关联规则分析

本次报告主要包括以下内容：

> 1. 数据介绍
> 2. 基本原理介绍
> 3. 结合理论进行案例分析
> 4. 最后总结
> 5. 附录加上参考和代码

数据介绍

本次报告所使用的案例数据为购物篮数据，存储在shop_basket.csv文件中。主要有有1000个观测数据，除去前面7个介绍消费者的编号、消费金额、支付方式、性别、是否是本地、收入、年龄变量之外，剩下还有11个商品的数据，这些数据主要介绍每个观测者是否购买这些商品。

本案例的重点在于分析商品之间的关联规则，主要是利用这后面11个商品的数据来进行分析，故在以下的分析中将数据前面7列删除。并且本次使用的数据中有每个列的观测值均是0和1，分别代表不购买、购买这些商品。这里我们需要将每个用户的观测数据进行转换，若在某一行观测数据中，某个字段（变量）的取值是1，则将这个字段名（变量）写入第一个观测值中，以下所有的观测值都这样处理。

基本原理介绍

基本概念：

$\mathcal{T}=i_1,i_2,\ldots,i_m$表示所有项的集合。$\mathcal{T}$的子集称为**项集**。

关联规则的形式为$A\Rightarrow B$，A,B是两个项集，满足$A\cup B=\varnothing$, A和B分别 称为**前项集**和**后项集**。

项集的支持度：项集X的支持度support(X)定义为数据集D的观测中包含X中所有项的比例。

关联规则的支持度：关联规则$A\Rightarrow B$的支持度confidence($A\Rightarrow B$)为数据集D的观测中同时包含A和B中所有项的比例，即support($A\cup B$).

关联规则的置信度：关联规则$A\Rightarrow B$的置信度confidence($A\Rightarrow B$)定义为数据集D中包含A的观测中同时包含B的比例，即support($A\Rightarrow B$)/support(A)，这等价于给定A，B出现的条件概率。

> 在数据挖掘时，需要先指定最小支持度阈值(min_sup)和最小置信度(min_conf)。

强关联规则：支持度不小于min_sup且置信度不小于min_conf的关联规则。其中如果项集A满足最小支持度，那么$A\Rightarrow\varnothing$是强关联规则。

> 在此处，满足强关联规则条件的规则为好规则，这是评价一个规则好坏的一个标准。

Apriori算法

简介：Apriori寻找最有影响力的关联规则挖掘的算法。

步骤：

1. 找到所有频繁项集，那些support $\geq$ min_sup称为频繁项集。
2. 从频繁项集中生成所有强关联规则。

算法的性质：

1. 一个频繁项集的任何自己必然是频繁项集。

2. 一个非频繁项集的任何超集必然是非频繁项集。

对性质的举例子**解释**：

- 如果项集{A}是频繁的，则其子集{A, B}, {A, B,C}也是频繁的，因为A出现，则可以推出{A, B}和{A, B, C}也会出现。
- 同上，如果{A, B}不出现，说明{A}不出现，{B}也不出现。是性质1的逆否命题。

有意义的关联规则

解释关联规则挖掘的结果光有支持度和置信度是不够的，还需要考察当前关联规则的购买某件商品的提升值。其中关联规则$A \Rightarrow B$的**提升值**为：该规则的置信度与B的支持度的比例，即

$$\frac{p(\tilde{B}|\tilde{A})}{p(\tilde{B})}$$

在引入提升值之后，**评价关联规则的好坏**变成，支持度和置信度均不小于相应的最小阈值，且提升度大于1.

