关联分析

In [6]:

```
# 检查已安装的package
%pip list
```

```
Package                       Version
----------------------------- --------------

aiobotocore                   2.4.2
aiofiles                      22.1.0
aiohttp                       3.8.3
aioitertools                  0.7.1
aiosignal                     1.2.0
aiosqlite                     0.18.0
alabaster                     0.7.12
anaconda-anon-usage           0.4.4
anaconda-catalogs             0.2.0
anaconda-client               1.11.3
anaconda-cloud-auth           0.5.0
anaconda-navigator            2.5.0
anaconda-project              0.11.1
annotated-types               0.6.0
anyio                         3.5.0
appdirs                       1.4.4
apyori                        1.1.2
argon2-cffi                   21.3.0
argon2-cffi-bindings          21.2.0
arrow                         1.2.3
astroid                       2.14.2
astropy                       5.1
asttokens                     2.0.5
async-timeout                 4.0.2
atomicwrites                  1.4.0
attrs                         22.1.0
Automat                       20.2.0
autopep8                      1.6.0
Babel                         2.11.0
backcall                      0.2.0
backports.functools-lru-cache 1.6.4
backports.tempfile            1.0
backports.weakref             1.0.post1
bcrypt                        3.2.0
beautifulsoup4                4.12.2
binaryornot                   0.4.4
black                         0.0
bleach                        4.1.0
bokeh                         3.1.1
boltons                       23.0.0
boto3                         1.24.28
botocore                      1.27.59
Bottleneck                    1.3.5
brotlipy                      0.7.0
certifi                       2024.7.4
cffi                          1.15.1
chardet                       4.0.0
charset-normalizer            2.0.4
click                         8.0.4
cloudpickle                   2.2.1
clyent                        1.2.2
colorama                      0.4.6
colorcet                      3.0.1
comm                          0.1.2
conda                         23.7.4
conda-build                   3.25.0
conda-content-trust           0.1.3
conda_index                   0.2.3
conda-libmamba-solver         23.5.0
conda-pack                    0.6.0
conda-package-handling        2.1.0
conda_package_streaming       0.8.0
conda-repo-cli                1.0.41
conda-token                   0.4.0
conda-verify                  3.4.2
constantly                    15.1.0
contourpy                     1.0.5
```

```
cookiecutter                  1.7.3
cryptography                  39.0.1
cssselect                     1.1.0
cycler                        0.11.0
cytoolz                       0.12.0
daal4py                       2023.1.1
dask                          2023.6.0
datashader                    0.15.0
datashape                     0.5.4
debugpy                       1.5.1
decorator                     5.1.1
defusedxml                    0.7.1
diff-match-patch              20200713
dill                          0.3.6
distributed                   2023.6.0
docstring-to-markdown         0.11
docutils                      0.18.1
entrypoints                   0.4
et-xmlfile                    1.1.0
executing                     0.8.3
fastjsonschema                2.16.2
filelock                      3.9.0
flake8                        6.0.0
Flask                         2.2.2
fonttools                     4.25.0
frozenlist                    1.3.3
fsspec                        2023.3.0
future                        0.18.3
gensim                        4.3.0
glob2                         0.7
greenlet                      2.0.1
h5py                          3.7.0
HeapDict                      1.0.1
holoviews                     1.16.2
hvplot                        0.8.4
hyperlink                     21.0.0
idna                          3.4
imagecodecs                   2021.8.26
imageio                       2.26.0
imagesize                     1.4.1
imbalanced-learn              0.10.1
importlib-metadata            6.0.0
incremental                   21.3.0
inflection                    0.5.1
iniconfig                     1.1.1
intake                        0.6.8
intervaltree                  3.1.0
ipykernel                     6.19.2
ipython                       8.12.0
ipython-genutils              0.2.0
ipywidgets                    8.0.4
isort                         5.9.3
itemadapter                   0.3.0
itemloaders                   1.0.4
itsdangerous                  2.0.1
jaraco.classes                3.2.1
jedi                          0.18.1
jellyfish                     0.9.0
Jinja2                        3.1.2
jinja2-time                   0.2.0
jmespath                      0.10.0
joblib                        1.2.0
json5                         0.9.6
jsonpatch                     1.32
jsonpointer                   2.1
jsonschema                    4.17.3
jupyter                       1.0.0
jupyter_client                7.4.9
jupyter-console               6.6.3
```

```
jupyter_core                  5.3.0
jupyter-events                0.6.3
jupyter_server                2.5.0
jupyter_server_fileid         0.9.0
jupyter_server_terminals      0.4.4
jupyter_server_ydoc           0.8.0
jupyter-ydoc                  0.2.4
jupyterlab                    3.6.3
jupyterlab-pygments           0.1.2
jupyterlab_server             2.22.0
jupyterlab-widgets            3.0.5
keyring                       23.13.1
kiwisolver                    1.4.4
lazy_loader                   0.2
lazy-object-proxy             1.6.0
libarchive-c                  2.9
libmambapy                    1.4.1
linkify-it-py                 2.0.0
llvmlite                      0.40.0
lmdb                          1.4.1
locket                        1.0.0
lxml                          4.9.2
lz4                           4.3.2
Markdown                      3.4.1
markdown-it-py                2.2.0
MarkupSafe                    2.1.1
matplotlib                    3.7.1
matplotlib-inline             0.1.6
mccabe                        0.7.0
mdit-py-plugins               0.3.0
mdurl                         0.1.0
menuinst                      1.4.19
mistune                       0.8.4
mkl-fft                       1.3.6
mkl-random                    1.2.2
mkl-service                   2.4.0
more-itertools                8.12.0
mpmath                        1.2.1
msgpack                       1.0.3
multidict                     6.0.2
multipledispatch              0.6.0
munkres                       1.1.4
mypy-extensions               0.4.3
navigator-updater             0.4.0
nbclassic                     0.5.5
nbclient                      0.5.13
nbconvert                     6.5.4
nbformat                      5.7.0
nest-asyncio                  1.5.6
