数据挖掘作业二

——关联规则挖掘

姓名： 高建花

班级： 硕士 4 班

学号： 2120171010

关联规则挖掘

1. 问题描述

关联规则挖掘主要用于发现大量数据中项集之间有趣的关联或相关联系。如果两项或多项属性之间存在关联，那么其中一项的属性就可以依据其他属性值进行预测。它在数据挖掘中是一个重要的课题，最近几年已被业界所广泛研究。

关联规则挖掘的一个典型例子是购物篮分析。关联规则研究有助于发现交易数据库中不同商品（项）之间的联系，找出顾客购买行为模式，如购买了某一商品对购买其他商品的影响。分析结果可以应用于商品货架布局、货存安排以及根据购买模式对用户进行分类。

最著名的关联规则是 Apriori 算法。关联规则挖掘问题可以分为两个子问题：第一步是找出事务数据库中所有大于等于用户指定的最小支持度的数据项集，也就是频繁项集；第二步是利用频繁项集生成所需要的关联规则，根据用户设定的最小置信度进行取舍，最后得到强关联规则。识别或发现所有频繁项目集市关联规则发现算法的核心。

本实验利用 Apriori 算法对数据集 San Francisco Building Permits 进行关联规则挖掘，主要实验过程如下。

2. 实验环境

Item	**Description**
Language	R
IDE	RGui
Package	arules; arulesViz

3. 关联规则挖掘

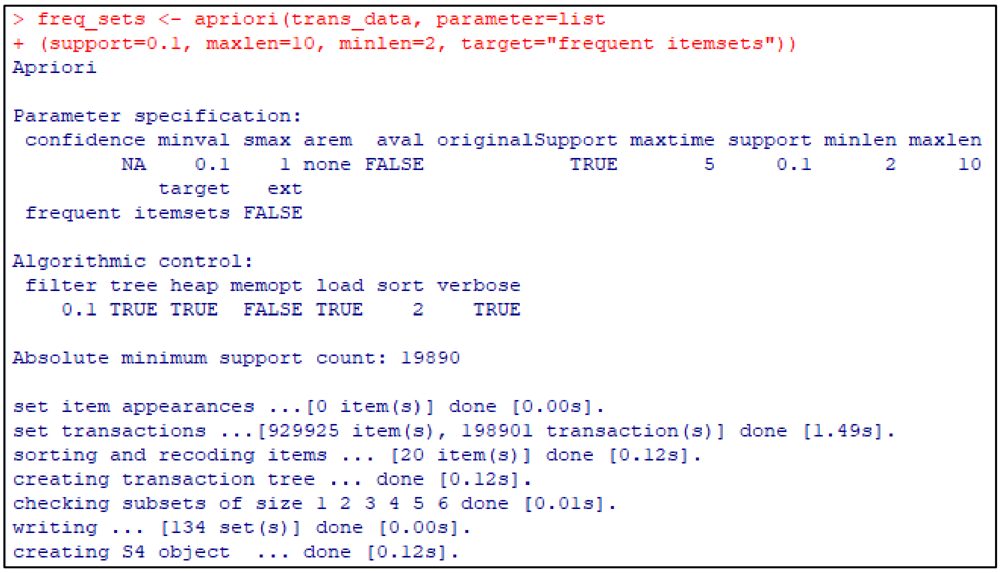
3.1 数据转换

利用 read.transactions 函数在读入数据的同时，将数据转换为 transactions 格式：

```
trans_data <- read.transactions("Building_Permits.csv")
```

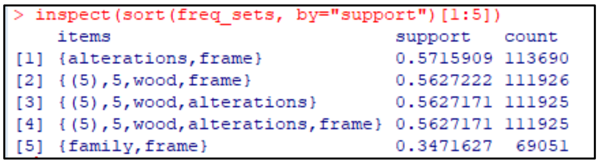
3.2 频繁项集

利用 Apriori 算法得到满足其支持度、置信度、最大长度等阈值的频繁项集：



```
> freq_sets <- apriori(trans_data, parameter=list
+ (support=0.1, maxlen=10, minlen=2, target="frequent itemsets"))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen
         NA    0.1    1 none FALSE            TRUE       5     0.1      2     10
            target   ext
 frequent itemsets FALSE

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

Absolute minimum support count: 19890

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[929925 item(s), 198901 transaction(s)] done [1.49s].
sorting and recoding items ... [20 item(s)] done [0.12s].
creating transaction tree ... done [0.12s].
checking subsets of size 1 2 3 4 5 6 done [0.01s].
writing ... [134 set(s)] done [0.00s].
creating S4 object  ... done [0.12s].
```