案例分析

首先读取数据，并且需要将数据转换成项集数据，将原始商品类数据

fruitveg	freshmeat	dairy	cannedveg	cannedme	frozenmea	beer	wine	softdrink	fish	confection
0	1	1	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	0	0	1
0	0	0	1	0	1	1	0	0	1	0
0	0	1	0	0	0	0	1	0	0	0

通过R语言的这种处理转变成每一行数据只有当前列名为1的那些列名，这些列表代表顾客购买的商品。

```
> inspect(shopBasket[1:4])
    items
[1] {confectionery,dairy,freshmeat}
[2] {confectionery,freshmeat}
[3] {beer,cannedveg,fish,frozenmeal}
[4] {dairy,wine}
```

以上说明，第一个顾客购买了confectionery,dairy,freshmeat三件商品，第二个顾客购买了confectionery,freshmeat两件商品。

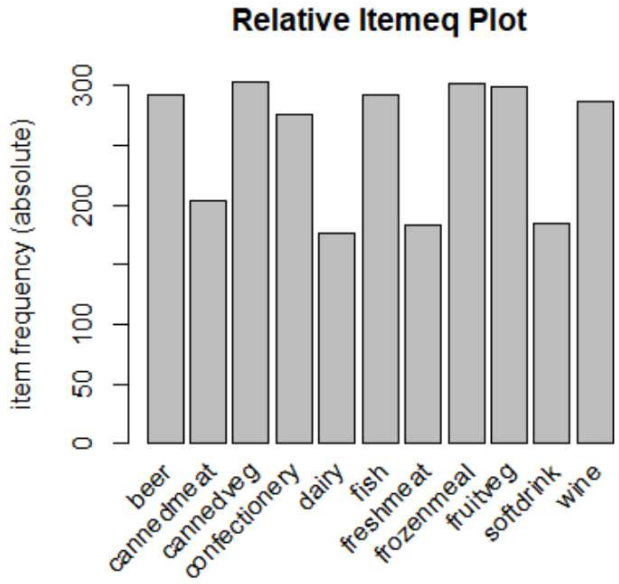
再查看商品篮数据的概览

```
> summary(shopBasket)
transactions as itemMatrix in sparse format with
 1000 rows (elements/itemsets/transactions) and
 11 columns (items) and a density of 0.2545455

most frequent items:
 cannedveg frozenmeal   fruitveg       beer       fish    (Other)
       303        302        299        293        292       1311

element (itemset/transaction) length distribution:
sizes
  0   1   2   3   4   5   6   7   8
 60 174 227 220 175  81  38  21   4

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

```
    0.0     2.0     3.0     2.8     4.0     8.0

includes extended item information - examples:
      labels
1       beer
2 cannedmeat
3  cannedveg
```

由上结果可以得知，数据中有1000个交易，所有购买的商品种类有11种。在一个1000*11的矩阵中，只有25.45455%个单元格有值，其它都是空的。同时也可得知，最常被购买的商品有cannedveg, frozenmeal fruitveg, beer, fish等5类商品。有4次交易同时购买了8种商品，购买商品的数量最少购买0件商品，最多一次性购买8件商品。

其中关于最频繁购买的商品的频率直方图如下：



在以上对数据格式进行预处理之后，使用Apriori算法进行数据之间的关联性分析。

由于在调用Apriori算法的时候，其中有一个sup_min，我们首先设置一个比较低的值，以查看大致的support的取值情况，根据这些结果再进一步选取更加合适的support_min的取值。

调整关联规则的sup_min和conf_min:

首先，固定conf_min=0.5(default)，这里取sup_min为0.01，查看大致的情况：

	Min	1st Qu	Median	Mean	3rd Qu	Max
support	0.01000	0.01200	0.01700	0.02923	0.03100	0.30300

使用`summary(rules)`得到Apriori算法分析的结果：

```
> rules <- apriori(shopBasket,
+                  parameter=list(support=0.01, confidence=0.1,target="rules"))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime
        0.1    0.1    1 none FALSE            TRUE       5
 support minlen maxlen target  ext
    0.01      1     10  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 10

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[11 item(s), 1000 transaction(s)] done [0.00s].
sorting and recoding items ... [11 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 done [0.00s].
writing ... [845 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

