**关联规则挖掘**

计算机学院 战奕竹 2120141077

# 1. 问题描述

泰坦尼克号的沉没是史上最臭名昭著的沉船之一。1912年4月15日,在她的处女航中，泰坦尼克号与冰山相撞后沉没，造成1502人死亡的2224名乘客和机组人员。这耸人听闻的悲剧震惊了国际社会和导致更好的船舶安全法规。

导致海难发生的原因之一,是没有足够的救生艇运载乘客和机组人员。虽然幸存者们有一些运气的成分，但某些群体确实比其他人更容易生存,如妇女、孩子、上流阶层。

请分析哪类人有可能幸存，并用机器学习的手段预测哪些乘客在灾难中生还。

# 2. 数据说明

数据源：[Titanic存活数据](https://www.kaggle.com/c/titanic/data)，其中训练数据有近900条，测试数据有400余条。数据来源于网站：<https://www.kaggle.com/c/titanic/data>

train.csv

# 3. 数据分析要求

## 3.1 对数据集进行处理，转换成适合关联规则挖掘的形式

（1）数据样例：

PassengerId,Survived,Pclass,Name,Sex,Age,SibSp,Parch,Cabin,Embarked,Ticket,Fare1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,,S,A/5 21171,7.25

（2）字段说明：

|  |  |
| --- | --- |
| 字段 | 说明 |
| survived | Survival  (0 = No; 1 = Yes) |
| pclass | Passenger Class  (1 = 1st; 2 = 2nd; 3 = 3rd) |
| name | Name |
| Sex | Sex |
| Age | Age |
| sibsp | Number of Siblings/Spouses Aboard |
| parch | Number of Parents/Children Aboard |
| cabin | Cabin |
| embarked | Port of Embarkation  (C = Cherbourg; Q = Queenstown; S = Southampton) |
| ticket | Ticket Number |
| fare | Passenger Fare |

在本次试验中，使用了R语言的Apriori算法，采用的输入格式是Basket方式，因此需要对数据的格式进行预处理。其中，主要是对数值内容进行预处理。

在Survive字段中，将0替换为为Dead，1替换为Survive。

在Pclass字段中，分别将数字替换为坐席等级：1 = 1st; 2 = 2nd; 3 = 3rd。

对年龄，进行了划分。0-12岁为child，12-20为teenager，20-40为youth，40-60为middle-aged，60以上为elderly。

亲属情况，按照是否为0，简单的分为了有和无两种情况。

在关联规则挖掘中，我们认为，乘客ID，票号，费用以及姓名信息对规则没有任何帮助，故将其去除，只对其他属性进行关联规则分析。

对于缺失项，因其对关联规则抽取意义不大，所以直接去除。

预处理后的文件名称为traindata.csv。

## 3.2找出频繁项集

首先，读取文件，读取数据表时，采用basket方式读入。

使用eclat函数获得频繁项集，设定频繁项集的最小支持度为0.1，设置规则中的最大变元数为4。

#设置工作路径

setwd('F:\\课件\数据挖掘2016\数据挖掘作业二\第二次作业')

#加载算法库

library("Matrix")

library("arules")

#读取数据，以Basket方式

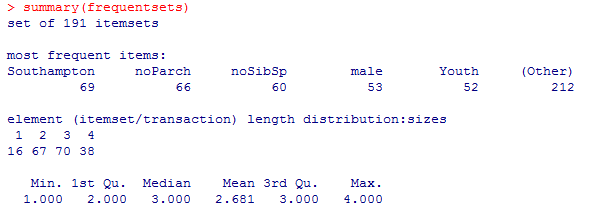
tr<-read.transactions("F:\\课件\数据挖掘2016\数据挖掘作业二\第二次作业\Program\traindata.csv",format="basket",sep=",",rm.duplicates=TRUE)

#获得频繁项集

frequentsets=eclat(tr,parameter=list(support=0.1,maxlen=4))

summary(frequentsets)