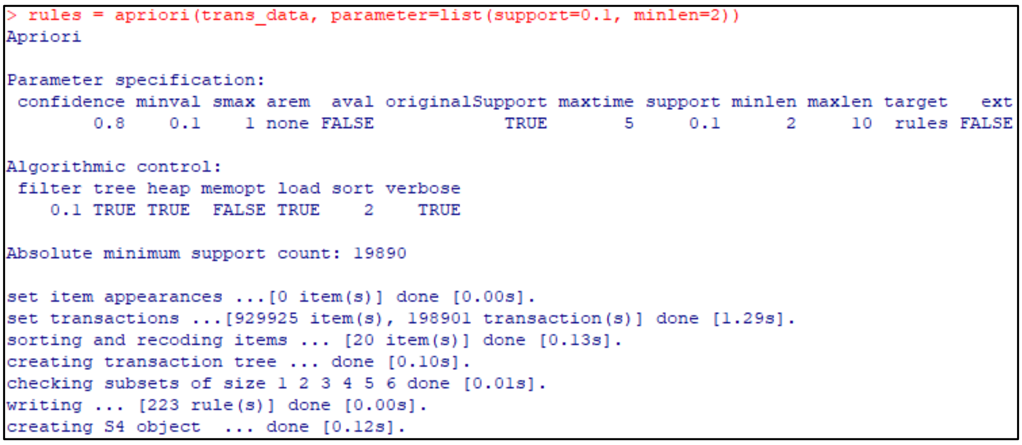
可见设定阈值后，满足条件的交易数据共有 134 项，对频繁项集排序后查看前五项：



```
> inspect(sort(freq_sets, by="support")[1:5])
    items                             support   count
[1] {alterations,frame}               0.5715909 113690
[2] {(5),5,wood,frame}                0.5627222 111926
[3] {(5),5,wood,alterations}          0.5627171 111925
[4] {(5),5,wood,alterations,frame}    0.5627171 111925
[5] {family,frame}                    0.3471627  69051
```

