关联分析

Auzzer\_Pang

6/18/2020

rm (list = ls())  
setwd("/Users/auzzer\_pang")  
library(arules)

## Loading required package: Matrix

##   
## Attaching package: 'arules'

## The following objects are masked from 'package:base':  
##   
## abbreviate, write

datafile = read.csv("data.csv",  
 header = T,fileEncoding = 'GBK') # 读取数据  
  
type<-6;index<-8 #数据离散化的分组个数;TNM分期数据所在列  
typelabel<-c("A","B","C","D","E","F")#数据离散化后的标识前缀  
  
set.seed(1212) # 固定随机化种子  
cols<-ncol(datafile[, 1:6]) # 取六种证型列数  
rows<-nrow(datafile[, 1:6]) # 行数  
disdata <- matrix(NA, rows, cols + 1) # 初始化  
for (i in 1:cols) {# 聚类离散化  
 cl <- kmeans(datafile[, i], type, nstart = 20) # 对单个属性列进行聚类  
 disdata[, i] <- paste(typelabel[i], cl$cluster)}  
  
#数据处理2  
##运用apriori算法  
disdata[, cols + 1] <- datafile[,index]  
disdata[, cols + 1] <- paste("H", disdata[, cols + 1], seq = "")  
  
colnames(disdata)<-c("肝气郁结证型系数","热毒蕴结证型系数",  
 "冲任失调证型系数","气血两虚证型系数",  
 "脾胃虚弱证型系数","肝肾阴虚证型系数","TNM分期")  
  
write.csv(disdata,file="processedfile.csv",quote=F,row.names=F)  
a <- read.csv("processedfile.csv", header = TRUE) # 读入数据  
#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#  
trans <- as(a,"transactions") # 将数据转换为transactions格式

## Warning: Column(s) 1, 2, 3, 4, 5, 6, 7 not logical or factor. Applying default  
## discretization (see '? discretizeDF').

inspect(trans[1:5]) # 观测trans数据集中前5行数据items

## items transactionID  
## [1] {肝气郁结证型系数=A 5,   
## 热毒蕴结证型系数=B 1,   
## 冲任失调证型系数=C 4,   
## 气血两虚证型系数=D 2,   
## 脾胃虚弱证型系数=E 5,   
## 肝肾阴虚证型系数=F 5,   
## TNM分期=H H4 } 1  
## [2] {肝气郁结证型系数=A 3,   
## 热毒蕴结证型系数=B 3,   
## 冲任失调证型系数=C 4,   
## 气血两虚证型系数=D 2,   
## 脾胃虚弱证型系数=E 5,   
## 肝肾阴虚证型系数=F 1,   
## TNM分期=H H4 } 2  
## [3] {肝气郁结证型系数=A 5,   
## 热毒蕴结证型系数=B 4,   
## 冲任失调证型系数=C 1,   
## 气血两虚证型系数=D 3,   
## 脾胃虚弱证型系数=E 2,   
## 肝肾阴虚证型系数=F 3,   
## TNM分期=H H4 } 3  
## [4] {肝气郁结证型系数=A 4,   
## 热毒蕴结证型系数=B 5,   
## 冲任失调证型系数=C 4,   
## 气血两虚证型系数=D 4,   
## 脾胃虚弱证型系数=E 3,   
## 肝肾阴虚证型系数=F 5,   
## TNM分期=H H4 } 4  
## [5] {肝气郁结证型系数=A 1,   
## 热毒蕴结证型系数=B 6,   
## 冲任失调证型系数=C 6,   
## 气血两虚证型系数=D 1,   
## 脾胃虚弱证型系数=E 3,   
## 肝肾阴虚证型系数=F 5,   
## TNM分期=H H4 } 5

