Pollution_experiments-Copy2

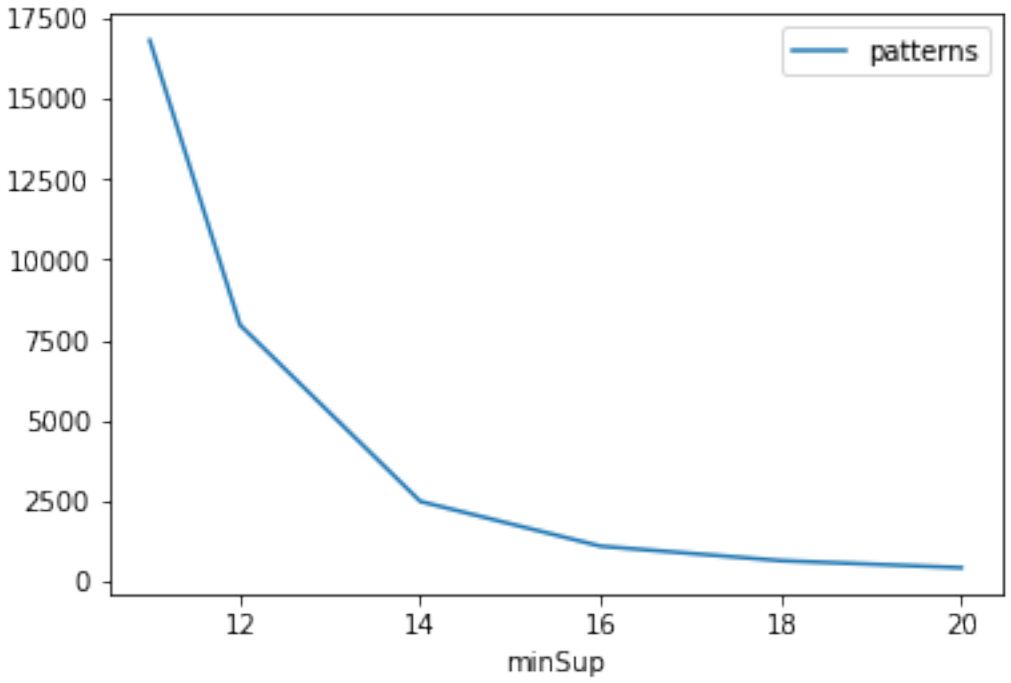
June 3, 2023

[1]:
```
import FFI_newVersion_pollution  as alg
import pandas as pd
inputFile = 'updated_pollution_24.txt'

minimumSupportCountList = [11, 12, 14, 16, 18, 20]  #Users can also specify this constraint between 0 to 1.
seperator = '\t'
result = pd.DataFrame(columns=['algorithm', 'minSup', 'patterns', 'runtime', 'memory'])
#initialize a data frame to store the results of FFIMiner algorithm

algorithm = 'FFI'  #specify the algorithm name
for minSupCount in minimumSupportCountList:
    obj = alg.FFIMiner(iFile=inputFile, minSup=minSupCount, sep=seperator)
    obj.startMine()
    #obj.save('pollution_patterns.txt')
    #store the results in the data frame
    result.loc[result.shape[0]] = [algorithm, minSupCount, len(obj.getPatterns()), obj.getRuntime(), obj.getMemoryRSS()]

print(result)
```