networkx                      2.8.4
nltk                          3.7
notebook                      6.5.4
notebook_shim                 0.2.2
numba                         0.57.0
numexpr                       2.8.4
numpy                         1.24.3
numpydoc                      1.5.0
openpyxl                      3.0.10
packaging                     23.0
pandas                        1.5.3
pandocfilters                 1.5.0
panel                         1.1.0
param                         1.13.0
paramiko                      2.8.1
parsel                        1.6.0
parso                         0.8.3
partd                         1.2.0
pathlib                       1.0.1
pathspec                      0.10.3
```

```
patsy                         0.5.3
pep8                          1.7.1
pexpect                       4.8.0
pickleshare                   0.7.5
Pillow                        9.4.0
pip                           23.1.2
pkce                          1.0.3
pkginfo                       1.9.6
platformdirs                  2.5.2
plotly                        5.9.0
pluggy                        1.0.0
ply                           3.11
pooch                         1.4.0
poyo                          0.5.0
prometheus-client             0.14.1
prompt-toolkit                3.0.36
Protego                       0.1.16
psutil                        5.9.0
ptyprocess                    0.7.0
pure-eval                     0.2.2
py-cpuinfo                    8.0.0
pyarrow                       11.0.0
pyasn1                        0.4.8
pyasn1-modules                0.2.8
pycodestyle                   2.10.0
pycosat                       0.6.4
pycparser                     2.21
pyct                          0.5.0
pycurl                        7.45.2
pydantic                      2.5.3
pydantic_core                 2.14.6
PyDispatcher                  2.0.5
pydocstyle                    6.3.0
pyerfa                        2.0.0
pyflakes                      3.0.1
Pygments                      2.15.1
PyJWT                         2.4.0
pylint                        2.16.2
pylint-venv                   2.3.0
pyls-spyder                   0.4.0
PyNaCl                        1.5.0
pyodbc                        4.0.34
pyOpenSSL                     23.0.0
pyparsing                     3.0.9
PyQt5                         5.15.7
PyQt5-sip                     12.11.0
PyQtWebEngine                 5.15.4
pyrsistent                    0.18.0
PySocks                       1.7.1
pytest                        7.3.1
python-dateutil               2.8.2
python-dotenv                 0.21.0
python-json-logger            2.0.7
python-lsp-black              1.2.1
python-lsp-jsonrpc            1.0.0
python-lsp-server             1.7.2
python-slugify                5.0.2
python-snappy                 0.6.1
pytoolconfig                  1.2.5
pytz                          2022.7
pyviz-comms                   2.3.0
PyWavelets                    1.4.1
pywin32                       305.1
pywin32-ctypes                0.2.0
pywinpty                      2.0.10
PyYAML                        6.0
pyzmq                         24.0.1
QDarkStyle                    3.0.2
qstylizer                     0.2.2
```

```
QtAwesome                     1.2.2
qtconsole                     5.4.2
QtPy                          2.2.0
queuelib                      1.5.0
regex                         2022.7.9
requests                      2.29.0
requests-file                 1.5.1
requests-toolbelt             0.9.1
rfc3339-validator             0.1.4
rfc3986-validator             0.1.1
rope                          1.7.0
Rtree                         1.0.1
ruamel.yaml                   0.17.21
ruamel-yaml-conda             0.17.21
s3fs                          2023.3.0
s3transfer                    0.6.0
sacremoses                    0.0.43
scikit-image                  0.20.0
scikit-learn                  1.2.2
scikit-learn-intelex          20230426.121932
scipy                         1.10.1
Scrapy                        2.8.0
seaborn                       0.12.2
semver                        3.0.2
Send2Trash                    1.8.0
service-identity              18.1.0
setuptools                    67.8.0
sip                           6.6.2
six                           1.16.0
smart-open                    5.2.1
sniffio                       1.2.0
snowballstemmer               2.2.0
sortedcontainers              2.4.0
soupsieve                     2.4
Sphinx                        5.0.2
sphinxcontrib-applehelp       1.0.2
sphinxcontrib-devhelp         1.0.2
sphinxcontrib-htmlhelp        2.0.0
sphinxcontrib-jsmath          1.0.1
sphinxcontrib-qthelp          1.0.3
sphinxcontrib-serializinghtml 1.1.5
spyder                        5.4.3
spyder-kernels                2.4.3
SQLAlchemy                    1.4.39
stack-data                    0.2.0
statsmodels                   0.13.5
sympy                         1.11.1
tables                        3.8.0
tabulate                      0.8.10
TBB                           0.2
tblib                         1.7.0
tenacity                      8.2.2
terminado                     0.17.1
text-unidecode                1.3
textdistance                  4.2.1
threadpoolctl                 2.2.0
three-merge                   0.1.1
tifffile                      2021.7.2
tinycss2                      1.2.1
tldextract                    3.2.0
toml                          0.10.2
tomli                         2.0.1
tomlkit                       0.11.1
toolz                         0.12.0
tornado                       6.2
tqdm                          4.65.0
traitlets                     5.7.1
transformers                  2.1.1
Twisted                       22.10.0
```

```
twisted-iocpsupport           1.0.2
typing_extensions             4.6.3
uc-micro-py                   1.0.1
ujson                         5.4.0
Unidecode                     1.2.0
urllib3                       1.26.16
w3lib                         1.21.0
watchdog                      2.1.6
wcwidth                       0.2.5
webencodings                  0.5.1
websocket-client              0.58.0
Werkzeug                      2.2.3
whatthepatch                  1.0.2
wheel                         0.38.4
widgetsnbextension            4.0.5
win-inet-pton                 1.1.0
wrapt                         1.14.1
xarray                        2022.11.0
xlrd                          2.0.1
xlwings                       0.29.1
xyzservices                   2022.9.0
y-py                          0.5.9
yapf                          0.31.0
yarl                          1.8.1
ypy-websocket                 0.8.2
zict                          2.2.0
zipp                          3.11.0
zope.interface                5.4.0
zstandard                     0.19.0
Note: you may need to restart the kernel to use updated packages.
```