抽取完后，使用Summary函数查看抽取的频繁项集分布情况：

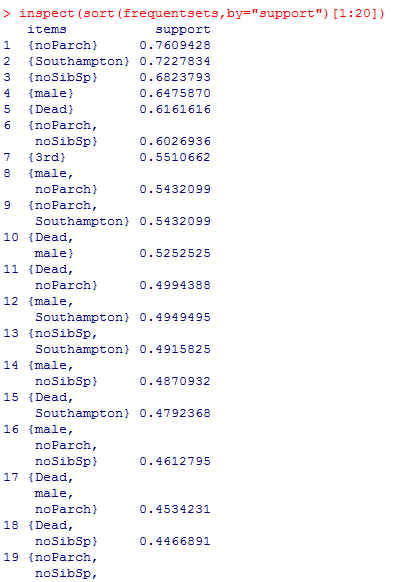


**图1频繁项集的分布情况**

可以看出，总共抽出了191项频繁集，其中3个变元的数目最多，有70条。按照支持度排序，查看抽出的频繁项集：

#查看支持度最高的前20个频繁项集

inspect(sort(frequentsets,by="support")[1:20])



**图2 支持度最大的前20个频繁项集**

其频繁项集的组合整体支持度都在43%以上。

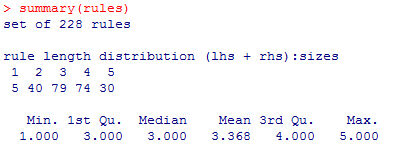
## 3.3导出关联规则，计算其支持度和置信度

使用Apriori算法，抽取关联规则。这里，我们设置的筛选条件为支持度大于20%，置信度大于60%。

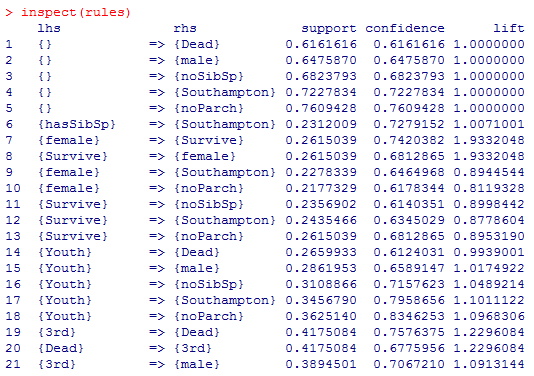
#抽取关联规则

rules = apriori(tr,parameter = list(support = 0.2,confidence = 0.6))

summary(rules)

****

**图3关联规则的变元数目分布情况**



**图4查看抽取的关联规则**

## 3.4去除冗余的规则

抽取了228条关联规则，使用inspect函数查看抽取到的规则，发现许多关联规则并不是我们想要的，我们更希望专注于右边的预测变元（如是否存活），因此，我们需要对规则集进行筛选与评价，获得我们想要的规则。

## 3.5对规则进行评价，可使用Lift，也可以使用教材中所提及的其它指标

(1)按照变元

对于关联规则，首先按照变元进行选取。我们想知道哪种人最可能存活，因此将右变元为存活的规则子集筛选出来：

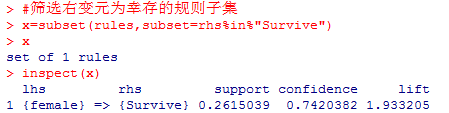
#筛选右变元为幸存的规则子集

x=subset(rules,subset=rhs%in%"Survive")

x

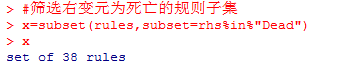
inspect(x)

我们可以得到图5所示结果，只有一条有价值的规则，就是，女性的存活率比较高，有26%的数据(约231条)支持这个结论，其置信度达到了74%。



**图5右变元为存活的规则**

接下来，我们希望得出出哪些人更可能在这场灾难中丧命。因此，抽取右变元为死亡的子集。

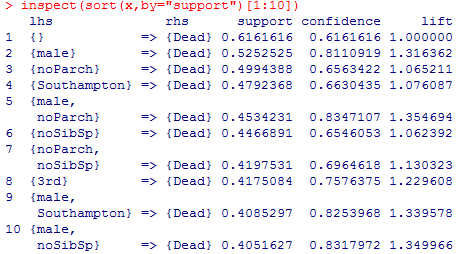


**图6右变元为死亡的规则数目**

以上结果显示此次抽取了38条规则。由于规则的数目较多，需要对其进行评价筛选。这里分别用支持度、置信度与提升值对挖掘出的关联规则进行评价。

#根据支持度对求得的关联规则子集排序并察看

inspect(sort(x,by="support")[1:10])



