Mining Periodic Frequent Patterns in Uncertain Temporal Databases

What is periodic-frequent pattern mining?

Periodic-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 **periodicity** no greater than user-specified **maximum period** (**maxPer**). The **minSup** controls the minimum number of transactions that a pattern must appear in a database and **maxPer** controls the maximum interval time a pattern must reappear.

What is the uncertain temporal database?

A temporal database is a collection of transactions at different timestamps, where each transaction contains a timestamp and a set of items with their respective uncertain value.
A hypothetical transactional database containing the items ***a, b, c, d, e, f, and g*** as shown below

TS	Transactions
1	a(0.4) b(0.5) c(0.2) g(0.1)
2	b(0.2) c(0.3) d(0.4) e(0.2)
3	a(0.3) b(0.1) c(0.3) d(0.4)
4	a(0.2) c(0.6) d(0.2) f(0.1)
5	a(0.3) b(0.2) c(0.4) d(0.5) g(0.3)
6	c(0.2) d(0.7) e(0.34) f(0.2)
7	a(0.6) b(0.4) c(0.3) d(0.2)
8	a(0.2) e(0.2) f(0.2)
9	a(0.1) b(0.3) c(0.2) d(0.4)
10	b(0.3) c(0.2) d(0.1) e(0.6)

Note: Duplicate items must not exist in a transaction.

Acceptable format of uncertain temporal databases in PAMI

Each row in a transactional database must contain timestamp and items with their respective uncertain values.

1 a(0.4) b(0.5) c(0.2) g(0.1)
2 b(0.2) c(0.3) d(0.4) e(0.2)
3 a(0.3) b(0.1) c(0.3) d(0.4)
4 a(0.2) c(0.6) d(0.2) f(0.1)
5 a(0.3) b(0.2) c(0.4) d(0.5) g(0.3)
6 c(0.2) d(0.7) e(0.34) f(0.2)
7 a(0.6) b(0.4) c(0.3) d(0.2)
8 a(0.2) e(0.2) f(0.2)
9 a(0.1) b(0.3) c(0.2) d(0.4)
10 b(0.3) c(0.2) d(0.1) e(0.6)

What is the input to uncertain periodic-frequent pattern mining algorithms

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

- Temporal database in following formats:

  > - In string format (`/Users/Likhitha/Downlaods/sampleInputFile.txt')
  > - In URL format (`https://www.u-aizu.ac.jp/~udayrage/datasets/transactionalDatabases/transactional_T10
  > - In DataFrame format (dataframe variable with heading `TS` and `Transactions` which contains only items and `uncertain` which contains uncertain values of each item in transaction respectively)

- 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)
- Specify the seperator of input file after maxPer. (If no seperator is specified the default tab seperator is considered for input file)

What is the output of uncertain periodic-frequent pattern mining algorithms

The output of these algorithms is in two ways:

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

How to run the frequent pattern algorithm in terminal

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

Execute the following command on terminal.

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

Sample command to execute the UPFPGrowth code in uncertainPeriodicFrequentPattern/basic folder

python3 `UPFPGrowth.py` `/Users/Donwloads/inputFile.txt` `/Users/Downloads/outputFile.txt` 0.05 4 `' '`

How to implement the uncertain periodic-frequent 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 seperator of input file after minSup. (If no seperator is specified the default tab seperator is considered for input file)

import PAMI.uncertainPeriodicFrequentPattern.basic.UPFPGrowth as alg
obj = alg.UPFPGrowth(iFile, minSup, maxPer, ',')
obj.startMine()
obj.savePatterns(oFile) (to store the patterns in file).
Df = obj.getPatternsAsDataFrame() (to store the patterns in dataframe)
obj.printStats()

What is the output of periodic-frequent pattern mining algorithms

Returns the pattern and support respectively for $minSup=0.7$ and $maxPer=2$

The output in file format:

e :1.3399999999999999:4
b :2.0:2
a :2.0999999999999996:2
c :2.7:2
d :2.9000000000000004:2
c d :0.8600000000000001:2

The output in DataFrame format:

	Patterns	Support	Periodicity
0	e	1.34	4
1	b	2.00	2
2	a	2.09	2
3	c	2.70	2
4	d	2.90	2
5	c d	0.86	2