目录

[关联规则挖掘 2](#_Toc455584650)

[1. 数据集处理 2](#_Toc455584651)

[1.1数据集的选取 2](#_Toc455584652)

[1.2数据集的处理 2](#_Toc455584653)

[2. 找出频繁项集 3](#_Toc455584654)

[2.1基本术语 3](#_Toc455584655)

[2.2具体实现 4](#_Toc455584656)

[3. 导出关联规则 5](#_Toc455584657)

[3.1Apriori算法 5](#_Toc455584658)

[3.2具体实现 5](#_Toc455584659)

[4. 去除冗余规则 6](#_Toc455584660)

[5. 规则评价 7](#_Toc455584661)

[6. 可视化 8](#_Toc455584662)

[6.1规则分布图 8](#_Toc455584663)

[6.2矩阵分组图 9](#_Toc455584664)

[6.3规则联系图 10](#_Toc455584665)

[关联规则挖掘小结 11](#_Toc455584666)

关联规则是无监督的机器学习方法，用于知识发现，而非预测。关联规则的学习器无需事先对训练数据进行打标签，因为无监督学习没有训练这个步骤。缺点是很难对关联规则学习器进行模型评估，一般都可以通过肉眼观测结果是否合理。

关联规则主要用来发现模式，最经典的应用是购物篮分析，当然其他类似于购物篮交易数据的案例也可以应用关联规则进行模式发现，如电影推荐、约会网站或者药物间的相互副作用。本次试验是对泰坦尼克号的数据进行分析，发现其中的关联规则，从而寻找到影响泰坦尼克成员的成活率的因素。

## 数据集处理

### **1.1数据集的选取**

本次实验我选择的数据集是从链接<https://www.kaggle.com/c/titanic/> 下载的泰坦尼克号中人员信息数据。其中每一位成员包括有”Passenger Id”,” Survived”, “Pclass”, “Name”, “Sex”, “Age”, “SibSp”, “Parch”, “Ticket”, “Fare”,” Cabin”, “Embarked”等共计12项属性。

### 1.2数据集的处理

本次实验的目的是希望探究存活人员与哪些因素有关系，故选择其中”Survived”,” Pclass”, “Sex”,” Age”,” SibSpring”,” Fare”6个属性作为研究对象。从而对原始数据集做了以下处理：

* 1. 对于缺失数据，直接全部删除。
  2. 在Survived属性中，我们使用Survived表示1， notSurvived表示0。
  3. 在Sex属性中，使用female表示女性，male表示男性。
  4. 在Age类型中，年龄小于10的为“kid”；大于10小于20的为“teenager”；大于20小于40的为“youth”；大于40小于60的为“midage”；大于60的为“elder” 。
  5. 在SibSp类型中，有后代的为“Has”，没有后代的为“notHas”。
  6. 在PClass类别中，用”low”代替1，”mid”代替2，”high”代替3。
  7. 在FARE类别中，票价小于20的标记为”poor”;票价大于20小于50的标记为”midclass”;票价大于50小于100的标记为”rich”;票价大于100的标记为”superrich”。

本次实验采用了Python语言，处理结果如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Index | Survived | PClass | Sex | Age | SibSp | Fare |
| 1 | notSurvived | high | male | youth | has | superrich |
| 2 | Survived | low | female | youth | has | superrich |
| 3 | Survived | high | female | youth | nohas | superrich |
| 4 | Survived | low | female | youth | has | superrich |
| 5 | notSurvived | high | male | youth | nohas | superrich |
| 6 | notSurvived | high | male | kid | nohas | superrich |
| 7 | notSurvived | low | male | midage | nohas | superrich |
| 8 | notSurvived | high | male | teenager | nohas | midclass |
| 9 | Survived | high | female | youth | nohas | poor |
| 10 | Survived | mid | female | teenager | has | midclass |

具体文件见per\_data.csv

## 找出频繁项集

### 2.1基本术语

#### 2.1.1项集 ItemSet

这是一条关联规则：http://img.blog.csdn.net/20150420154100822?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvZ2p3YW5nMTk4Mw==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center 。括号内的Item集合称为项集。如上例，{News, Finance}是一个项集，{Sports}也是一个项集。这个例子就是一条关联规则：基于历史记录，同时看过News和Finance版块的人很有可能会看Sports版块。