In [5]:

```
# 必要的话，要安装版本大于1.0.0的xlrd包，以让pandas正确读取excel文件
# %pip install xlrd
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting xlrd
  Downloading xlrd-2.0.1-py2.py3-none-any.whl (96 kB)
                                              0.0/96.5 kB ? eta -:--:--
     ------------                             30.7/96.5 kB 1.3 MB/s eta 0:00:01
     ----------------------------           71.7/96.5 kB 653.6 kB/s eta 0:00:01
     -------------------------------------- 96.5/96.5 kB 784.5 kB/s eta 0:00:00
Installing collected packages: xlrd
Successfully installed xlrd-2.0.1
Note: you may need to restart the kernel to use updated packages.
```

In [1]:

```
import numpy as np
import pandas as pd
```

1. 读取数据

In [2]:

```
## 读取购物篮数据
inputfile = 'shopping_cart.xls'
data = pd.read_excel(inputfile, header = None)
print("\n>>> 购物篮原始数据：\n",data)
```

```
>>> 购物篮原始数据：
     0    1     2     3
0  牛奶  啤酒   尿布   NaN
1  面包  黄油   牛奶   NaN
2  牛奶  尿布   饼干   NaN
3  面包  黄油   饼干   NaN
4  啤酒  饼干   尿布   NaN
5  牛奶  尿布   面包    黄油
6  面包  黄油   尿布   NaN
7  啤酒  尿布  NaN   NaN
8  牛奶  尿布   面包    黄油
9  啤酒  饼干  NaN   NaN
```