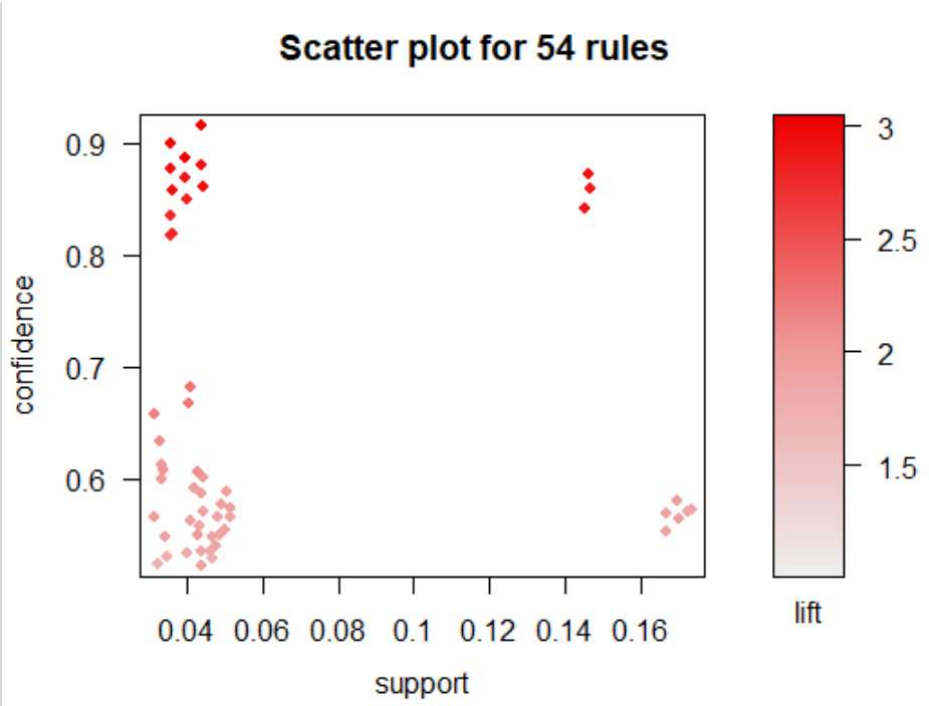
关联规则数为845有些多，因为实际中不需要这么多的关联规则，因此在这里我尝试将取值以上support的上三分位数0.031来作为新的支持度。

同理，固定sup_min，将conf_min设置为0.1查看得到上三分位数为0.5233。最终设置sup_min为0.031，设置conf_min为0.5233来作为后续的最小支持度和最小置信度，此时关联规则的数量为54条，大小合适。

最后得到强关联规则的前6条数据。如下所示，其中每条关联规则的支持度，置信度以及提升度都比较大。根据关联规则好坏的判别准则来讲，以下的关联规则均是在我们设置的规则下的强关联规则。

```
> inspect(head(rules,by="lift"))
    lhs                                rhs           support confidence coverage lift  count
[1] {beer,cannedveg,fish}           => {frozenmeal} 0.044   0.9167     0.048    3.035 44
[2] {cannedveg,frozenmeal,fruitveg} => {beer}       0.040   0.8889     0.045    3.034 40
[3] {beer,cannedmeat,frozenmeal}    => {cannedveg}  0.036   0.9000     0.040    2.970 36
[4] {cannedveg,fish,frozenmeal}     => {beer}       0.044   0.8627     0.051    2.945 44
[5] {cannedveg,frozenmeal,wine}     => {beer}       0.036   0.8571     0.042    2.925 36
[6] {beer,cannedmeat,cannedveg}     => {frozenmeal} 0.036   0.8780     0.041    2.907 36
```

画图可视化最后得出强关联规则的支持度、置信度及提升度的分布：



可见，数据关联规则的置信度主要集中在上图的四个角，因此我们可以多加关注那些支持度、置信度以及提升度位于以上四角的商品。但是如果仅仅是在保证支持度和置信度的情况下，想要极大地提升关联规则的置信度的话，可以考虑将上方左右两角的商品放在一起。