# 使用apriori函数生成关联规则  
#生成关联规则  
rules <- apriori(trans,parameter=list(support=0.06,  
 confidence=0.75,target="rules"))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.75 0.1 1 none FALSE TRUE 5 0.06 1  
## maxlen target ext  
## 10 rules TRUE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 55   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[40 item(s), 930 transaction(s)] done [0.00s].  
## sorting and recoding items ... [34 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 done [0.00s].  
## writing ... [2 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

# 使用inspect函数提取规则  
inspect(rules) # 观测rules中关联规则

## lhs rhs support   
## [1] {肝肾阴虚证型系数=F 5} => {TNM分期=H H4 } 0.14301075  
## [2] {肝气郁结证型系数=A 1,肝肾阴虚证型系数=F 5} => {TNM分期=H H4 } 0.06666667  
## confidence coverage lift count  
## [1] 0.7556818 0.18924731 1.693456 133   
## [2] 0.8611111 0.07741935 1.929719 62

size(x=rules)

## [1] 2 3

rules.sorted<-sort(x=rules,by="lift",decreasing=TRUE)  
inspect(rules.sorted)

## lhs rhs support   
## [1] {肝气郁结证型系数=A 1,肝肾阴虚证型系数=F 5} => {TNM分期=H H4 } 0.06666667  
## [2] {肝肾阴虚证型系数=F 5} => {TNM分期=H H4 } 0.14301075  
## confidence coverage lift count  
## [1] 0.8611111 0.07741935 1.929719 62   
## [2] 0.7556818 0.18924731 1.693456 133

#保存规则1  
write(rules,file='rules1.csv',sep=',',quote=T,row.names=F)  
  
#筛选关联规则  
rules.D1<-subset(x=rules,subset=size(rules)==3)#考虑两个系数  
inspect(rules.D1)

## lhs rhs support   
## [1] {肝气郁结证型系数=A 1,肝肾阴虚证型系数=F 5} => {TNM分期=H H4 } 0.06666667  
## confidence coverage lift count  
## [1] 0.8611111 0.07741935 1.929719 62

rules.D2<-subset(x=rules,subset=size(rules)==2)#考虑一个系数  
inspect(rules.D2)

## lhs rhs support confidence coverage   
## [1] {肝肾阴虚证型系数=F 5} => {TNM分期=H H4 } 0.1430108 0.7556818 0.1892473  
## lift count  
## [1] 1.693456 133

#保存规则2  
write(rules.D1,file='rules.D1.csv',sep=',',quote=T,row.names=F)  
write(rules.D2,file='rules.D2.csv',sep=',',quote=T,row.names=F)

#病程阶段  
library(Matrix)  
library(arules)  
#数据处理1  
type<-6;index<-7 #数据离散化的分组个数;病程阶段数据所在列  
typelabel<-c("A","B","C","D","E","F")#数据离散化后的标识前缀  
set.seed(1212) # 固定随机化种子  
cols<-ncol(datafile[, 1:6]) # 取六种证型列数  
rows<-nrow(datafile[, 1:6]) # 行数  
disdata = matrix(NA, rows, cols + 1) # 初始化  
for (i in 1:cols) {# 聚类离散化  
 cl <- kmeans(datafile[, i], type, nstart = 20) # 对单个属性列进行聚类  
 disdata[, i] <- paste(typelabel[i], cl$cluster)}  
#数据处理2  
disdata[, cols + 1] <- datafile[,index]  
disdata[, cols + 1] <- paste("S", disdata[, cols + 1], seq = "")  
  
colnames(disdata)<-c("肝气郁结证型系数","热毒蕴结证型系数",  
 "冲任失调证型系数","气血两虚证型系数",  
 "脾胃虚弱证型系数","肝肾阴虚证型系数","病程阶段")  
  
write.csv(disdata,file="processedfile2.csv",  
 quote=F,row.names=F)  
a <- read.csv("processedfile2.csv", header = TRUE) # 读入数据  
  
trans <- as(a,"transactions") # 将数据转换为transactions格式

## Warning: Column(s) 1, 2, 3, 4, 5, 6, 7 not logical or factor. Applying default  
## discretization (see '? discretizeDF').