2. transferToZeroOneMatrix( )

In [3]:

```
## 功能：将原始数据（二维表）转换成对应的0-1矩阵
## 输入：
##     1）data：原始的购物篮数据（二维表，DataFrame）
## 输出：
##     1）购物篮数据对应的0-1矩阵(DataFrame)
def transferToZeroOneMatrix(data):
    ct = lambda x : pd.Series(1, index = x[pd.notnull(x)])
    ### 上述lamda表达式的功能如下：将一个1D数组x转换成对应的全为1的Series
    ### 例如，若x=['面包','啤酒','尿布' ,None]，则转成的Series相当于pd.Series(1, index=[面包',

    a = map(ct, data.values) # 利用map函数，将lamda算子ct作用到data.values的每一行上
    b = list(a) # a是一个map类型的对象，将它转换成list类型，b中的每个元素是一个Series
    zeroOneMat= pd.DataFrame(b).fillna(0)  # 实现矩阵转换，空值用0填充
    del a,b  # 删除中间变量b，节省内存

    return zeroOneMat
```

In [4]:

```
## 关键代码剖析
ct = lambda x : pd.Series(1, index = x[pd.notnull(x)]) # 将一个1D数组x转换成对应的全为1的Serie
x = np.array(['面包','啤酒','尿布',None])   ## 或者：x = pd.Series(['面包','啤酒','尿布',np.Na
print("\n>>>x= \n",x)
s = ct(x)
print("\n>>>s= \n",s)

b = map(ct, data.values) # b是一个map对象
a = list(b) # 将map对象转成列表
print("\n>>>type(a)= \n",type(a))
print("\n>>>type(a[0])= \n",type(a[0]))
print("\n>>>a= \n",a)
```

```
>>>x=
 ['面包' '啤酒' '尿布' None]

>>>s=
 面包    1
啤酒    1
尿布    1
dtype: int64

>>>type(a)=
 <class 'list'>

>>>type(a[0])=
 <class 'pandas.core.series.Series'>

>>>a=
 [牛奶    1
啤酒    1
尿布    1
dtype: int64, 面包    1
黄油    1
牛奶    1
dtype: int64, 牛奶    1
尿布    1
饼干    1
dtype: int64, 面包    1
黄油    1
饼干    1
dtype: int64, 啤酒    1
饼干    1
尿布    1
dtype: int64, 牛奶    1
尿布    1
面包    1
黄油    1
dtype: int64, 面包    1
黄油    1
尿布    1
dtype: int64, 啤酒    1
尿布    1
dtype: int64, 牛奶    1
尿布    1
面包    1
黄油    1
dtype: int64, 啤酒    1
饼干    1
dtype: int64]
```

In [5]:

```
## 函数测试
zeroOneMat = transferToZeroOneMatrix(data)
print("\n>>> 购物篮数据对应的0-1矩阵: \n",zeroOneMat)
```

```
>>> 购物篮数据对应的0-1矩阵:
     牛奶   啤酒   尿布   面包   黄油   饼干
0  1.0  1.0  1.0  0.0  0.0  0.0
1  1.0  0.0  0.0  1.0  1.0  0.0
2  1.0  0.0  1.0  0.0  0.0  1.0
3  0.0  0.0  0.0  1.0  1.0  1.0
4  0.0  1.0  1.0  0.0  0.0  1.0
5  1.0  0.0  1.0  1.0  1.0  0.0
6  0.0  0.0  1.0  1.0  1.0  0.0
7  0.0  1.0  1.0  0.0  0.0  0.0
8  1.0  0.0  1.0  1.0  1.0  0.0
9  0.0  1.0  0.0  0.0  0.0  1.0
```

3. generateF1( )

In [6]:
```
## 功能：生成频繁1-项集及其支持度
## 输入：
##     1）M: 0-1矩阵 (DataFrame)
##     2）minsupp: 支持度阈值（标量）
## 输出：
##     1）频繁1-项集及其支持度，series类型，索引为频繁1-项集，值为相应的支持度
def generateF1(M,minsupp):
    support_series = 1.0*M.sum()/len(M) #支持度计算，M.sum()按列求和，len(M)计算记录个数（行数
    F1 = support_series[support_series >= minsupp] # 过滤，选择支持度满足要求的itemset
    return F1 # 返回频繁1-项集及其支持度
```

