

Mining Periodic-Frequent Patterns in Temporal Fuzzy Databases

What is fuzzy periodic-frequent pattern mining?

Fuzzy Periodic-Frequent pattern mining aims to discover all fuzzy periodic frequent patterns in a temporal database that have **support** no less than the user-specified **minimum support (minSup)** constraint and **periodicity** no greater than user-specified **maximum periodicity (maxPer)** constraint. 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.

Research paper: R. U. Kiran et al., "Discovering Fuzzy Periodic-Frequent Patterns in Quantitative Temporal Databases," 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2020, pp. 1-8, doi: 10.1109/FUZZ48607.2020.9177579.

What is a temporal fuzzy database?

A fuzzy temporal database is a collection of transactions at a particular timestamp, where each transaction contains a timestamp and a set of items.

A hypothetical fuzzy temporal database containing the items **a, b, c, d, e, f, and g** as shown below

TS	Transactions	Item	Profit
1	(a,2) (b,3) (c,1) (g,1)	a	4
2	(b,3) (c,2) (d,3) (e,2)	b	3
3	(a,2) (b,1) (c,3) (d,4)	c	6
4	(a,3) (c,2) (d,1) (f,2)	d	2
5	(a,3) (b,1) (c,2) (d,1) (g,2)	e	5
6	(c,2) (d,2) (e,3) (f,1)	f	2
7	(a,2) (b,1) (c,1) (d,2)	g	3
8	(a,1) (e,2) (f,2)		
9	(a,2) (b,2) (c,4) (d,2)		
10	(b,3) (c,2) (d,2) (e,2)		

Note: Duplicate items must not exist in a transaction.

Acceptable format of fuzzy 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
```

Understanding the statistics 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 length 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 below sample code prints the statistical details of a database.

```
In [ ]: import PAMI.extras.dbStats.temporalDatabaseStats as stats

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

What are the input parameters?

- **Fuzzy database**

Acceptable formats:

- String : E.g., 'fuzzyDatabase.txt'
- URL : E.g., https://u-aizu.ac.jp/~udayrage/datasets/transactionalDatabases/transactional_T10
- DataFrame with the header titled 'Transactions', 'Utility' and 'TransactionUtility'

- **minSup**

specified in

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

- **maxPer**

specified in

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

- **seperator**

default seperator is '\t' (tab space)

How to store the output of a fuzzy periodic frequent patternn mining algorithm?

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

How to run the fuzzy periodic-frequent pattern algorithm in a terminal?

- Download the PAMI source code from github.
- Unzip the PAMI source code folder and enter in to fuzzy periodic frequent pattern
- Enter into fuzzyPeriodicFrequentPattern folder
- Enter into the folder and execute the following command on terminal.

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

Example: python3 FPFPMiner.py inputFile.txt outputFile.txt 3 4 '

How to execute a Fuzzy Periodic Frequent mining algorithm in 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 [1]: import PAMI.fuzzyPeriodicFrequentPattern.basic.FPFPMiner as alg

iFile = 'sample_Input.txt' #specify the input temporal database <br>
minSup = 4 #specify the minSupvalue <br> #specify the maxPerAllConfVa
seperator = ' ' #specify the seperator. Default seperator is tab space. <
maxPer = 3
oFile = 'FuzzyPeriodicPatterns.txt' #specify the output file name<br>

obj = alg.FPFPMiner(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() #(to print the no of patterns, runtime and memory consu
```

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

```
In [2]: !cat 'FuzzyPeriodicPatterns.txt'
```

a.L : 5.4 : 2

b.L : 5.6 : 2

b.L d.L : 4.1999999999999999 : 2

b.L c.L : 4.6 : 2

d.L : 6.1999999999999999 : 2

d.L c.L : 5.4 : 2

c.L : 7.0 : 2

The dataframe containing the patterns is shown below:

In [3]:

Df

Out[3]:

	Patterns	Support
0	a.L	5.4 : 2\n
1	b.L	5.6 : 2\n
2	b.L d.L 4.1999999999999999	: 2\n
3	b.L c.L	4.6 : 2\n
4	d.L 6.1999999999999999	: 2\n
5	d.L c.L	5.4 : 2\n
6	c.L	7.0 : 2\n