inspect(trans[1:5]) # 观测trans数据集中前5行数据items

## items transactionID  
## [1] {肝气郁结证型系数=A 5,   
## 热毒蕴结证型系数=B 1,   
## 冲任失调证型系数=C 4,   
## 气血两虚证型系数=D 2,   
## 脾胃虚弱证型系数=E 5,   
## 肝肾阴虚证型系数=F 5,   
## 病程阶段=S S4 } 1  
## [2] {肝气郁结证型系数=A 3,   
## 热毒蕴结证型系数=B 3,   
## 冲任失调证型系数=C 4,   
## 气血两虚证型系数=D 2,   
## 脾胃虚弱证型系数=E 5,   
## 肝肾阴虚证型系数=F 1,   
## 病程阶段=S S4 } 2  
## [3] {肝气郁结证型系数=A 5,   
## 热毒蕴结证型系数=B 4,   
## 冲任失调证型系数=C 1,   
## 气血两虚证型系数=D 3,   
## 脾胃虚弱证型系数=E 2,   
## 肝肾阴虚证型系数=F 3,   
## 病程阶段=S S4 } 3  
## [4] {肝气郁结证型系数=A 4,   
## 热毒蕴结证型系数=B 5,   
## 冲任失调证型系数=C 4,   
## 气血两虚证型系数=D 4,   
## 脾胃虚弱证型系数=E 3,   
## 肝肾阴虚证型系数=F 5,   
## 病程阶段=S S4 } 4  
## [5] {肝气郁结证型系数=A 1,   
## 热毒蕴结证型系数=B 6,   
## 冲任失调证型系数=C 6,   
## 气血两虚证型系数=D 1,   
## 脾胃虚弱证型系数=E 3,   
## 肝肾阴虚证型系数=F 5,   
## 病程阶段=S S4 } 5

##运用apriori算法  
#生成关联规则  
rules <- apriori(trans,  
 parameter=list(support=0.1,  
 confidence=0.90,target="rules"))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.9 0.1 1 none FALSE TRUE 5 0.1 1  
## maxlen target ext  
## 10 rules TRUE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 93   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[40 item(s), 930 transaction(s)] done [0.00s].  
## sorting and recoding items ... [32 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 done [0.00s].  
## writing ... [2 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

inspect(rules) # 观测rules中关联规则

## lhs rhs support   
## [1] {热毒蕴结证型系数=B 1} => {病程阶段=S S3 } 0.1225806  
## [2] {热毒蕴结证型系数=B 3,脾胃虚弱证型系数=E 1} => {病程阶段=S S2 } 0.1172043  
## confidence coverage lift count  
## [1] 0.9047619 0.1354839 5.099567 114   
## [2] 0.9561404 0.1225806 2.615325 109

size(x=rules)

## [1] 2 3

rules.sorted<-sort(x=rules,by="lift",decreasing=TRUE)  
inspect(rules.sorted)

## lhs rhs support   
## [1] {热毒蕴结证型系数=B 1} => {病程阶段=S S3 } 0.1225806  
## [2] {热毒蕴结证型系数=B 3,脾胃虚弱证型系数=E 1} => {病程阶段=S S2 } 0.1172043  
## confidence coverage lift count  
## [1] 0.9047619 0.1354839 5.099567 114   
## [2] 0.9561404 0.1225806 2.615325 109

#保存规则  
write(rules,file='rules\_bincheng.csv',sep=',',quote=T,row.names=F)

#转移部位  
#数据处理1  
type = 6;index = 9 #数据离散化的分组个数;转移部位数据所在列  
typelabel=c("A","B","C","D","E","F")#数据离散化后的标识前缀  
set.seed(1212) # 固定随机化种子  
cols<-ncol(datafile[, 1:6]) # 取六种证型列数  
rows<-nrow(datafile[, 1:6]) # 行数  
disdata <- matrix(NA, rows, cols + 1) # 初始化  
for (i in 1:cols) {# 聚类离散化  
 cl = kmeans(datafile[, i], type, nstart = 20)#对单个属性列进行聚类  
 disdata[, i] = paste(typelabel[i], cl$cluster)}  
  