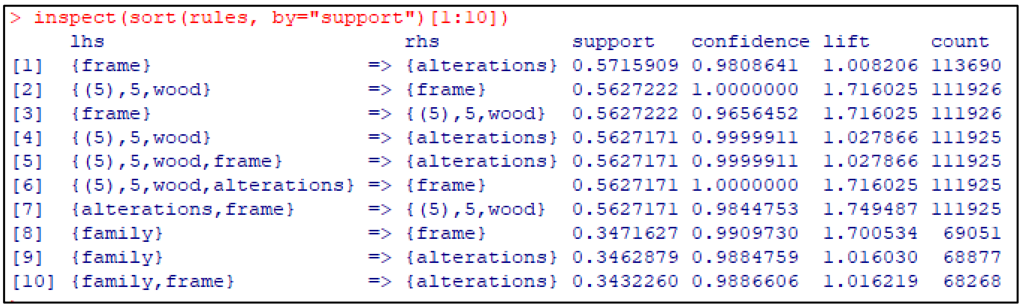
3.3 导出关联规则并计算支持度和置信度

利用 Apriori 算法导出关联规则如下：



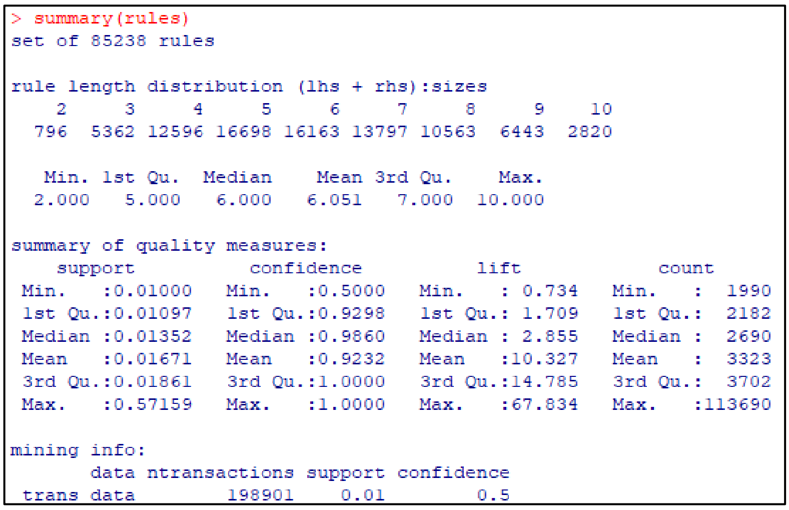
```
> rules = apriori(trans_data, parameter=list(support=0.1, minlen=2))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target   ext
        0.8    0.1    1 none FALSE            TRUE       5     0.1      2     10  rules FALSE

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

Absolute minimum support count: 19890

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[929925 item(s), 198901 transaction(s)] done [1.29s].
sorting and recoding items ... [20 item(s)] done [0.13s].
creating transaction tree ... done [0.10s].
checking subsets of size 1 2 3 4 5 6 done [0.01s].
writing ... [223 rule(s)] done [0.00s].
creating S4 object  ... done [0.12s].
```

以置信度为标准对其排序后，查看 rules 的前五项



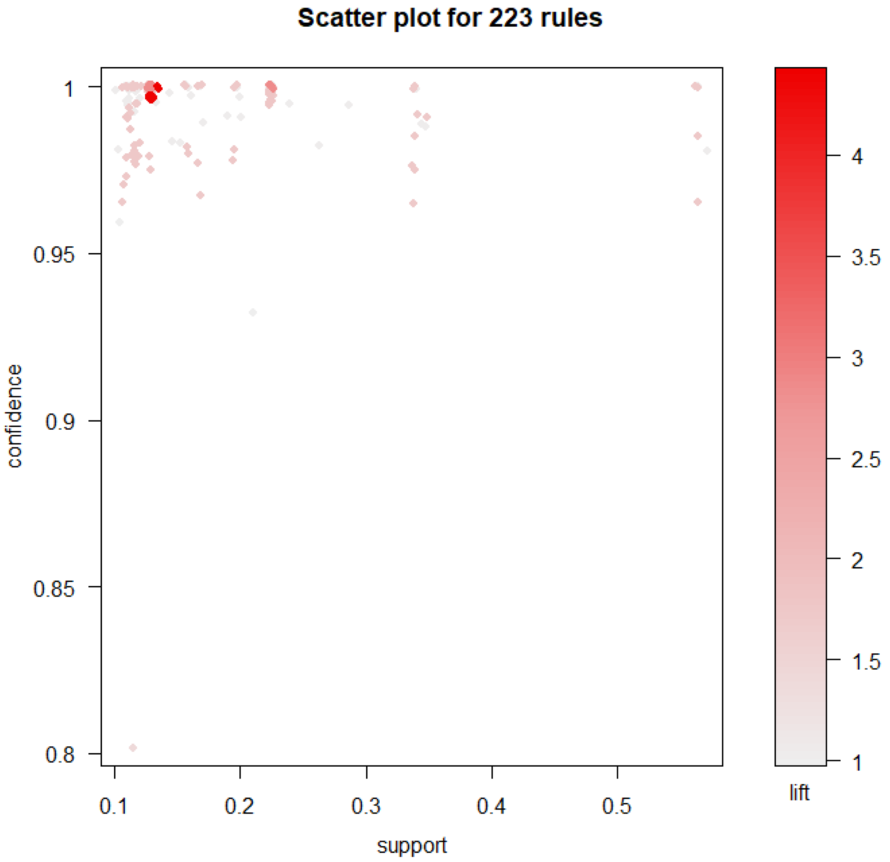
```
> inspect(sort(rules, by="support")[1:10])
     lhs                            rhs             support   confidence lift     count
[1]  {frame}                     => {alterations} 0.5715909 0.9808641  1.008206 113690
[2]  {(5),5,wood}                => {frame}       0.5627222 1.0000000  1.716025 111926
[3]  {frame}                     => {(5),5,wood}  0.5627222 0.9656452  1.716025 111926
[4]  {(5),5,wood}                => {alterations} 0.5627171 0.9999911  1.027866 111925
[5]  {(5),5,wood,frame}          => {alterations} 0.5627171 0.9999911  1.027866 111925
[6]  {(5),5,wood,alterations}    => {frame}       0.5627171 1.0000000  1.716025 111925
[7]  {alterations,frame}         => {(5),5,wood}  0.5627171 0.9844753  1.749487 111925
[8]  {family}                    => {frame}       0.3471627 0.9909730  1.700534  69051
[9]  {family}                    => {alterations} 0.3462879 0.9884759  1.016030  68877
[10] {family,frame}              => {alterations} 0.3432260 0.9886606  1.016219  68268
```

