

congestion_minSup-Copy1

March 22, 2023

```
[1]: import spatialFrequent as sp
import spatialFrequentNew as spn
import pandas as pd
```

```
[2]: inputFile = 'congestion_uncertain.txt'
separator = ' '
minimumSupportCountList = [100, 200, 300, 400, 500, 600]

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

```
[3]: algorithm = 'GPFP-Miner-New' #specify the algorithm name
for minSupCount in minimumSupportCountList:
    obj1 = spn.GFPGrowth(inputFile, nFile='congestion_new_nei_1.csv', '
    ↪minSup=minSupCount, sep=separator)
    obj1.startMine()
    st = 'congestion_output_' + str(minSupCount)
    obj1.savePatterns(st)
    #store the results in the data frame
    result.loc[result.shape[0]] = [algorithm, minSupCount, len(obj1.
    ↪getPatterns()), obj1.getRuntime(), obj1.getMemoryRSS()]
```

Frequent patterns were generated from uncertain databases successfully using GFP algorithm

Frequent patterns were generated from uncertain databases successfully using GFP algorithm

Frequent patterns were generated from uncertain databases successfully using GFP algorithm

Frequent patterns were generated from uncertain databases successfully using GFP algorithm

Frequent patterns were generated from uncertain databases successfully using GFP algorithm

Frequent patterns were generated from uncertain databases successfully using GFP algorithm

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

	algorithm	minSup	patterns	runtime	memory
0	GPFP-Miner-New	100	2511	846.920032	885944320
1	GPFP-Miner-New	200	1439	844.320011	881172480
2	GPFP-Miner-New	300	1045	842.636043	874807296
3	GPFP-Miner-New	400	798	842.073059	868855808
4	GPFP-Miner-New	500	648	842.824464	864272384
5	GPFP-Miner-New	600	545	850.256031	859783168

```
[ ]: import PUFgrowth as pf
algorithm = 'PUFGrowth' #specify the algorithm name
minimumSupportCountList = [500, 600]
for minSupCount in minimumSupportCountList:
    obj1 = pf.PUFGrowth(inputFile, minSup=minSupCount, sep=separator)
    obj1.startMine()
    #store the results in the data frame
    result.loc[result.shape[0]] = [algorithm, minSupCount, len(obj1.
    ↪getPatterns()), obj1.getRuntime(), obj1.getMemoryRSS()]
```

IOStream.flush timed out

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

	algorithm	minSup	patterns	runtime	memory
0	GPFP-Miner-New	100	2511	846.920032	885944320
1	GPFP-Miner-New	200	1439	844.320011	881172480
2	GPFP-Miner-New	300	1045	842.636043	874807296
3	GPFP-Miner-New	400	798	842.073059	868855808
4	GPFP-Miner-New	500	648	842.824464	864272384
5	GPFP-Miner-New	600	545	850.256031	859783168

```
[ ]:
```