{News,Finance} 是这条规则的Left-hand-side (LHS or Antecedent)

{Sports}是这条规则的Right-hand-side (RHS or Consequent)

**LHS**（Left Hand Side)的项集和**RHS**（Right Hand Side）的项集不能有交集。

#### 2.1.2支持度 Support

**项集的支持度**就是该项集出现的次数除以总的记录数（交易数）。

支持度的意义在于度量项集在整个事务集中出现的频次。我们在发现规则的时候，希望关注频次高的项集。

#### 2.1.3置信度 Confidence

关联规则 X -> Y 的置信度 计算公式如下 ：

http://img.blog.csdn.net/20150420154115423?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvZ2p3YW5nMTk4Mw==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center

规则的置信度的意义在于项集{X，Y}同时出现的次数占项集{X}出现次数的比例。发生X的条件下，又发生Y的概率。

## 2.2具体实现

**R语言实现代码：**

library(arules)

require(arules)

setwd("D:\Program Files\python3\data mining\code")

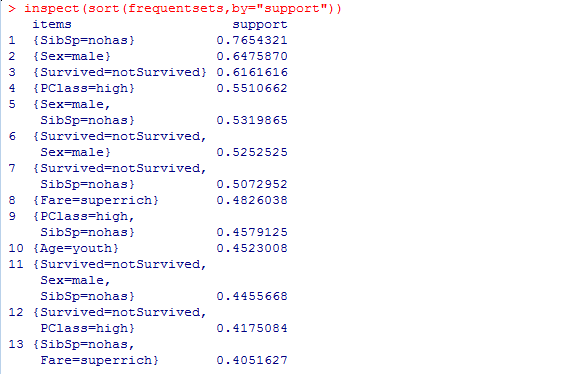
mydata=read.csv("pre\_data.csv")

#找出频繁项集

frequentsets=eclat(mydata,parameter=list(support=0.2,maxlen=10))

inspect(sort(frequentsets,by="support"))#查看求得的频繁项集

取其中支持度大于20%的项，共63项，在此选出前13项的截图如下，剩余结果见文件频繁项集.txt：



## 导出关联规则

### 3.1Apriori算法

如果项集A是频繁的，那么它的子集都是频繁的。如果项集A是不频繁的，那么所有包括它的父集都是不频繁的。例子：{X, Y}是频繁的，那么{X}，{Y}也是频繁的。如果{Z}是不频繁的，那么{X,Z}, {Y, Z}, {X, Y, Z}都是不频繁的。

**生成频繁项集**

给定最小支持度Sup，计算出所有大于等于Sup的项集。

第一步，计算出单个item的项集，过滤掉那些不满足最小支持度的项集。

第二步，基于第一步，生成两个item的项集，过滤掉那些不满足最小支持度的项集。

第三步，基于第二步，生成三个item的项集，过滤掉那些不满足最小支持度的项集。

### 3.2具体实现

**R语言实现代码**如下：

rules=apriori(mydata,parameter=list(support=0.2,confidence=0.2))

summary(rules)

inspect(sort(rules,by="support"))

关联规则的属性如图3.1。

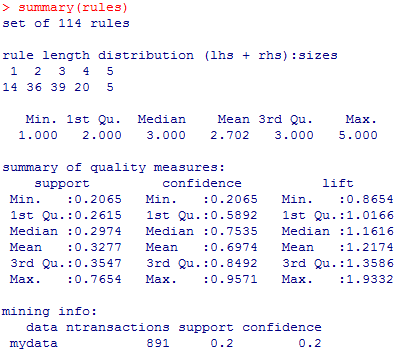


图3.1 关联规则属性

取支持度和置信度均大于0.2的规则，共提取出155条，在此显示**前15条规则**如图3.2，其余见关联规则.txt。

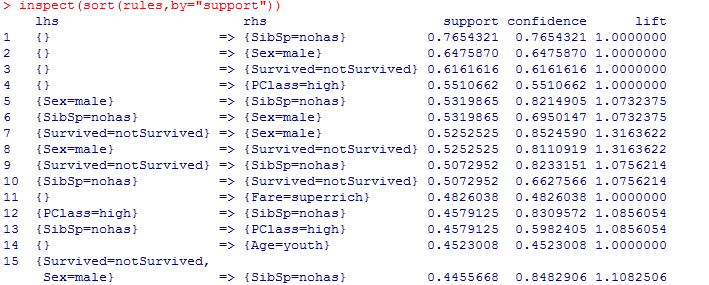


图3.2 关联规则

## 去除冗余规则

上面满足支持度阈值和置信度阈值的规则共有114条，其中有很多的冗余规则。

冗余规则的定义是：如果rules2的lhs和rhs是包含于rules1的，而且rules2的lift小于或者等于rules1，则称rules2是rules1的冗余规则。