In [7]:
```
## 测试
F1 = generateF1(zeroOneMat,0.3)
print("\n>>> 频繁1-项集：\n",F1) # 注意，F1的索引（index）为itemset,值为相应的支持度
```

```
>>> 频繁1-项集：
 牛奶    0.5
啤酒    0.4
尿布    0.7
面包    0.5
黄油    0.5
饼干    0.4
dtype: float64
```

4. generateCk( )

In [8]:
```
## 功能：生成候选的频繁k-项集C_k
## 输入：
##     1）x: 频繁(k-1)-项集（带分割符）
##     2）ms：项之间的分割符，例如，若以 ‘-’ 为分隔符，则 ‘a-b-c’ 代表的项集是{‘a’,’b’,’c’}
## 输出：
##    1）候选的频繁k-项集C_k
def generateCk(x, ms):
    str2list = lambda s:sorted(s.split(ms)) # 以ms为分割符，提取字符串s中的项并进行排序，返回
    x = list(map(str2list, x)) ## 去掉item之间的分割符，提取item并排序
    L = len(x[0]) # x中的每个元素都是长度相同的列表（即，(k-1)-项集）
    Ck = []
    ## 由频繁(k-1)-项集的笛卡尔积生成C_k: F_k-1*F_k-1 -> C_k
    for i in range(len(x)):
        for j in range(i,len(x)):
            if x[i][:L-1] == x[j][:L-1] and x[i][L-1] != x[j][L-1]: ## x[i]和x[j]的最后一项不
                Ck.append(x[i][:L-1]+sorted([x[j][L-1],x[i][L-1]]))
    return Ck # 候选的频繁k-项集C_k
```

In [9]:
```
## 关键代码剖析
ms = '-'
s = '牛奶-啤酒'
ct = lambda s:sorted(s.split(ms)) # 以ms为分割符，提取字符串s中的项并进行排序，返回的结果是一
a = ct(s)
print("\n>>> 原始字符串：\n",s)
print("\n>>> 根据分隔符分隔之后的字符串：\n",a)
```

```
>>> 原始字符串：
 牛奶-啤酒

>>> 根据分隔符分隔之后的字符串：
 ['啤酒', '牛奶']
```

In [10]:
```
## 函数测试
F1_itemset = list(F1.index) # 频繁1-项集
print("\n>>频繁1-项集：\n",F1_itemset)
C2 = generateCk(F1_itemset,'-') # 候选频繁2-项集
print("\n>>候选频繁2-项集：\n",C2)
```

```
>>频繁1-项集：
 ['牛奶', '啤酒', '尿布', '面包', '黄油', '饼干']

>>候选频繁2-项集：
 [['啤酒', '牛奶'], ['尿布', '牛奶'], ['牛奶', '面包'], ['牛奶', '黄油'], ['牛奶', '饼干'],
['啤酒', '尿布'], ['啤酒', '面包'], ['啤酒', '黄油'], ['啤酒', '饼干'], ['尿布', '面包'], ['尿
布', '黄油'], ['尿布', '饼干'], ['面包', '黄油'], ['面包', '饼干'], ['饼干', '黄油']]
```

5. calSupport( )

In [11]:

```
## 功能：计算k-项集的支持度
## 输入：
##    1）M：购物篮数据对应的0-1矩阵
##    2）itemsets：k-项集列表，列表中的每个元素是列表，表示一个k-itemset；如，[['a','b'],['a','
##    3）ms:item之间的分割符
## 输出：
##    1）k-项集的支持度，Series类型，索引为频繁k-项集，值为相应的支持度
def calSupport(M,itemsets,ms):
    tmpItemsets = [ms.join(i) for i in itemsets] # 用分隔符连接k-itemset，例如，['a','b']变成'
    ## print(tmpItemsets) # 打印增加了分隔符的k-itemset
    sf = lambda cols: M[cols].prod(axis=1, numeric_only = True) # 计算矩阵M中cols对应的多个列
    items_zeroOneMatr = pd.DataFrame(list(map(sf,itemsets)), index = tmpItemsets).T # 转置
    ## print(items_zeroOneMatr) # 打印k-itemset的0-1矩阵
    support_series= 1.0*items_zeroOneMatr[tmpItemsets].sum()/len(M) #计算支持度
    return support_series # 返回k-itemset的支持度 (series类型)
```

