns: 610
Total number of False Patterns: 303
Total Memory in USS: 1001152512
Total Memory in RSS 1060397056
Total ExecutionTime in ms: 76.65591835975647
Periodic frequent patterns were generated successfully using UPFP algorithm
Total number of Uncertain Periodic-Frequent Patterns: 532
Total number of False Patterns: 113
Total Memory in USS: 990633984
Total Memory in RSS 1049882624
Total ExecutionTime in ms: 45.799009561538696
Periodic frequent patterns were generated successfully using UPFP algorithm
Total number of Uncertain Periodic-Frequent Patterns: 477
Total number of False Patterns: 58
Total Memory in USS: 976277504
Total Memory in RSS 1035530240
Total ExecutionTime in ms: 35.953229904174805
Periodic frequent patterns were generated successfully using UPFP algorithm
Total number of Uncertain Periodic-Frequent Patterns: 435
Total number of False Patterns: 10
Total Memory in USS: 962727936
Total Memory in RSS 1021984768
Total ExecutionTime in ms: 26.950753927230835
Periodic frequent patterns were generated successfully using UPFP algorithm
Total number of Uncertain Periodic-Frequent Patterns: 398
Total number of False Patterns: 3
Total Memory in USS: 948744192
Total Memory in RSS 1007996928
Total ExecutionTime in ms: 25.406779050827026
Periodic frequent patterns were generated successfully using UPFP algorithm

Total number of Uncertain Periodic-Frequent Patterns: 363
 Total number of False Patterns: 2
 Total Memory in USS: 934080512
 Total Memory in RSS 993333248
 Total ExecutionTime in ms: 23.03202986717224
 Periodic frequent patterns were generated successfully using UPFP algorithm
 Total number of Uncertain Periodic-Frequent Patterns: 337
 Total number of False Patterns: 0
 Total Memory in USS: 921899008
 Total Memory in RSS 981020672
 Total ExecutionTime in ms: 22.09625244140625

5 Step 5: Print all the results

```
[5]: print(result)
```

	algorithm	minSup	maxPer	patterns	runtime	memory
0	UPFPGrowth++	400	8000	609	64.347102	1006804992
1	UPFPGrowth++	500	8000	532	37.519752	1006247936
2	UPFPGrowth++	600	8000	478	30.614928	991379456
3	UPFPGrowth++	700	8000	435	24.866676	977100800
4	UPFPGrowth++	800	8000	398	23.889670	962265088
5	UPFPGrowth++	900	8000	363	22.860890	946683904
6	UPFPGrowth++	1000	8000	337	21.281519	933785600
7	UPFPGrowth	400	8000	610	76.655918	1001152512
8	UPFPGrowth	500	8000	532	45.799010	990633984
9	UPFPGrowth	600	8000	477	35.953230	976277504
10	UPFPGrowth	700	8000	435	26.950754	962727936
11	UPFPGrowth	800	8000	398	25.406779	948744192
12	UPFPGrowth	900	8000	363	23.032030	934080512
13	UPFPGrowth	1000	8000	337	22.096252	921899008

```
[ ]:
```