Mining Geo Referenced Periodic-Frequent Patterns in Temporal Databases

What is geo referenced periodic-frequent pattern mining?

Geo Referenced Periodic-Frequent pattern mining aims to discover all interesting patterns in a temporal database that have **support** no less than the user-specified **minimum support** (**minSup**) constraint, **periodicity** no greater than user-specified **maximum periodicity** (**maxPer**) constraint and **distance** between two items is no less than **maximum distance** (**maxDist**). The **minSup** controls the minimum number of transactions that a pattern must appear in a database and the **maxPer** controls the maximum time interval within which a pattern must reappear in the database.

What is the temporal database?

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

TS	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.

Acceptable format of temporal databases in PAMI

Each row in a temporal database must contain timestamp and items.

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

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.

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
- Minimum periodicity exists in database
- Average periodicity exists in database
- Maximum periodicity exists in database
- Standard deviation of transaction length
- Variance in transaction length
- Sparsity of database

The sample code

```
import PAMI.extras.dbStats.temporalDatabaseStats as stats

obj = stats.temporalDatabaseStats('sampleInputFile.txt', ' ')
obj.run()
obj.printStats()
```

What is the input to geo referenced periodic-frequent spatial pattern mining algorithms

Algorithms to mine the geo referenced periodic-frequent patterns requires temporal database, neighbour database, minSup and maxPer (specified by user).

- Input temporal database is accepted following formats:

  > - In string format (`/Users/Likhitha/Downlaods/sampleInputFile.txt`)
  > - In URL format (`https://www.u-aizu.ac.jp/~udayrage/datasets/transactionalDatabases/tra`
  > - In DataFrame format (dataframe variable with heading `TS` and `Transactions`

- Spatial database in following formats:

  > - In string format (`/Users/Likhitha/Downlaods/sampleNeighbourFile.txt`)
  > - In URL format (`https://www.u-aizu.ac.jp/~udayrage/datasets/transactionalDatabases/tra`
  > - In DataFrame format (dataframe variable with headings `item` and `Neighbours`)

- minSup should be mentioned in **count (beween 0 to length of database)** or __percentage (multiplied with length of database)
- maxPer should be mentioned in **count (beween 0 to length of database)** or __percentage (multiplied with length of database)
- seperator (delimiter used in input file) default delimiter is `\t`

What is the output of geo referenced periodic-frequent spatial pattern mining algorithms

The output of these algorithms is in two ways:

> - Save the patterns in user specified output file.
> - Returns the patterns in dataframe variable.

How to run the geo referenced periodic-frequent spatial pattern algorithm in terminal

- Download the code from github.
- Navigate to PAMI folder where you downloaded the file.
- Go to periodicFrequentSpatialPattern folder

Execute the following command on terminal.

python3 algorithmName.py `path of Sample input file` `path of output file` `path of neighbour file` $minSup$ $maxPer$ `seperator`

Sample command to execute the GPFPMiner algorithm in periodicFrequentSpatialPattern/basic folder

python3 `GPFPMiner.py` `/Users/Donwloads/inputFile.txt` `/Users/Downloads/outputFile.txt` `3` `4` `' '`

How to implement the GPFPMiner algorithm by importing PAMI package

Import the PAMI package executing: **pip3 install PAMI**

Run the below sample code by making simple changes

- Replace sampleInputFile name or path in place of iFile and sampleOutputFile name or path in place of oFile
- Specify the minSup (like 10 or 0.1) in place of minSup
- Specify the maxPer (like 10 or 0.1) in place of maxPer
- Specify the seperator of input file after maxPer. (If no seperator is specified the default tab seperator is considered for input file)

import PAMI.geoReferencedPeriodicFrequentPattern.GPFPMiner as alg
obj = alg.GPFPMiner(iFile, nFile, minSup, maxPer, sep)
obj.startMine()
obj.savePatterns(oFile) (to store the patterns in file)
Df = obj.getPatternsAsDataFrame() (to store the patterns in dataframe)
obj.printStats() (to print the no of patterns, runtime and memory consumption details)

What is the output of geo referenced periodic-frequent pattern mining algorithms

Returns the geo referenced pattern with support and periodicity respectively with $minSup=4$ and $maxPer=3$

The output in file format:

d c a : 5: 3
c d : 8: 2
c a : 6: 2
c : 9: 2
d a : 5: 3
d : 8: 2
b a : 5: 2
b : 7: 2
a : 7: 2

The output in DataFrame format:

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