In [12]:

```
## 关键代码剖析
M = zeroOneMat
cols = ['牛奶','啤酒','尿布']
sf = lambda cols: M[cols].prod(axis=1, numeric_only = True) # 计算矩阵M中cols对应的多个列向量
print("\n>>购物篮0-1矩阵：\n",M)
print("\n>>{0}这几列的点积为：\n".format(cols),sf(cols))
```

```
>>购物篮0-1矩阵：
      牛奶   啤酒   尿布   面包   黄油   饼干
0   1.0  1.0  1.0  0.0  0.0  0.0
1   1.0  0.0  0.0  1.0  1.0  0.0
2   1.0  0.0  1.0  0.0  0.0  1.0
3   0.0  0.0  0.0  1.0  1.0  1.0
4   0.0  1.0  1.0  0.0  0.0  1.0
5   1.0  0.0  1.0  1.0  1.0  0.0
6   0.0  0.0  1.0  1.0  1.0  0.0
7   0.0  1.0  1.0  0.0  0.0  0.0
8   1.0  0.0  1.0  1.0  1.0  0.0
9   0.0  1.0  0.0  0.0  0.0  1.0

>>['牛奶', '啤酒', '尿布']这几列的点积为：
 0     1.0
1     0.0
2     0.0
3     0.0
4     0.0
5     0.0
6     0.0
7     0.0
8     0.0
9     0.0
dtype: float64
```

In [13]:

```
## 函数测试
ss = calSupport(zeroOneMat,C2,ms)
print("\n>>2-项集：\n",C2)
print("\n>>2-项集对应的支持度：\n",ss)
```

```
>>2-项集:
 [['啤酒', '牛奶'], ['尿布', '牛奶'], ['牛奶', '面包'], ['牛奶', '黄油'], ['牛奶', '饼干'],
['啤酒', '尿布'], ['啤酒', '面包'], ['啤酒', '黄油'], ['啤酒', '饼干'], ['尿布', '面包'], ['尿
布', '黄油'], ['尿布', '饼干'], ['面包', '黄油'], ['面包', '饼干'], ['饼干', '黄油']]

>>2-项集对应的支持度:
 啤酒-牛奶    0.1
尿布-牛奶    0.4
牛奶-面包    0.3
牛奶-黄油    0.3
牛奶-饼干    0.1
啤酒-尿布    0.3
啤酒-面包    0.0
啤酒-黄油    0.0
啤酒-饼干    0.2
尿布-面包    0.3
尿布-黄油    0.3
尿布-饼干    0.2
面包-黄油    0.5
面包-饼干    0.1
饼干-黄油    0.1
dtype: float64
```

6. generateFk( )

In [14]:

```
## 功能:生成频繁k-项集F_k及其支持度
## 输入:
##     1) itemsets_support: k-itemsets及对应的支持度(Series类型,索引为带分隔符的频繁k-itemset,值
##     2) minsupp: 支持度阈值
## 输出:
##    1) 频繁k-项集及其支持度, Series类型, 索引为带分隔符的频繁k-itemset, 值为相应的支持度
def generateFk(itemsets_support,minsupp):
    Fk = itemsets_support[itemsets_support >= minsupp] # 过滤, 选择支持度满足要求的itemset
    return Fk # 返回频繁k-项集及其支持度
```

7. generateAllFIS( )

In [15]:

```
## 功能:生成所有的频繁项集
## 输入:
##     1) data: 原始的购物篮数据(二维表, DataFrame)
##     2) minsupp: 支持度阈值
##     3) ms:item之间的分割符
## 输出:
##    1) 频繁项集,list类型, 第k个元素(仍然是list)表示频繁k-项集
##    2) 带支持度的频繁项集, list类型, 第k个元素(Series)表示频繁k-项集及其支持度(索引为带分隔
def generateAllFIS(data,minsupp,ms):
    M = transferToZeroOneMatrix(data) # 将原始数据集转成相应的0-1矩阵

    fItemsets = [] # 频繁项集列表, 第1个元素表示频繁1-项集, 第2个元素表示频繁2-项集, ...
    fItemsets_support = [] # 频繁项集及其支持度的列表, 第1个元素表示频繁1-项集及其支持度的Serie

    ## 生成频繁1-项集
    F1 = generateF1(M,minsupp) # 频繁1-项集及其支持度
    fItemsets.append(list(F1.index))
    fItemsets_support.append(F1)
    ## 生成其它所有的频繁项集
    for k in range(M.shape[1]): # k=0,...,num_of_items-1
        Ck = generateCk(fItemsets[k],ms)
        if not Ck: # Ck is emtpy, then exit the loop
            break
        Sk = calSupport(M,Ck,ms) # compute the support value
        Fk = generateFk(Sk,minsupp) # freqent k-itemsets
        fItemsets.append(list(Fk.index))
        fItemsets_support.append(Fk)
    return fItemsets,fItemsets_support
```

