Mining Frequent Spatial Patterns in Transactional Databases

What is frequent spatial pattern mining?

Frequent pattern mining aims to discover all interesting patterns in a transactional database that have **support** no less than the user-specified **minimum support** (**minSup**) constraint and **distance** no greater than user-specified **maximum distance** (**maxDist**). The **minSup** controls the minimum number of transactions that a pattern must appear in a database. The **maxDist** controls the maximum distance between two items in pattern should be less.

Reference: R. Uday Kiran, Sourabh Shrivastava, Philippe Fournier-Viger, Koji Zettsu, Masashi Toyoda, and Masaru Kitsuregawa. 2020. Discovering Frequent Spatial Patterns in Very Large Spatiotemporal Databases. In Proceedings of the 28th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '20). Association for Computing Machinery, New York, NY, USA, 445–448. https://doi.org/10.1145/3397536.3422206

What is a transactional database?

A transactional database is a collection of transactions, where each transaction contains a transaction-identifier and a set of items.
A hypothetical transactional database containing the items ***a, b, c, d, e, f, and g*** as shown below

tid	Transactions
1	a b c g
2	b c d e
3	a b c d
4	a c d f
5	a b c d g
6	c d e f
7	a b c d
8	a e f
9	a b c d
10	b c d e

Note: Duplicate items must not exist in a transaction.

What is acceptable format of a transactional databases in PAMI?

Each row in a transactional database must contain only items. PAMI algorithms implicitly consider the row number as the transactional-identifier to reduce storage and processing costs.

a b c g
b c d e
a b c d
a c d f
a b c d g
c d e f
a b c d
a e f
a b c d
b c d e

What is the spatial database?

Spatial database contain the spatial (neighbourhood) information of items. It contains the items and its nearset neighbours satisfying the **maxDist** constraint.
A hypothetical spatial database containing items ***a, b, c, d, e, f and g*** and neighbours respectively is shown below.

Items	neighbours
a	b, c, d
b	a, e, g
c	a, d
d	a, c
e	b, f
f	e, g
g	b, f

Understanding the statisctics of database

To understand about the database. The below code will give the detail about the transactional database.

- Total number of transactions (Database size)
- Total number of unique items in database
- Minimum lenth of transaction that existed in database
- Average length of all transactions that exists in database
- Maximum length of transaction that existed in database
- Standard deviation of transaction length
- Variance in transaction length
- Sparsity of database

The sample code

In [ ]:
```
import PAMI.extras.dbStats.transactionalDatabaseStats as stats
obj = stats.transactionalDatabaseStats('sampleInputFile.txt', ' ')
obj.run()
obj.printStats()
```

What are the input parameters?

Algorithms to mine the frequent spatial patterns requires transactional database, spatial database and minSup (specified by user).

- Transactional database can be provided in following formats:

  > - String : E.g., 'transactionalDatabase.txt'
  > - URL : E.g., https://u-aizu.ac.jp/~udayrage/datasets/transactionalDatabases/transactional_T10
  > - DataFrame. Please note that dataframe must contain the header titled 'Transactions'

- Spatial database can be provided in following formats:

  > - String : E.g., 'spatialDatabase.txt'
  > - URL : E.g., https://u-aizu.ac.jp/~udayrage/datasets/transactionalDatabases/transactional_T10
  > - DataFrame. Please note that dataframe must contain the header titled 'Transactions'

- **minSup**
  specified in

  > - **count (beween 0 to length of a database)** or
  > - [0, 1]

- **seperator**
  default seperator is '\t' (tab space)

How to store the output of a frequent pattern mining algorithm?

The patterns discovered by a frequent pattern mining algorithm can be saved into a file or a data frame.

How to run the frequent spatial pattern algorithm in a terminal?

- Download the PAMI source code from github.
- Unzip the PAMI source code folder and enter into frequent spatial pattern folder.
- Enter into frequentSpatialPattern folder
- You will find folder like **basic**
- Enter into a specific folder of your choice and execute the following command on terminal.

syntax: python3 algorithmName.py `<path to the input file>` `<path to the output file>` `<path to the neighbour file>` `<minSup>` `<seperator>`

Example: python3 `FSPGrowth.py` `inputFile.txt` `outputFile.txt` `neighbourFile.txt` 3 `' '`

How to execute a frequent pattern mining algorithm in a Jupyter Notebook?

- Install the PAMI package from the PYPI repository by executing the following command: **pip3 install PAMI**
- Run the below sample code by making necessary changes

In [ ]:
```
import PAMI.frequentSpatialPattern.basic.FSPGrowth as alg

iFile = 'sampleInputFile.txt'  #specify the input transactional database
minSup = 5  #specify the minSupvalue <br>
seperator = ' ' #specify the seperator. Default seperator is tab space. <
oFile = 'frequentPatterns.txt'   #specify the output file name<br>
nFile = 'sampleNeighbourFile.txt'  #specify the neighbour file name<br>

obj = alg.FSPGrowth(iFile, nFile, minSup, seperator) #initialize the algo
obj.startMine()                         #start the mining process <br>
obj.savePatterns(oFile)                 #store the patterns in file <br>
df = obj.getPatternsAsDataFrame()       #Get the patterns discovered into a
obj.printStats()                        #Print the statistics of mining pro
```

The frequentPatterns.txt file contains the following patterns (*format:* pattern:support):!cat frequentPatterns.txt

In [ ]:
```
!cat frequentPatterns.txt
```

a: 7
b: 7
c: 9
d: 8
a b: 5
a c: 6
a d: 5
c d: 8

The dataframe containing the patterns is shown below:

In [ ]:
```
df
```

	Patterns	Support
0	a	7
1	b	7
2	c	9
3	d	8
4	a b	5
5	a c	6
6	a d	5
7	c d	8