数据挖掘作业2

姓名：胡宗晖 学号：3220220922

网页浏览行为关联规则挖掘

In [1]:

```
#导入必要的包
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
import warnings
warnings.filterwarnings('ignore')
```

1.查看数据集并进行预处理

In [2]:

```
#读入文件
data=[]
with open('D:/study/data/anonymous-msweb.data', 'r') as f_input:
    for line in f_input:
        data.append(list(line.strip().split(',')))
data
```

Out[2]:

```
[['I', '4', '"www.microsoft.com"', '"created by getlog.pl"'],
 ['T', '1', '"VRoot"', '0', '0', '"VRoot"'],
 ['N', '0', '"0"'],
 ['N', '1', '"1"'],
 ['T', '2', '"Hide1"', '0', '0', '"Hide"'],
 ['N', '0', '"0"'],
 ['N', '1', '"1"'],
 ['A', '1287', '1', '"International AutoRoute"', '"/autoroute"'],
 ['A', '1288', '1', '"library"', '"/library"'],
 ['A', '1289', '1', '"Master Chef Product Information"', '"/masterchef"'],
 ['A', '1297', '1', '"Central America"', '"/centroam"'],
 ['A', '1215', '1', '"For Developers Only Info"', '"/developer"'],
 ['A', '1279', '1', '"Multimedia Golf"', '"/msgolf"'],
 ['A', '1239', '1', '"Microsoft Consulting"', '"/msconsult"'],
 ['A', '1282', '1', '"home"', '"/home"'],
 ['A', '1251', '1', '"Reference Support"', '"/referencesupport"'],
 ['A', '1121', '1', '"Microsoft Magazine"', '"/magazine"'],
 ['A', '1083', '1', '"MS Access Support"', '"/msaccesssupport"'],
```

In [3]:

```
#提取网页id与title之间的关系
vroot = pd.DataFrame(columns=['id','title'])

for i in range(len(data)):
    if(data[i][0]=='A'):
        df2 = pd.DataFrame({'id': data[i][1],'title':data[i][3]},index=[0])
        vroot = pd.concat([vroot,df2])
vroot = vroot.set_index(['id'])
vroot
```

Out[3]:

id	title
1287	"International AutoRoute"
1288	"library"
1289	"Master Chef Product Information"
1297	"Central America"
1215	"For Developers Only Info"
...	...
1219	"Corporate Advertising Content"
1030	"Windows NT Server"
1182	"Fortran"
1100	"MS in Education"
1210	"SNA Support"

294 rows × 1 columns

In [4]:

```
#建立数据集1，其中每一条记录由访问者id与访问网站id组成
df = pd.DataFrame(columns=['user','vroot'])
usr=''
count=0
for i in range(len(data)):
    if(data[i][0]=='C'):
        usr = data[i][2]
    if(data[i][0]=='V'):
        df2 = pd.DataFrame({'user': usr,'vroot':data[i][1]},index=[count])
        count = count+1
        df = pd.concat([df,df2])
df
```

Out[4]:

	user	vroot
0	10001	1000
1	10001	1001
2	10001	1002
3	10002	1001
4	10002	1003
...	...	...
98649	42709	1003
98650	42710	1035
98651	42710	1001
98652	42710	1018
98653	42711	1008

98654 rows × 2 columns

In [5]:

```
#建立数据集2，其中每一条记录对应一个访问者id，包含该访问者访问的所有网站id
id_set= []
vroot_set = []
vr = []

dataset = {}
for i in range(len(data)):
    if(data[i][0]=='C'):
        vroot_set.append(vr)
        id_set.append(data[i][2])
        vr = []
    if(data[i][0]=='V'):
        vr.append(data[i][1])
vroot_set.append(vr)
vroot_set.pop(0)

dataset['id']=id_set
dataset['vroot']=vroot_set
data1 = pd.DataFrame(dataset)
#data1 = data1.set_index(['id'])
data1
```

Out[5]:

	id	vroot
0	10001	[1000, 1001, 1002]
1	10002	[1001, 1003]
2	10003	[1001, 1003, 1004]
3	10004	[1005]
4	10005	[1006]
...	...	...
32706	42707	[1008, 1030, 1009, 1058, 1004, 1018]
32707	42708	[1008, 1027, 1123, 1038, 1026, 1041]
32708	42709	[1001, 1003]
32709	42710	[1035, 1001, 1018]
32710	42711	[1008]

32711 rows × 2 columns

2.数据探索性分析

In [6]:

```
#查看页面访问量分布
pd.value_counts(df['vroot'])
```

Out[6]:

```
1008    10836
1034     9383
1004     8463
1018     5330
1017     5108
        ...
1196        1
1199        1
1233        1
1128        1
1284        1
Name: vroot, Length: 285, dtype: int64
```