下面对冗余规则进行删除，**R语言实现代码**如下，其结果如图4.1。



图4.1 去除冗余规则

## 规则评价

本次试验采用Lift值进行规则评价，具体计算如下：



Lift(X->Y)=Confidence(X->Y)/Support(Y)

规则的提升度的意义在于度量项集{X}和项集{Y}的独立性。即，Lift(X->Y)= 1 表面 {X}，{Y}相互独立。如果该值=1,说明两个条件没有任何关联,如果<1,说明A条件(或者说A事件的发生)与B事件是相斥的,一般在数据挖掘中当提升度大于3时,我们才承认挖掘出的关联规则是有价值的。

**R语言实现代码**如下：

inspect（sort（rules,by="lift"））

结果如下图5.1所示，具体见规则评价.txt



图5.1规则评价

## 可视化

### 6.1规则分布图

规则的散点分布图表示了规则的整体分布，图中每个点的颜色深浅代表了不同的lift的值，颜色越深，lift值越大。

**R语言实现代码**如下，效果如图6.1。

plot(rules)

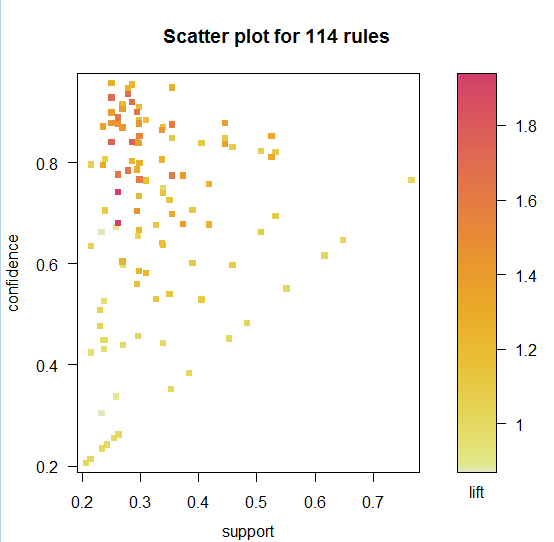


图6.1 整体规则散点图

这幅散点图表示了所有规则的分布图：大部分规则的support在0.3以内，Confidence在0.6-1内。每个点的颜色深浅代表了lift的值。

我们观察去除冗余后的14条规则的整体分布，**R语言实现代码**如下，结果如图6.2。

plot(rules.pruned)

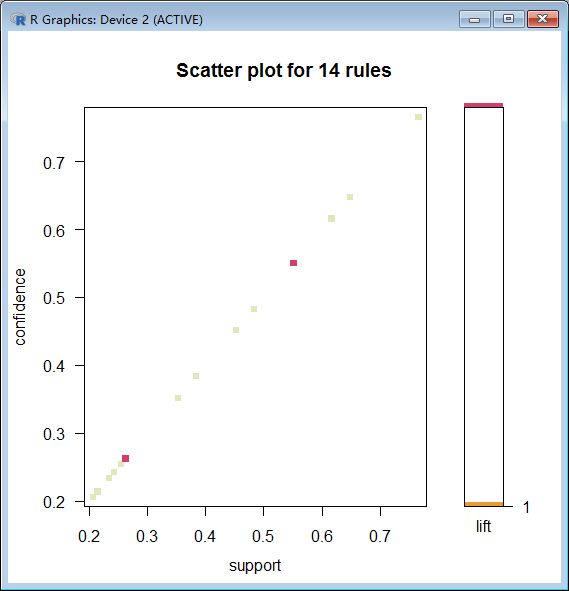


图6.2 去除冗余规则散点图

从图中发现，剩余的14条规则在support和confidence之间呈线性分布。

### 6.2矩阵分组图

**R语言实现代码**如下**，**矩阵分组后的结果如图6.3。

plot(rules,method="grouped")

