**关联规则挖掘**

2120151046 魏思杰

**基于R语言进行数据分析，数据集为UCI的“急性炎症”数据集。首先打开diagnosis.data文件，读取表格并加上表头：**

diagnosis <- read.table('diagnosis.data', header=F, dec = ",", sep='\t', fileEncoding="UTF-16LE", as.is=TRUE, col.names=c('temperature','nausea','lumbar','urine','micturition','urethra','bladder','nephritis'))

colum <- list('temperature','nausea','lumbar','urine','micturition','urethra','bladder','nephritis')

**对数据集进行预处理，转换成适合关联规则挖掘的形式，输出到data\_preprocess.csv文件：**

str\_all=""

for(i in 1:nrow(diagnosis))

{

if(as.numeric(diagnosis[i,1])<=36.9){str='normal'}

else if(as.numeric(diagnosis[i,1])<=37.9){str='low'}

else if(as.numeric(diagnosis[i,1])<=39.9){str='midde'}

else if(as.numeric(diagnosis[i,1])<=42.0){str='high'}

for(j in 2:ncol(diagnosis))

{

if(diagnosis[i,j]=="yes"){str <- paste(str,colum[[j]],sep=',')}

}

if(str\_all==''){str\_all=str}

else{str\_all <- paste(str\_all,str,sep='\r')}

}

write(str\_all,'data\_preprocess.csv')

**找出频繁项集，输出到frequent\_items文件：**

a <- read.transactions('data\_preprocess.csv',format = 'basket',sep=',')

sink("summary.txt")

summary(a)

sink()

sink("frequent\_items.txt")

itemsets\_apr = apriori (a, parameter=list(support=0.1,confidence=0.5,maxlen=8,target="frequent itemsets"), control=list(sort=-1))

inspect(itemsets\_apr)

sink()

**导出关联规则，输出到rules.txt文件，计算其支持度和置信度：**

sink("rules.txt")

rules = apriori (a, parameter=list (support=0.1,confidence=0.5,minlen=3,maxlen=8,targe="rules"), appearance=list(rhs=c("bladder","nephritis"), default="lhs"), control=list(sort=-1))

inspect(rules)

sink()

**去除冗余的规则，输出到rules\_delete\_redundant.txt文件：**

去除冗余的规则

subset.matrix<-is.subset(rules,rules)#生成一个所有规则的子集矩阵,行和列分别是每条rules，其中的值是TRUE和FALSE，当rules2是rules1的子集时，rules2在rules1的值为TRUE

#subset.matrix[row(subset.matrix)==col(subset.matrix)]<-NA

subset.matrix[lower.tri(subset.matrix,diag=T)]<-NA#将矩阵对角线以下的元素置为空，只保留上三角

redundant<-colSums(subset.matrix,na.rm=T)>=1#R会将矩阵中的TRUE当做1，统计每列的和（忽略缺失值），如果该列的和大于等于1，也就是表示该列（规则）是别的规则的子集，应该删除。

rules.pruned<-rules[!redundant]#去掉冗余的规则

sink("rules\_delete\_redundant.txt")

inspect(rules.pruned)

sink()

**使用lift对规则进行评价，输出到rules\_delete\_redundant\_sorted\_lift.txt文件：**

sink("rules\_delete\_redundant\_sorted\_lift.txt")

rules.pruned.sorted\_lift = sort(rules.pruned,by='lift')

inspect(rules.pruned.sorted\_lift)

sink()

**使用可视化技术对规则进行展示，分别将散点图、平行坐标、泡泡图输出到scatter\_chart.png、parallel\_coordinates.png、bubble\_chart.png文件：**

library (grid)

library (arulesViz)

#散点图

png(file="scatter\_chart.png", bg="transparent")

plot(rules.pruned.sorted\_lift)

dev.off()

#平行坐标

png(file="parallel\_coordinates.png", bg="transparent")

plot(rules.pruned.sorted\_lift,method="paracoord")

dev.off()

#泡泡图

png(file="bubble\_chart.png", bg="transparent")

plot(rules.pruned.sorted\_lift,method="grouped")

dev.off() ote=F)

结果如图所示：

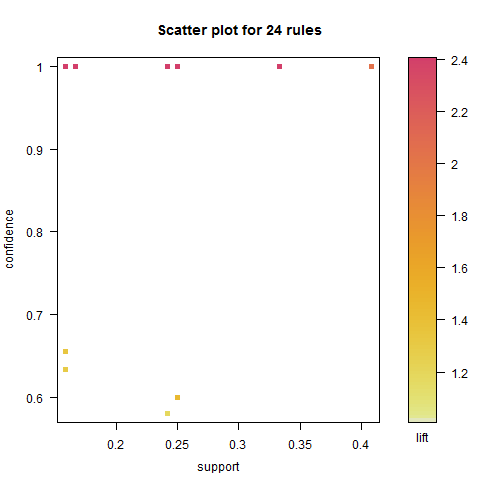


图1 散点图

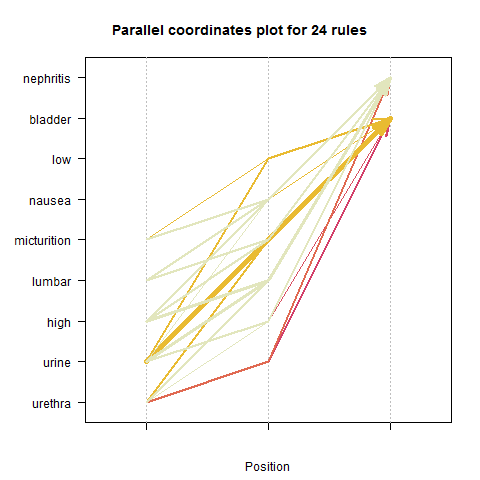


图2 平行坐标

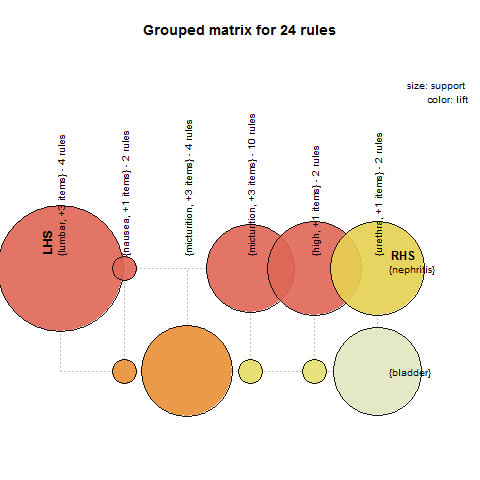


图3 泡泡图