总结反思

本案例主要分析了一下shop_basket.csv数据中顾客购买的商品之间的关联规则，调用R语言中的Apriori算法来分析这些关联规则，此处并没有考虑到之前所给出的顾客的消费金额以及收入等信息，这些仅仅是关联规则的初步分析。所以在以后的学习中，学习完了数据挖掘的建模算法之后，还需要回来将前7列的变量列入我们的分析范畴之内，以便挖掘出更加完整的信息。

此外，在进行选择min_sup和min_conf的时候，我个人仅凭借着自身的直觉判断了一下min_sup和min_conf的取值。仅仅从（1）强规则的个数。（2）设置一个比较低的min_sup和min_conf值，固定其中一个值查看min_sup的上三分位数（3）调节min_sup之后，再查看min_conf的上三分位数来作为最终的min_conf。这还没有参考到完善的数据挖掘理论，此处还需要进一步学习以改善。

最后画图分析查看强关联规则的支持度、可信度以及支持度的分布，更加清晰与直观。

参考

[1]薛薇.R语言数据挖掘[M] 北京:中国人民大学出版社.2016.324-339

[2]普拉迪帕塔 · 米什拉.R语言数据挖掘：实用项目解析[M] 北京:机械工业出版社.2017.112-121

[3] 关联规则transaction数据准备

[4] 【Python数据挖掘课程】八.关联规则挖掘及Apriori实现购物推荐

[5] 关联分析(3):Apriori R语言实现

代码

```
library(arules)
library(arulesViz)

# ----------1 读取数据
setwd("D:/lagua/CODING/R-learn/R-code/Chap4_AssociationRule")
shopBasket = read.csv("shop_basket.csv", sep=",")
# 去除前7个与购买商品无关的数据
shopBasket = shopBasket[, -seq(1, 7, 1)]

# ----------2 查看数据、转换数据、画图查看
# ----------2 查看数据
# 查看前4个观测数据顾客购买的商品
# 等价于inspect(shopBasket[1:4])
for (i in 1:4){
  print(names(shopBasket[i,])[shopBasket[i,]==1])
}

# ----------2 转换数据
# 将数据每一行字段取值为1的列名拿出来，这些列名代表每一条观测购买的商品
shopBasket <- apply(shopBasket,1,function(x) names(x)[x==1])
# 或者直接list(shopBasket)
# 再shopBaseket = as(shopBasket, "transactions")
# 转换成Apriori可以识别的数据类型
shopBasket = as(shopBasket, "transactions")
```

```r
# 查看的概览
summary(shopBasket)
inspect(shopBasket[1:4])

# ----------2 画图查看
help("itemFrequencyPlot")
itemFrequencyPlot(shopBasket, support=0.01,
                  main="Relative Itemeq Plot",
                  type="absolute")


# ----------3 关联分析 调整sup_min, conf_min
# 关联分析 初步分析
help(apriori)
rules <- apriori(shopBasket,
                 parameter=list(support=0.01, confidence=0.1,target="rules"))
summary(rules)

# 固定min_conf 设置support为0.031（上三分位数）
rules <- apriori(shopBasket,
                 parameter=list(support=0.031, confidence=0.1,target="rules"))
summary(rules)

# 调整min_sup之后并固定 设置confidence为0.5233（上三分位数）
rules <- apriori(shopBasket,
                 parameter=list(support=0.031,
confidence=0.5233,target="rules"))
summary(rules)


# ----------4 查看关联规则，按照support, confidence, lift排序
# 把所有规则按照lift（提升度）排序查看关联规则
shopBasket.sorted<-sort(x=rules,by="lift",decreasing=TRUE)
inspect(shopBasket.sorted)

# 逐条查看数据集shopBasket.sorted的前6条记录
# 这其实跟前面排序是等价的
inspect(head(shopBasket.sorted))

# 查看分析结果
options(digits=4)
#设置输出小数位数为4位数
inspect(head(rules,by="lift"))
# inspect函数逐条查看关联规则
#  by="lift"指定按提升值降序排列。


# ----------5 关联分析结果可视化
plot(rules)
# 对关联规则的支持度、置信度和提升值进行可视化
```