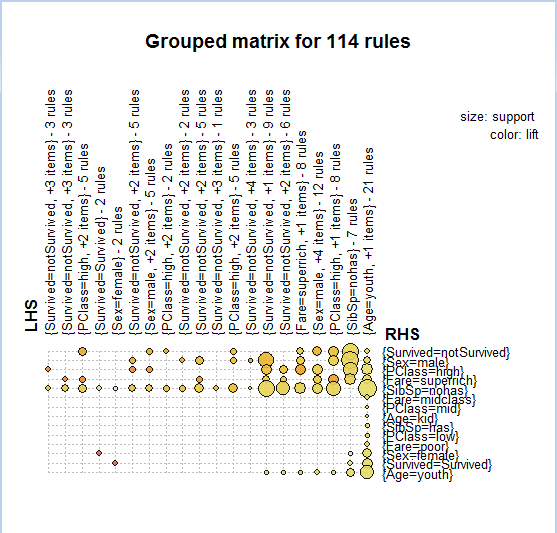


图6.3 规则分组图

### 6.3规则联系图

规则联系图用来展示规则之间的相关联系。其中圆圈大小代表了规则的支持度，支持度越大，圆圈越大。

**R语言实现代码**如下，规则之间的联系图如图6.4。plot(rules,measure="confidence",method="graph",control=list(type="items"),shading="lift")

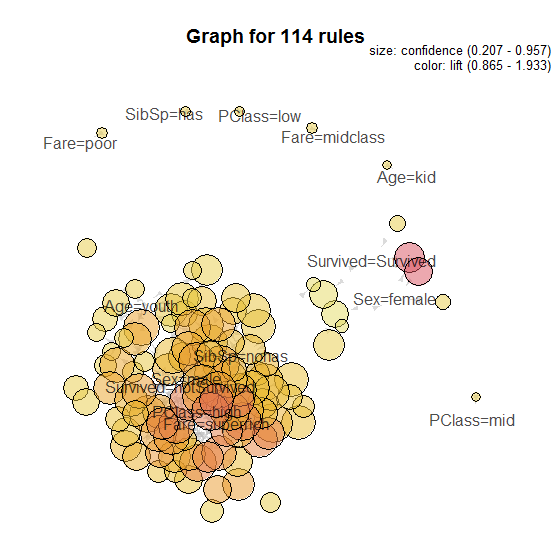


图6.5 整体规则关联图

去除冗余规则后的规则联系图如下，R语言实现代码如下，效果如图6.5。

plot(rules.pruned,measure="confidence",method="graph",control=list(type="items"),shading="lift")

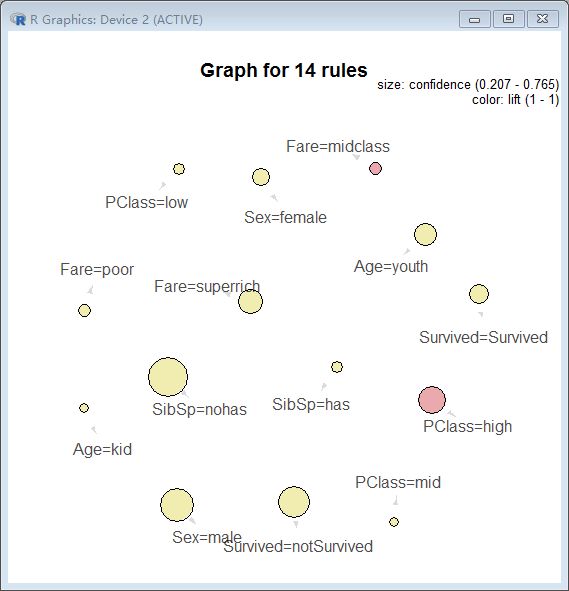


图6.5 去除冗余的规则联系图

## 关联规则挖掘小结

1. 关联规则是发现数据间的关系：可能会共同发生的那些属性co-occurrence
2. 一个好的规则可以用lift进行校验。
3. 当属性（商品）越多的时候，支持度会比较低。
4. 关联规则的发掘是交互式的，需要不断的检查、优化。