Mining Fuzzy Frequent Patterns in Fuzzy Databases

What is Fuzzy Frequent pattern mining?

Fuzzy Frequent itemset (FFI)-Miner aims to mine complete set of FFIs without candidate generation and items with support no less than the user-specified **minimum support**(minSup) constraint. The **minSup** controls the minimum number of transactions that a pattern must appear in a database.

Reference : Lin, Chun-Wei & Li, Ting & Fournier Viger, Philippe & Hong, Tzung-Pei. (2015). A fast Algorithm for mining fuzzy frequent itemsets. Journal of Intelligent & Fuzzy Systems. 29. 2373-2379. 10.3233/IFS-151936.

What is a Fuzzy database?

A fuzz database is a collection of transaction, where each transaction contains a set of items called **fuzzy terms** and a positive integer called **fuzzy value** respectively. A hypothetical fuzzy database with items ***a, b, c, d, e, f and g*** is shown below.

Transactions	fuzzy values
a b c g	5 4 3 2
b c d e	5 2 9 3
a b c d	2 3 5 6
a c d f	1 3 4 6
a b c d g	2 5 3 6 1
c d e f	2 3 4 5
a b c d	5 4 3 2
a e f	4 8 3
a b c d	7 4 9 8
b c d e	5 9 10 24

Note: Duplicate items must not exist in a transaction.

What is acceptable format of a fuzzy databases in PAMI?

The format we accept for fuzzy database is the same as the utility database format.

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

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 utility value exists in database
- Average utility exists in database
- Maximum utility 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 [2]:
```
import PAMI.extras.dbStats.utilityDatabaseStats as stats
obj = stats.utilityDatabaseStats('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**

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

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

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

How to run the fuzzy frequent pattern mining algorithms in a terminal?

- Download the PAMI source code from github.
- Unzip the PAMI source code folder and enter into fuzzyFrequentPatterns folder.
- Enter into fuzzyFrequentPatterns folder
- You will find folder like **basic**
- Enter into a specific folder and execute the following command on terminal.

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

Example: python3 `FFIMiner.py` `inputFile.txt` `outputFile.txt` `$5$` `'	'`

How to execute a fuzzy 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 [6]:
```
import PAMI.fuzzyFrequentPatterns.basic.FFIMiner as alg

iFile = 'sample_Input.txt'  #specify the input utility database <br>
minSup = 5  #specify the minSupvalue <br>
seperator = ' ' #specify the seperator. Default seperator is tab space. <
oFile = 'fuzzyPatterns.txt'   #specify the output file name<br>

obj = alg.FFIMiner(iFile, minSup, seperator) #initialize the algorithm <b
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 [8]:
```
!cat fuzzyPatterns.txt
```

```
a.L : 5.4
b.L : 5.6
d.L : 6.199999999999999
d.L c.L : 5.4
c.L : 7.0
```

The dataframe containing the patterns is shown below:

In [ ]:
```
df
```

Out[ ]:

	Patterns	Support
0	a.L	5.4\n
1	b.L	5.6\n
2	d.L	6.199999999999999\n
3	d.L c.L	5.4\n
4	c.L	7.0\n