Mining High-Utility Frequent Patterns in Utility Databases

What is High-Utility Frequent pattern mining?

High utility frequent pattern mining aims to discover all the patterns with ***utility*** of pattern is no less than user-specified ***minimum utility*** (***minutil***) and ***support*** is no less than user-specified ***minimum support*** (***minSup***).

What is the utility database?

A utility database is a collection of transaction, where each transaction contains a set of items and a positive integer called ***internal utility*** respectively. And each unique item in database is also associated with another positive number called ***external utility***.
A hypothetical utility database with items ***a, b, c, d, e, f and g*** and its ***internal utility*** is shown below at right side and items with its ***external utility*** is presented at left side.

Transactions	Item	Profit
(a,2) (b,3) (c,1) (g,1)	a	4
(b,3) (c,2) (d,3) (e,2)	b	3
(a,2) (b,1) (c,3) (d,4)	c	6
(a,3) (c,2) (d,1) (f,2)	d	2
(a,3) (b,1) (c,2) (d,1) (g,2)	e	5
(c,2) (d,2) (e,3) (f,1)	f	2
(a,2) (b,1) (c,1) (d,2)	g	3
(a,1) (e,2) (f,2)		
(a,2) (b,2) (c,4) (d,2)		
(b,3) (c,2) (d,2) (e,2)		

Note: Duplicate items must not exist in a transaction.

Acceptable format of utility databases in PAMI

Each row in a utility database must contain only items, total sum of utilties and utility values.

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 sample code

In [ ]:

```
import PAMI.extras.dbStats.utilityDatabaseStats as stats
obj = stats.utilityDatabaseStats('sampleInputFile.txt', ' ')
obj.run()
obj.printStats()
```

What is the input parameters?

Algorithms to mine the high-utility patterns requires utility database, minUtil (specified by user).

- Utility database can be provided in following formats:

  > - String : E.g., 'utilityDatabase.txt'
  > - URL : E.g., https://u-aizu.ac.jp/~udayrage/datasets/transactionalDatabases/transactional_T10
  > - In DataFrame format (dataframe variable with heading `Transactions` , `Utilities` and `TransactionUtility`

- **minUtil**
  specified in

  > - [0, 1]

- **minSup**
  specified in

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

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

How to run the high-utility frequent pattern algorithm in terminal

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

And execute the following command on terminal.

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

__Examplepython3 `HUFIM.py` `inputFile.txt` `outputFile.txt` $20$ $5$ `' '`

How to implement the HUFIM 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.highUtilityFrequentPatterns.basic.HUFIM as alg

iFile = 'sample_Input.txt'  #specify the input transactional database <br
minUtil = 20  #specify the minSupvalue <br>
minSup = 5  #specify the minSupvalue <br>
seperator = ' '  #specify the seperator. Default seperator is tab space. <
oFile = 'utilityfrequentPatterns.txt'   #specify the output file name<br>

obj = alg.HUFIM(iFile, minUtil, minSup, seperator) #initialize the algori
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 utilityfrequentPatterns.txt file contains the following patterns (*format:* pattern:utility:support):!cat utilityfrequentPatterns.txt

In [3]:

```
!cat utilityfrequentPatterns.txt
```

```
c d : 35:8
c d a : 34:5
c d b : 39:6
c a : 27:6
c a b : 30:5
c b : 29:7
d a : 22:5
d b : 25:6
```

In [4]:

```
df
```

Out[4]:

	Patterns	Utility:Support
0	c d	35:8
1	c d a	34:5
2	c d b	39:6
3	c a	27:6
4	c a b	30:5
5	c b	29:7
6	d a	22:5
7	d b	25:6