#数据处理2  
diadata=as.data.frame(disdata)  
diadata[, cols + 1] <- datafile[,index]  
colnames(diadata)<-c("肝气郁结证型系数","热毒蕴结证型系数",  
 "冲任失调证型系数","气血两虚证型系数",  
 "脾胃虚弱证型系数","肝肾阴虚证型系数",  
 "转移部位")  
write.csv(diadata,  
 file="processedfile3.csv",quote=F,row.names=F)  
a = read.csv("processedfile3.csv", header = TRUE) # 读入数据  
trans = as(a,"transactions") # 将数据转换为transactions格式

## Warning: Column(s) 1, 2, 3, 4, 5, 6, 7 not logical or factor. Applying default  
## discretization (see '? discretizeDF').

inspect(trans[1:5]) # 观测trans数据集中前5行数据items

## items transactionID  
## [1] {肝气郁结证型系数=A 5,   
## 热毒蕴结证型系数=B 1,   
## 冲任失调证型系数=C 4,   
## 气血两虚证型系数=D 2,   
## 脾胃虚弱证型系数=E 5,   
## 肝肾阴虚证型系数=F 5,   
## 转移部位=R1} 1  
## [2] {肝气郁结证型系数=A 3,   
## 热毒蕴结证型系数=B 3,   
## 冲任失调证型系数=C 4,   
## 气血两虚证型系数=D 2,   
## 脾胃虚弱证型系数=E 5,   
## 肝肾阴虚证型系数=F 1,   
## 转移部位=R1} 2  
## [3] {肝气郁结证型系数=A 5,   
## 热毒蕴结证型系数=B 4,   
## 冲任失调证型系数=C 1,   
## 气血两虚证型系数=D 3,   
## 脾胃虚弱证型系数=E 2,   
## 肝肾阴虚证型系数=F 3,   
## 转移部位=R2} 3  
## [4] {肝气郁结证型系数=A 4,   
## 热毒蕴结证型系数=B 5,   
## 冲任失调证型系数=C 4,   
## 气血两虚证型系数=D 4,   
## 脾胃虚弱证型系数=E 3,   
## 肝肾阴虚证型系数=F 5,   
## 转移部位=R2} 4  
## [5] {肝气郁结证型系数=A 1,   
## 热毒蕴结证型系数=B 6,   
## 冲任失调证型系数=C 6,   
## 气血两虚证型系数=D 1,   
## 脾胃虚弱证型系数=E 3,   
## 肝肾阴虚证型系数=F 5,   
## 转移部位=R2R5} 5

##运用apriori算法  
  
#生成关联规则  
rules <- apriori(trans,  
 parameter=list(support=0.15,  
 confidence=0.7,target="rules"))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.7 0.1 1 none FALSE TRUE 5 0.15 1  
## maxlen target ext  
## 10 rules TRUE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 139   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[53 item(s), 930 transaction(s)] done [0.00s].  
## sorting and recoding items ... [22 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 done [0.00s].  
## writing ... [2 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

size(x=rules)

## [1] 2 2

rules.sorted<-sort(x=rules,by="lift",decreasing=TRUE)  
inspect(rules.sorted)

## lhs rhs support confidence coverage   
## [1] {肝肾阴虚证型系数=F 3} => {转移部位=R0} 0.1602151 0.7883598 0.2032258  
## [2] {肝肾阴虚证型系数=F 6} => {转移部位=R0} 0.1784946 0.7280702 0.2451613  
## lift count  
## [1] 1.423640 149   
## [2] 1.314768 166

#保存规则  
write(rules,file='rules\_zhuanyi.csv',sep=',',quote=T,row.names=F)