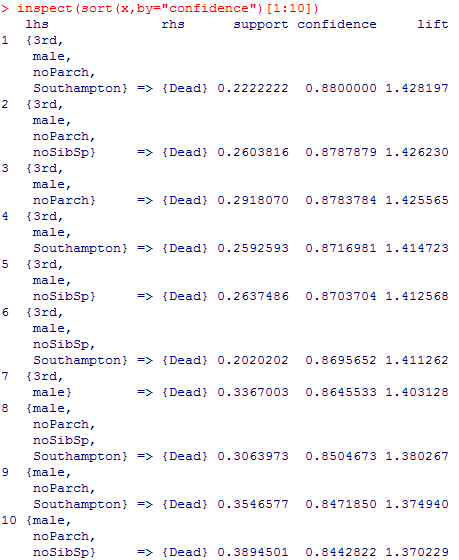
**图6选取支持度最高的前十条“死亡”规则**

上图显示，男性出现的频率较高，并且有较大的数据支持量。

按照置信度对这些规则进行排序，获取到推测更为准确的规则：

#根据置信度对求得的关联规则子集排序并察看

inspect(sort(x,by="confidence")[1:10])

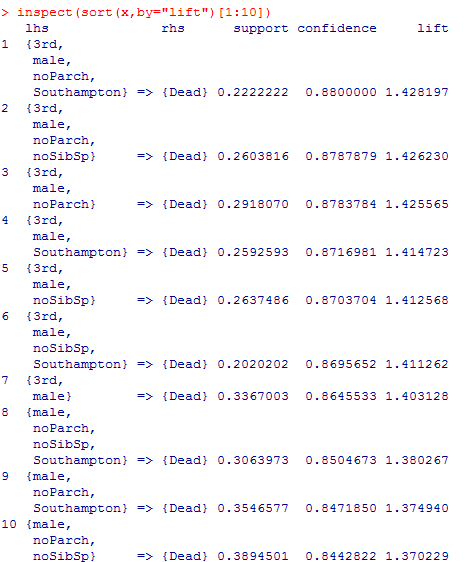


**图7选取置信度最高的前十条“死亡”规则**

按照提升值对这些规则进行排序，可以得到质量较高的规则。

#根据lift对求得的关联规则子集排序并察看

inspect(sort(x,by="lift")[1:10])

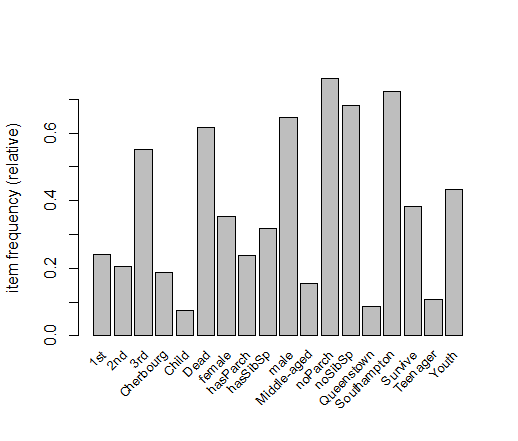


**图6选取置信度最高的前十条“死亡”规则**

出现频率较高的几个关键词为noParch（没有父母或孩子）、noSibSp（没有兄弟姐妹）、男性、以及Southampton登船口。

## 3.6使用可视化技术，如散点图、平行坐标、泡泡图等，对规则进行展示

itemFrequencyPlot(tr,support = 0.05,cex.names =0.8)



# 4. 参考文献

* http://www.360doc.com/content/13/1126/02/9482\_332187754.shtml（关联规则）
* http://blog.csdn.net/dyllove98/article/details/9673791（频繁项集挖掘Apriori JAVA可运行）
* https://m.douban.com/note/435675724/（频繁项集挖掘Apriori）
* http://blog.csdn.net/viewcode/article/details/9122789（频繁项集）
* http://mp.weixin.qq.com/s?src=3&timestamp=1467942705&ver=1&signature=3QG6fo6dYJZSjwOYCRHWDNsSdNgKzrtZ7fxem2EZTdIcxNK1SbRTXb-QKQFoJn7Op6IkOP7e1\*ntf0hyf1o0THjWwKKardWys6Dmxiq1aIrcvoCJW2txorV9Hy2Zon0v0JYi\*35H98h0uKzVx9kFZcLWiQd-ed-SqCR8IXYodyM=（泡泡图之Excel）
* http://blog.infographics.tw/2015/05/d3js-tutorial-bubble-chart/（泡泡图之D3.js）
* http://blog.sina.com.cn/s/blog\_4ff49c7e0102uwdj.html（关联规则评价）
* http://blog.csdn.net/rav009/article/details/8985322（关联规则评价）

# 5. 提交内容说明

* 对数据集进行处理的源程序
* 关联规则挖掘的源程序
* 挖掘结果及分析
* 挖掘过程的报告