In [16]:

```
## 函数测试
minsupp = 0.3
ms = '-'
fItemsets,fItemsets_support = generateAllFIS(data,minsupp,ms)
print("\n>>> 购物篮数据：\n",data)
print("\n>>> 所有的频繁项集：\n",fItemsets)
print("\n>>> 所有的频繁项集及其支持度：\n",fItemsets_support)
```

```
>>> 购物篮数据：
     0    1     2     3
0  牛奶  啤酒   尿布  NaN
1  面包  黄油   牛奶  NaN
2  牛奶  尿布   饼干  NaN
3  面包  黄油   饼干  NaN
4  啤酒  饼干   尿布  NaN
5  牛奶  尿布   面包   黄油
6  面包  黄油   尿布  NaN
7  啤酒  尿布  NaN  NaN
8  牛奶  尿布   面包   黄油
9  啤酒  饼干  NaN  NaN

>>> 所有的频繁项集：
 [['牛奶', '啤酒', '尿布', '面包', '黄油', '饼干'], ['尿布-牛奶', '牛奶-面包', '牛奶-黄油', '啤酒-尿布', '尿布-面包', '尿布-黄油', '面包-黄油'], ['牛奶-面包-黄油', '尿布-面包-黄油']]

>>> 所有的频繁项集及其支持度：
 [牛奶    0.5
啤酒    0.4
尿布    0.7
面包    0.5
黄油    0.5
饼干    0.4
dtype: float64, 尿布-牛奶    0.4
牛奶-面包    0.3
牛奶-黄油    0.3
啤酒-尿布    0.3
尿布-面包    0.3
尿布-黄油    0.3
面包-黄油    0.5
dtype: float64, 牛奶-面包-黄油    0.3
尿布-面包-黄油    0.3
dtype: float64]
```

8. generateRule( )

In [17]:

```
## 功能：生成某个itemset对应的所有关联规则 L->R
## 输入：
##    1）items：k-项集，集合类型，形如，{'a','b','c'}
##    2）H：用以存储关联规则后件(右部R)的列表，列表中每个元素是字典；H同时也是输出参数
##    3）m：规则的层次，即后件包含元素的个数；例如，'a-b'->'c'、'a'->'b-c'分别是属于第1层和第2层
##    4）fItemsets_support：已经计算得到的所有频繁项集及其支持度
##    5）minconf：置信度阈值
##    6）ms：item之间的分割符
## 输出：
##    1）H：用以存储关联规则后件(右部R)的列表，列表中每个元素是字典，第1个元素是第1层的所有后件
def generateRule(items,H,m,fItemsets_support,minconf,ms):
     if 1 <= m < len(items)-1: # 从第2层(m=1)开始递归生成各层的关联规则
        itemList = list(items) # 将set转成list，如，{'a','b','c'}转成['a','b','c']
        itemList.sort() # 排序
        nextH = {} # 字典
        for post in H[m-1].keys():
            pre = items - set(post) # 规则的前件（item的集合）
            for i in pre: # 尝试将前件pre中的每个item移动到后件post
                tmpS = set([])
                tmpS.add(i)

                L = list(pre - tmpS) # 当前规则的前件
                L.sort()
                L_index = ms.join(L) # 用分割符将item联结起来，以便在fItemsets_support中查找相

                R = list(post) # 当前规则的后件
                R.append(i)
                R.sort()
                #计算置信度
                confidence = fItemsets_support[len(items)-1][ms.join(itemList)] / fItemsets_sup

                if confidence >= minconf: # 找到一条符合要求的规则，存储相应的规则后件
                    nextH[tuple(R)] = confidence # 存储规则的后件（以tuple的形式放在字典的key中
        if nextH:
            H.append(nextH)

        generateRule(items,H,m+1,fItemsets_support,minconf,ms) # 递归调用
```

9. generateRuleWrapper( )