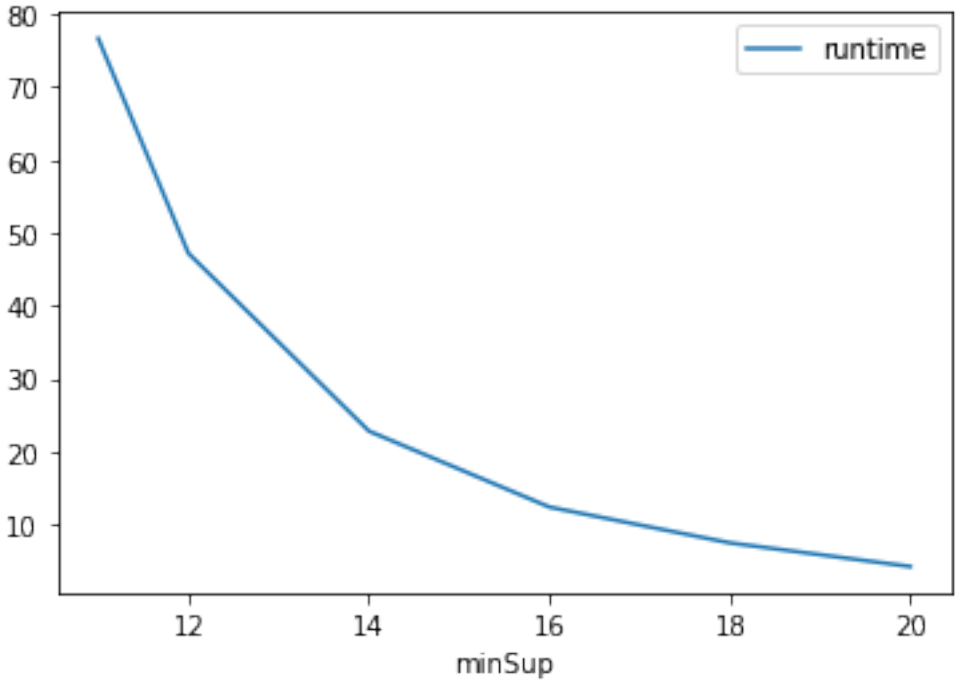
```
146
2882
146
2882
146
2882
146
2882
146
2882
146
2882
  algorithm  minSup  patterns    runtime     memory
0       FFI      11     16814  76.689468  144478208
1       FFI      12      7977  47.183742  143900672
2       FFI      14      2494  22.769460  143278080
```

```
3       FFI      16      1104  12.337866  143294464
4       FFI      18       655   7.420281  142262272
5       FFI      20       436   4.172746  142262272
```

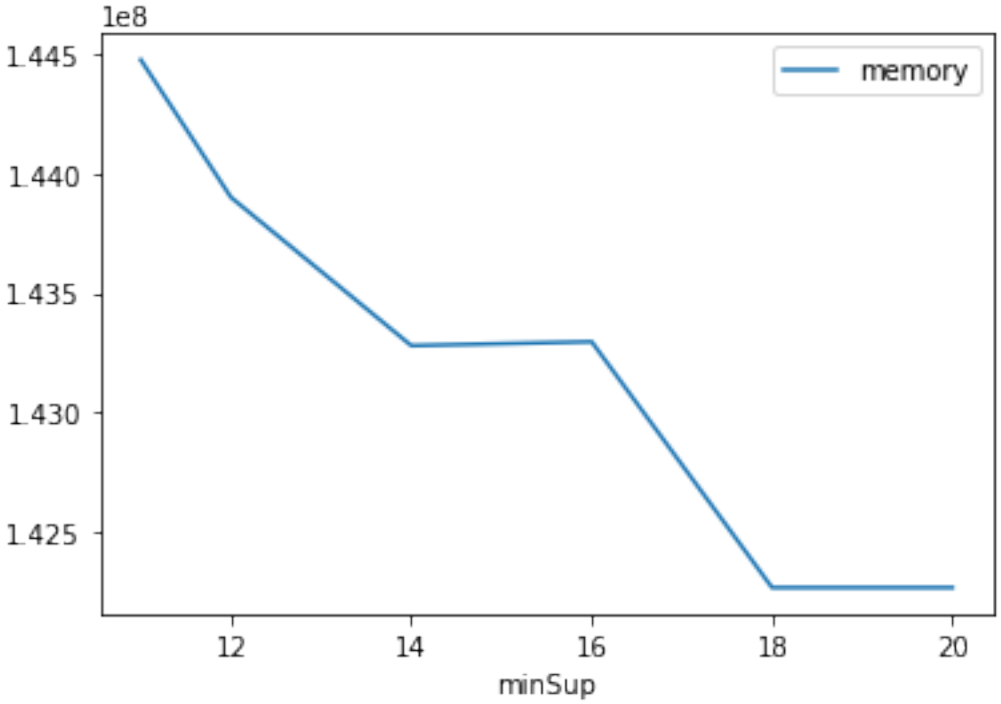
[2]:
```
from PAMI.extras.graph import plotLineGraphsFromDataFrame as plt
ab = plt.plotGraphsFromDataFrame(result)
ab.plotGraphsFromDataFrame() #drawPlots()
```




```
Graph for No Of Patterns is successfully generated!
```




```
Graph for Runtime taken is successfully generated!
```



```
Graph for memory consumption is successfully generated!
```

[3]:
```
from PAMI.extras.graph import generateLatexFileFromDataFrame as gdf
gdf.generateLatexCode(result)
```

```
Latex files generated successfully
```

[ ]:
```

```