In [7]:

```
#最常被访问的页面，id为1008
vroot.loc['1008']['title']
```

Out[7]:

```
'"Free Downloads"'
```

3.关联规则挖掘

使用Apriori算法

In [8]:

```
#对数据集进行处理，转换成one-hot编码
data_id = data1.drop('vroot',1)
data2 = data1['vroot'].str.join(',')
data2 = data2.str.get_dummies(',')
new_data = data_id.join(data2)
new_data
```

Out[8]:

	id	1000	1001	1002	1003	1004	1005	1006	1007	1008	...	1276	1277	1278
0	10001	1	1	1	0	0	0	0	0	0	...	0	0	0
1	10002	0	1	0	1	0	0	0	0	0	...	0	0	0
2	10003	0	1	0	1	1	0	0	0	0	...	0	0	0
3	10004	0	0	0	0	0	1	0	0	0	...	0	0	0
4	10005	0	0	0	0	0	0	1	0	0	...	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
32706	42707	0	0	0	0	1	0	0	0	1	...	0	0	0
32707	42708	0	0	0	0	0	0	0	0	1	...	0	0	0
32708	42709	0	1	0	1	0	0	0	0	0	...	0	0	0
32709	42710	0	1	0	0	0	0	0	0	0	...	0	0	0
32710	42711	0	0	0	0	0	0	0	0	1	...	0	0	0

32711 rows × 286 columns

In [22]:

```
#计算频繁项集，修改最小支持度为0.05
frequent_itemsets = apriori(new_data.drop('id',1),min_support=0.05,use_colnames=True)
frequent_itemsets
```

Out[22]:

	support	itemsets
0	0.136070	(1001)
1	0.090734	(1003)
2	0.258720	(1004)
3	0.331265	(1008)
4	0.141481	(1009)
5	0.156155	(1017)
6	0.162942	(1018)
7	0.064902	(1025)
8	0.098438	(1026)
9	0.286845	(1034)
10	0.054752	(1035)
11	0.055211	(1003, 1001)
12	0.059430	(1018, 1001)
13	0.060438	(1004, 1008)
14	0.053285	(1034, 1004)
15	0.077925	(1009, 1008)
16	0.061233	(1017, 1008)
17	0.073064	(1018, 1008)
18	0.160802	(1034, 1008)

In [23]:

```
#计算关联规则
association_rules(frequent_itemsets,metric='lift')
```

Out[23]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leve
0	(1003)	(1001)	0.090734	0.136070	0.055211	0.608491	4.471879	0.04
1	(1001)	(1003)	0.136070	0.090734	0.055211	0.405752	4.471879	0.04
2	(1018)	(1001)	0.162942	0.136070	0.059430	0.364728	2.680435	0.03
3	(1001)	(1018)	0.136070	0.162942	0.059430	0.436756	2.680435	0.03
4	(1009)	(1008)	0.141481	0.331265	0.077925	0.550778	1.662652	0.03
5	(1008)	(1009)	0.331265	0.141481	0.077925	0.235234	1.662652	0.03
6	(1017)	(1008)	0.156155	0.331265	0.061233	0.392130	1.183736	0.00
7	(1008)	(1017)	0.331265	0.156155	0.061233	0.184847	1.183736	0.00
8	(1018)	(1008)	0.162942	0.331265	0.073064	0.448405	1.353616	0.01
9	(1008)	(1018)	0.331265	0.162942	0.073064	0.220561	1.353616	0.01
10	(1034)	(1008)	0.286845	0.331265	0.160802	0.560588	1.692267	0.06
11	(1008)	(1034)	0.331265	0.286845	0.160802	0.485419	1.692267	0.06

4.结果评估

由上述关联规则表可以看出强关联规则有：

（1）[1001,1003]支持度为0.055211，提升度为4.471879，置信度（1003-1001）为0.608491，置信度（1001-1003）为0.405752

（2）[1001,1018]支持度为0.059430，提升度为2.680435，置信度（1018-1001）为0.364728，置信度（1001-1018）为0.436756

（3）[1008,1009]支持度为0.077925，提升度为1.662652，置信度（1009-1008）为0.550778，置信度（1008-1009）为0.235234

（4）[1008,1017]支持度为0.061233，提升度为1.183736，置信度（1017-1008）为0.392130，置信度（1008-1017）为0.184847

（5）[1008,1018]支持度为0.073064，提升度为1.353616，置信度（1018-1008）为0.448405，置信度（1008-1018）为0.220561

（6）[1008,1034]支持度为0.160802，提升度为1.692267，置信度（1034-1008）为0.560588，置信度（1008-1034）为0.485419

5.结果分析与应用

根据以上得到的关联规则，为提升用户体验，网站应对导航结构进行如下优化：

(1) 在网站1001中，提供网站1003，1018的导航；

(2) 在网站1008中，提供网站1009，1017，1018，1034的导航。