In [18]:

```
## 功能：给定一个频繁k-项集，生成相应的所有关联规则
## 输入参数：
##    1）fItems：带分隔符的频繁k-项集，形如，'a-b-c'(假设以'-'为分隔符)
##    2）fItemsets_support：已经计算得到的所有频繁项集及其支持度
##    5）minconf：置信度阈值
##    6）ms：item之间的分割符
## 输出参数：
##    1）H：用以存储关联规则后件(右部R)的列表，列表中每个元素是字典，第1个元素是第1层的所有后件

def generateRuleWrapper(fItems,fItemsets_support,minconf,ms):
    items = set(fItems.split(ms)) # 将'a-b-c'转成{'a','b','c'}（假设以'-'为分隔符）
    # print(items)

    ## 首先，生成规则后件只有一个item的（即第1层的）关联规则，形如，'a-b'->'c'
    H = []
    nextH = {}
    for i in items:
        tmpS = set([])
        tmpS.add(i)

        L = list(items - tmpS) # 当前规则的前件
        L.sort()
        L_index = ms.join(L) # 用分割符将item联结起来，以便在fItemsets_support中查找相应的支持

        R = [i] # 当前规则的后件（只包含一个item）

        #print("L={0},R={1}".format(L,R))

        confidence = fItemsets_support[len(items)-1][fItems] / fItemsets_support[len(L)-1][L_i
        if confidence >= minconf: # 找到一条符合要求的规则，存储相应的规则后件
            nextH[tuple(R)] = confidence # 存储规则的后件（以tuple的形式放在字典的key中）及规则

    if nextH:
        H.append(nextH) # 生成规则的第1层后件（即后件只含1个item）


    ## 其次，调用genRule()生成其它层的关联规则
    m=1
    generateRule(items,H,m,fItemsets_support,minconf,ms) #从第2层开始，继续生成其它各层的后件

    return H
```

10. generateAllRules( )

In [19]:

```
## 功能：生成所有的关联规则
## 输入参数：
##    1）fItemsets：所有的频繁项集
##    2）fItemsets_support：已经计算得到的所有频繁项集及其支持度
##    3）minconf：置信度阈值
##    4）ms：item之间的分割符
## 输出：
##    1）关联规则的列表，列表中每个元素是字典，字典有两个key，分别是'items'和'post'；
##       'items'对应的是某个频繁项集，而'post'对应的所有符合要求的后件（一个后件对应一条关联规则
##       例如：items:'a-b-c'
##             post:[{(a,):0.6},{(c,):0.65},{(b,c):0.7}]   （注意，post列表中每个元素是一个字典）
def generateAllRules(fItemsets,fItemsets_support,minconf,ms):
    rules=[]
    for k in range(1,len(fItemsets)): # 从频繁2-项集开始（1-项集无法产生关联规则，没有后件），
        for fItems in fItemsets[k]: # 遍历Fk中的所有k-itemset
            H = generateRuleWrapper(fItems,fItemsets_support,minconf,ms)
            rules.append({'items':fItems,'post':H}) # 列表中的元素

    return rules
```

In [20]:

```
## 函数测试
minconf = 0.5  # 最小置信度
ms = '-'
rules = generateAllRules(fItemsets,fItemsets_support,minconf,ms)
print(rules)
```

```
[{'items': '尿布-牛奶', 'post': [{('尿布',): 0.8, ('牛奶',): 0.5714285714285715}]}, {'items':
'牛奶-面包', 'post': [{('面包',): 0.6, ('牛奶',): 0.6}]}, {'items': '牛奶-黄油', 'post': [{('黄
油',): 0.6, ('牛奶',): 0.6}]}, {'items': '啤酒-尿布', 'post': [{('尿布',): 0.7499999999999999
9}]}, {'items': '尿布-面包', 'post': [{('尿布',): 0.6}]}, {'items': '尿布-黄油', 'post': [{('尿
布',): 0.6}]}, {'items': '面包-黄油', 'post': [{('黄油',): 1.0, ('面包',): 1.0}]}, {'items':
'牛奶-面包-黄油', 'post': [{('黄油',): 1.0, ('面包',): 1.0, ('牛奶',): 0.6}, {('面包', '黄油'):
0.6, ('牛奶', '黄油'): 0.6, ('牛奶', '面包'): 0.6}]}, {'items': '尿布-面包-黄油', 'post':
[{('黄油',): 1.0, ('面包',): 1.0, ('尿布',): 0.6}, {('尿布', '黄油'): 0.6, ('尿布', '面包'): 0.
6}]}]
```

In [ ]: