关联规则挖掘——使用 UCI “急性炎症”数据集

郑越 2120151072

一、环境

mac 系统下，采用 R 语言编程

二、对数据集进行处理

```
#1、读数据
data<-list()
x<-read.transactions("~/Desktop/Analysis.csv",format="basket",sep = "")#转换数据
# for (n in 1:length(x)) {data[n]<-strsplit(x[n],",")}
#
x
summary(x)
trans<-as(x,"transactions")
```

使用 transactions 函数对数据集进行处理，转变为适用于关联规则挖掘的形式.转换成稀疏矩阵的形式。下图 2 和 3 中展示了部分转换过的数据：

```
> summary(x)
transactions as itemMatrix in sparse format with
 120 rows (elements/itemsets/transactions) and
 53 columns (items) and a density of 0.03773585

most frequent items:
,no,yes,yes,no,yes,no,yes ,no,no,yes,yes,yes,yes,no    ,no,yes,no,no,no,no,no     ,no,no,no,no,no,no,no   ,no,no,yes,no,no,yes,no
                      21                          20                         20                         10                         10
                  (Other)
                      159

element (itemset/transaction) length distribution:
sizes
  2
120

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      2       2       2       2       2       2

includes extended item information - examples:
                  labels
1    ,no,no,no,no,no,no,no
2  ,no,no,yes,no,no,yes,no
3 ,no,no,yes,yes,no,yes,no
> trans<-as(x,"transactions")
> trans
transactions in sparse format with
 120 transactions (rows) and
 53 items (columns)
```

图 1

```
> inspect(x)
    items
1   {,no,yes,no,no,no,no,no,35,5}
2   {,no,no,yes,yes,yes,yes,no,35,9}
3   {,no,yes,no,no,no,no,no,35,9}
4   {,no,no,yes,yes,yes,yes,no,36,0}
5   {,no,yes,no,no,no,no,no,36,0}
6   {,no,yes,no,no,no,no,no,36,0}
7   {,no,no,yes,yes,yes,yes,no,36,2}
8   {,no,yes,no,no,no,no,no,36,2}
9   {,no,no,yes,yes,yes,yes,no,36,3}
10  {,no,no,yes,yes,yes,yes,no,36,6}
11  {,no,no,yes,yes,yes,yes,no,36,6}
12  {,no,yes,no,no,no,no,no,36,6}
13  {,no,yes,no,no,no,no,no,36,6}
14  {,no,no,yes,yes,yes,yes,no,36,7}
15  {,no,yes,no,no,no,no,no,36,7}
16  {,no,yes,no,no,no,no,no,36,7}
17  {,no,no,yes,yes,yes,yes,no,36,8}
18  {,no,no,yes,yes,yes,yes,no,36,8}
19  {,no,no,yes,yes,yes,yes,no,36,9}
20  {,no,yes,no,no,no,no,no,36,9}
21  {,no,no,yes,yes,no,yes,no,37,0}
22  {,no,no,yes,yes,no,yes,no,37,0}
23  {,no,yes,no,no,no,no,no,37,0}
24  {,no,no,yes,yes,yes,yes,no,37,0}
25  {,no,no,yes,yes,yes,yes,no,37,0}
26  {,no,no,yes,yes,yes,yes,no,37,0}
27  {,no,no,yes,yes,yes,yes,no,37,0}
28  {,no,no,yes,no,no,yes,no,37,0}
29  {,no,yes,no,no,no,no,no,37,1}
30  {,no,no,yes,yes,yes,yes,no,37,1}
31  {,no,no,yes,no,no,yes,no,37,1}
32  {,no,no,yes,yes,no,yes,no,37,2}
33  {,no,yes,no,no,no,no,no,37,2}
```

图 2

```
34  {,no,no,yes,no,no,yes,no,37,2}
35  {,no,yes,no,no,no,no,no,37,3}
36  {,no,no,yes,yes,yes,yes,no,37,3}
37  {,no,no,yes,no,no,yes,no,37,3}
38  {,no,yes,no,no,no,no,no,37,4}
39  {,no,no,yes,no,no,yes,no,37,4}
40  {,no,no,yes,yes,no,yes,no,37,5}
41  {,no,yes,no,no,no,no,no,37,5}
42  {,no,yes,no,no,no,no,no,37,5}
43  {,no,no,yes,yes,yes,yes,no,37,5}
44  {,no,no,yes,no,no,yes,no,37,5}
45  {,no,no,yes,no,no,yes,no,37,5}
46  {,no,no,yes,yes,no,yes,no,37,6}
47  {,no,no,yes,yes,no,yes,no,37,6}
48  {,no,no,yes,yes,yes,yes,no,37,6}
49  {,no,no,yes,yes,no,yes,no,37,7}
50  {,no,no,yes,yes,no,yes,no,37,7}
51  {,no,yes,no,no,no,no,no,37,7}
52  {,no,no,yes,no,no,yes,no,37,7}
53  {,no,yes,no,no,no,no,no,37,8}
54  {,no,no,yes,yes,yes,yes,no,37,8}
55  {,no,no,yes,no,no,yes,no,37,8}
56  {,no,no,yes,yes,no,yes,no,37,9}
57  {,no,no,yes,yes,no,yes,no,37,9}
58  {,no,yes,no,no,no,no,no,37,9}
59  {,no,no,yes,yes,yes,yes,no,37,9}
60  {,no,no,yes,no,no,yes,no,37,9}
61  {,no,yes,yes,no,yes,no,yes,38,0}
62  {,no,yes,yes,no,yes,no,yes,38,0}
63  {,no,yes,yes,no,yes,no,yes,38,1}
64  {,no,yes,yes,no,yes,no,yes,38,3}
65  {,no,yes,yes,no,yes,no,yes,38,5}
66  {,no,yes,yes,no,yes,no,yes,38,7}
67  {,no,yes,yes,no,yes,no,yes,38,9}
```

图 3

三、频繁项集

将支持度 support 设置为 0.01，求出所有的频繁项集

```
frequentsets<- eclat(trans,parameter=list(support=0.01,maxlen=10,minlen=2))#频繁项集
inspect(frequentsets)#所有频繁项集
```

```
> inspect(frequentsets)#所有频繁项集
   items                                    support
1  {,no,yes,yes,no,yes,no,yes,38,0}         0.01666667
2  {,no,no,yes,yes,yes,yes,no,36,8}         0.01666667
3  {,no,no,yes,yes,no,yes,no,37,6}          0.01666667
4  {,no,yes,no,no,no,no,no,36,7}            0.01666667
5  {,yes,yes,yes,yes,no,yes,yes,40,9}       0.01666667
6  {,no,yes,no,no,no,no,no,36,0}            0.01666667
7  {,no,yes,no,no,no,no,no,36,6}            0.01666667
8  {,no,no,yes,yes,yes,yes,no,36,6}         0.01666667
9  {,no,yes,yes,no,yes,no,yes,41,5}         0.01666667
10 {,no,no,yes,yes,no,yes,no,37,7}          0.01666667
11 {,no,no,yes,yes,no,yes,no,37,9}          0.01666667
12 {,yes,yes,yes,yes,no,yes,yes,40,4}       0.01666667
13 {,no,yes,no,no,no,no,no,37,5}            0.01666667
14 {,no,no,yes,no,no,yes,no,37,5}           0.01666667
15 {,no,no,yes,yes,yes,yes,no,37,0}         0.03333333
16 {,no,no,yes,yes,no,yes,no,37,0}          0.01666667
17 {,no,no,no,no,no,no,no,40,0}             0.01666667
18 {,yes,yes,no,yes,no,no,yes,40,0}         0.01666667
19 {,yes,yes,yes,yes,yes,yes,yes,40,0}      0.01666667
```

图 4

四、关联规则

将支持度设置为 0.01，置信度设置为 0.4，求出所有的关联规则

```
rules <- apriori(trans,parameter=list(support=0.01,confidence=0.4,minlen=2))#规则
inspect(sort(rules,by="support")[1:6])#按支持度查看前6条规则
inspect(sort(rules,by="confidence")[1:6])#按置信度查看前6条规则
summary(rules)
inspect(rules)#所有规则
sub.rules2=subset(rules, subset = rhs %pin% "2" &lift > 10)
```

图 5

结果如下：

```
> inspect(sort(rules,by="support")[1:6])#按支持度查看前6条规则
   lhs       rhs                             support    confidence lift
13 {37,0} => {,no,no,yes,yes,yes,yes,no}    0.03333333 0.5000000  3.000000
1  {36,8} => {,no,no,yes,yes,yes,yes,no}    0.01666667 1.0000000  6.000000
2  {38,0} => {,no,yes,yes,no,yes,no,yes}    0.01666667 1.0000000  5.714286
3  {36,0} => {,no,yes,no,no,no,no,no}       0.01666667 0.6666667  4.000000
4  {40,9} => {,yes,yes,yes,yes,no,yes,yes} 0.01666667 0.6666667  8.000000
5  {36,7} => {,no,yes,no,no,no,no,no}       0.01666667 0.6666667  4.000000
> inspect(sort(rules,by="confidence")[1:6])#按置信度查看前6条规则
  lhs       rhs                             support    confidence lift
1 {36,8} => {,no,no,yes,yes,yes,yes,no}    0.01666667 1.0000000  6.000000
2 {38,0} => {,no,yes,yes,no,yes,no,yes}    0.01666667 1.0000000  5.714286
3 {36,0} => {,no,yes,no,no,no,no,no}       0.01666667 0.6666667  4.000000
4 {40,9} => {,yes,yes,yes,yes,no,yes,yes} 0.01666667 0.6666667  8.000000
5 {36,7} => {,no,yes,no,no,no,no,no}       0.01666667 0.6666667  4.000000
6 {37,6} => {,no,no,yes,yes,no,yes,no}     0.01666667 0.6666667  8.000000
```

图 6

```
> summary(rules)
set of 13 rules

rule length distribution (lhs + rhs):sizes
 2
13

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      2       2       2       2       2       2

summary of quality measures:
    support           confidence          lift
 Min.   :0.01667   Min.   :0.4000   Min.   :2.857
 1st Qu.:0.01667   1st Qu.:0.5000   1st Qu.:3.000
 Median :0.01667   Median :0.5000   Median :4.800
 Mean   :0.01795   Mean   :0.6128   Mean   :4.859
 3rd Qu.:0.01667   3rd Qu.:0.6667   3rd Qu.:6.000
 Max.   :0.03333   Max.   :1.0000   Max.   :8.000

mining info:
  data ntransactions support confidence
```

图 7

```
> inspect(rules)#所有规则
   lhs       rhs                                 support    confidence lift
1  {36,8} => {,no,no,yes,yes,yes,yes,no}    0.01666667 1.0000000  6.000000
2  {38,0} => {,no,yes,yes,no,yes,no,yes}    0.01666667 1.0000000  5.714286
3  {36,0} => {,no,yes,no,no,no,no,no}       0.01666667 0.6666667  4.000000
4  {40,9} => {,yes,yes,yes,yes,no,yes,yes} 0.01666667 0.6666667  8.000000
5  {36,7} => {,no,yes,no,no,no,no,no}       0.01666667 0.6666667  4.000000
6  {37,6} => {,no,no,yes,yes,no,yes,no}     0.01666667 0.6666667  8.000000
7  {37,7} => {,no,no,yes,yes,no,yes,no}     0.01666667 0.5000000  6.000000
8  {41,5} => {,no,yes,yes,no,yes,no,yes}    0.01666667 0.5000000  2.857143
9  {36,6} => {,no,yes,no,no,no,no,no}       0.01666667 0.5000000  3.000000
10 {36,6} => {,no,no,yes,yes,yes,yes,no}    0.01666667 0.5000000  3.000000
11 {40,4} => {,yes,yes,yes,yes,no,yes,yes} 0.01666667 0.4000000  4.800000
12 {37,9} => {,no,no,yes,yes,no,yes,no}     0.01666667 0.4000000  4.800000
13 {37,0} => {,no,no,yes,yes,yes,yes,no}    0.03333333 0.5000000  3.000000
```

图 8

五、去除冗余数据

```
#删除冗余规则
subset.matrix<-is.subset(rules,rules)
subset.matrix[lower.tri(subset.matrix,diag = T)]<-NA
redundant<-colSums(subset.matrix,na.rm = T)>=1
which(redundant)
rules.pruned<-rules[!redundant]
inspect(rules.pruned)
```

图 9

六、评价规则

采用 lift 评价指标

```
#根据lift排序
sorted_lift<-sort(rules,by='lift')
inspect(sorted_lift)
```

图 10

```
> inspect(sorted_lift)
   lhs       rhs                                 support    confidence lift
4  {40,9} => {,yes,yes,yes,yes,no,yes,yes} 0.01666667 0.6666667  8.000000
6  {37,6} => {,no,no,yes,yes,no,yes,no}     0.01666667 0.6666667  8.000000
1  {36,8} => {,no,no,yes,yes,yes,yes,no}    0.01666667 1.0000000  6.000000
7  {37,7} => {,no,no,yes,yes,no,yes,no}     0.01666667 0.5000000  6.000000
2  {38,0} => {,no,yes,yes,no,yes,no,yes}    0.01666667 1.0000000  5.714286
11 {40,4} => {,yes,yes,yes,yes,no,yes,yes} 0.01666667 0.4000000  4.800000
12 {37,9} => {,no,no,yes,yes,no,yes,no}     0.01666667 0.4000000  4.800000
3  {36,0} => {,no,yes,no,no,no,no,no}       0.01666667 0.6666667  4.000000
5  {36,7} => {,no,yes,no,no,no,no,no}       0.01666667 0.6666667  4.000000
9  {36,6} => {,no,yes,no,no,no,no,no}       0.01666667 0.5000000  3.000000
10 {36,6} => {,no,no,yes,yes,yes,yes,no}    0.01666667 0.5000000  3.000000
13 {37,0} => {,no,no,yes,yes,yes,yes,no}    0.03333333 0.5000000  3.000000
8  {41,5} => {,no,yes,yes,no,yes,no,yes}    0.01666667 0.5000000  2.857143
```

图 11

七、可视化

安装 arulesViz 包，用里面的可视化工具将结果展示出来，代码如下：

```
#可视化
install.packages(pkgs="arulesViz")
library(arulesViz)
plot(rules)
plot(rules,method="graph",control=list(type="items"))
plot(rules,method="paracoord",control=list(reorder=TRUE))
```

图 12

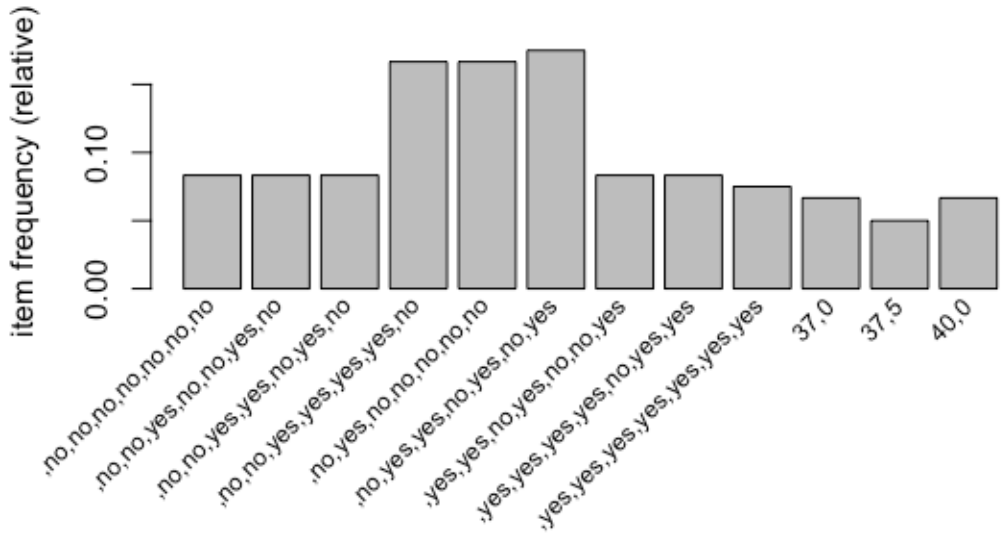

图 13　柱状图

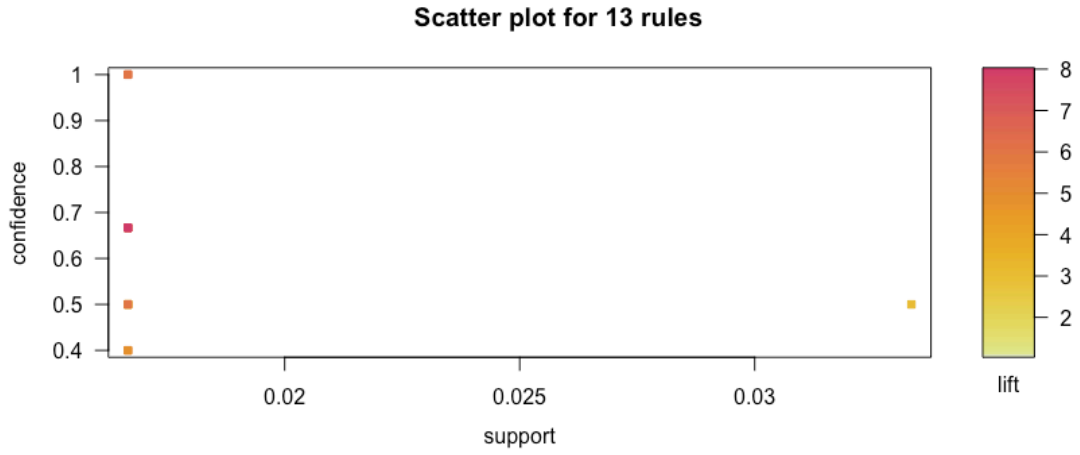

图 14　散点图

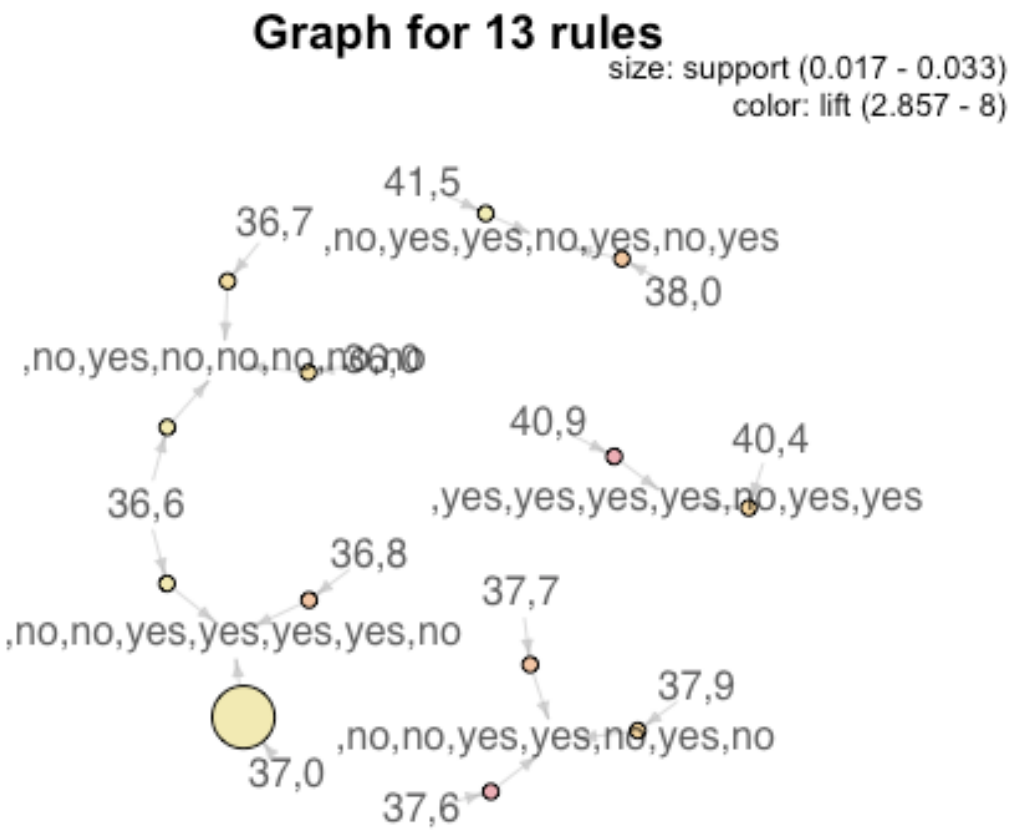

图 15

八、保存结果

```
#保存结果
df.rules=as(rules,"data.frame")
write.csv(df.rules,file = "~/Desktop/Result.csv",row.names = F, quote = F)
```

文件内容如下：

rules	support	confidence	lift								
{36	8} => {	no	no	yes	yes	yes	yes	no}	0.01666667	1	6
{38	0} => {	no	yes	yes	no	yes	no	yes}	0.01666667	1	5.71428571
{36	0} => {	no	yes	no	no	no	no	no}	0.01666667	0.66666667	4
{40	9} => {	yes	yes	yes	yes	no	yes	yes}	0.01666667	0.66666667	8
{36	7} => {	no	yes	no	no	no	no	no}	0.01666667	0.66666667	4
{37	6} => {	no	no	yes	yes	no	yes	no}	0.01666667	0.66666667	8
{37	7} => {	no	no	yes	yes	no	yes	no}	0.01666667	0.5	6
{41	5} => {	no	yes	yes	no	yes	no	yes}	0.01666667	0.5	2.85714286
{36	6} => {	no	yes	no	no	no	no	no}	0.01666667	0.5	3
{36	6} => {	no	no	yes	yes	yes	yes	no}	0.01666667	0.5	3
{40	4} => {	yes	yes	yes	yes	no	yes	yes}	0.01666667	0.4	4.8
{37	9} => {	no	no	yes	yes	no	yes	no}	0.01666667	0.4	4.8
{37	0} => {	no	no	yes	yes	yes	yes	no}	0.03333333	0.5	3

图 16