利用 summary 函数查看 rules 的情况，可看到其支持度、置信度和提升度等值的统计信息：



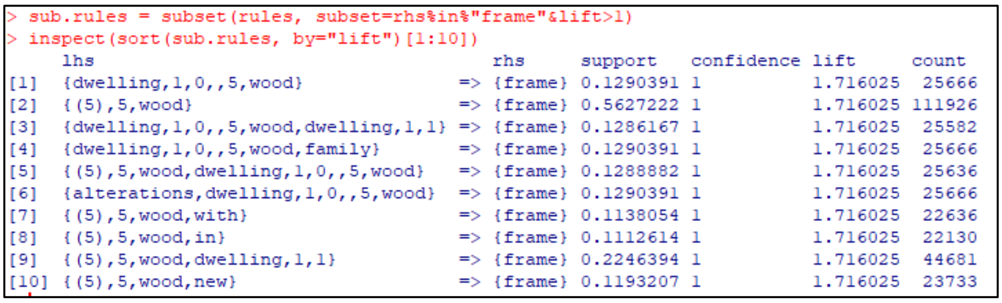
```
> summary(rules)
set of 85238 rules

rule length distribution (lhs + rhs):sizes
    2     3     4     5     6     7     8     9    10 
  796  5362 12596 16698 16163 13797 10563  6443  2820 

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  2.000   5.000   6.000   6.051   7.000  10.000 

summary of quality measures:
    support          confidence          lift            count        
 Min.   :0.01000   Min.   :0.5000   Min.   : 0.734   Min.   :  1990  
 1st Qu.:0.01097   1st Qu.:0.9298   1st Qu.: 1.709   1st Qu.:  2182  
 Median :0.01352   Median :0.9860   Median : 2.855   Median :  2690  
 Mean   :0.01671   Mean   :0.9232   Mean   :10.327   Mean   :  3323  
 3rd Qu.:0.01861   3rd Qu.:1.0000   3rd Qu.:14.785   3rd Qu.:  3702  
 Max.   :0.57159   Max.   :1.0000   Max.   :67.834   Max.   :113690  

mining info:
       data ntransactions support confidence
 trans_data        198901    0.01        0.5
```

可视化 rules 如下



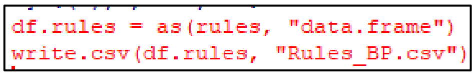
3.4 评价

利用 subset 函数以及条件逻辑表达式来得到所需要的关联规则子集，举例如下（此处是在右端项中包含 frame 且提升度大于 1 的子集）：



```
> sub.rules = subset(rules, subset=rhs%in%"frame"&lift>1)
> inspect(sort(sub.rules, by="lift")[1:10])
     lhs                                       rhs     support   confidence lift     count
[1]  {dwelling,1,0,,5,wood}                 => {frame} 0.1290391 1          1.716025  25666
[2]  {(5),5,wood}                           => {frame} 0.5627222 1          1.716025 111926
[3]  {dwelling,1,0,,5,wood,dwelling,1,1}    => {frame} 0.1286167 1          1.716025  25582
[4]  {dwelling,1,0,,5,wood,family}          => {frame} 0.1290391 1          1.716025  25666
[5]  {(5),5,wood,dwelling,1,0,,5,wood}      => {frame} 0.1288882 1          1.716025  25636
[6]  {alterations,dwelling,1,0,,5,wood}     => {frame} 0.1290391 1          1.716025  25666
[7]  {(5),5,wood,with}                      => {frame} 0.1138054 1          1.716025  22636
[8]  {(5),5,wood,in}                        => {frame} 0.1112614 1          1.716025  22130
[9]  {(5),5,wood,dwelling,1,1}              => {frame} 0.2246394 1          1.716025  44681
[10] {(5),5,wood,new}                       => {frame} 0.1193207 1          1.716025  23733
```

最后将挖掘结果写入文件：



```
df.rules = as(rules, "data.frame")
write.csv(df.rules, "Rules_BP.csv")
```