

目录

一、	实验软件	3
1.1	R	3
1.2	Rstudio	3
二、	基于规则的分类	5
2.1	实验目的	5
2.2	数据来源与说明	5
2.3	算法描述	5
2.4	实验过程及结果分析	8
三、	基于规则组合预测	19
3.1	实验目的	19
3.2	数据来源与说明	19
3.3	算法描述	19
3.4	实验过程及结果分析	21
四、	懒惰学习——基于近邻的分类预测	27
4.1	实验目的	27
4.2	数据来源与说明	27
4.3	算法描述	27
4.4	实验过程及结果分析	29
五、	基于神经网络的分类预测	35
5.1	实验目的	35
5.2	数据来源与说明	35
5.3	算法描述	36
5.4	实验过程及结果分析	37
六、	贝叶斯分类	43
6.1	实验目的	43
6.2	数据来源与说明	43
6.3	算法描述	43

6.4 实验过程及结果分析	43
七、 基于支持向量机的分类预测	49
7.1 实验目的	49
7.2 数据来源与说明	49
7.3 算法描述	49
7.4 实验过程及结果分析	52
八、 常规聚类	61
8.1 实验目的	61
8.2 数据来源与说明	61
8.3 算法描述	61
8.4 实验过程及结果分析	62
九、 特色聚类 kohonen 网络聚类	67
9.1 实验目的	67
9.2 数据来源与说明	67
9.3 算法描述	67
9.4 实验过程及结果分析	67
十、 发现数据中的关联特征	71
10.1 实验目的	71
10.2 数据来源与说明	71
10.3 算法描述	71
10.4 实验过程及结果分析	71
十一、 总结	75

附录

一、 实验软件

1.1 R

R 是一种为统计计算和绘图而生的语言和环境，它是一套开源的数据分析解决方案，由一个庞大且活跃的全球性研究型社区维护。R 语言具备可扩展能力且拥有丰富的功能选项，帮助开发人员构建自己的工具及方法，从而顺利实现数据分析。R 可运行与多种平台之上，包括 Windows、Unix 和 Mac OS X。这基本上意味着它可以运行于你所能拥有的任何计算机上。多数商业统计软件价格不菲，而 R 是免费的，国际上 R 语言已然是专业数据分析领域的标准。

1.2 Rstudio

Rstudio 界面简单地分为四个窗口，从左至右分别是程序编辑窗口，工作空间与历史信息，程序运行与输出窗口（控制台），画图和函数包帮助窗口。

1. 控制台（Console）

控制台功能与 RGui 中相同，显示程序运行的信息。Rstudio 提供的辅助功能有助于初学者顺利的输入函数，比如忘记画图函数 `plot`，输入前几位字母，如 `pl`，再按 `Tab` 键，会出现所有已安装的程序包中以 `pl` 开头的函数及简要介绍，回车键即可选择。同时，`Tab` 键还可以显示函数的各项参数，输入 `plot(`，Rstudio 会自动补上右括号，按 `Tab` 键则显示 `plot()` 的各项参数。在控制台中，`ctrl+向上键` 可以显示出最近运行的函数历史列表。如果重复运行前面刚进行的程序，该操作可以很方便的进行。

2. 程序编辑窗口

首先，系统会默认一个叫 `Untitled1*` 的编辑窗口（source editor）`File->New File->R script(或 Ctrl+Shift+N)` 中可以新建脚本窗口。`File-> New Project->New Directory->Empty Project` 可以创建项目

3. 工作空间（Workspace）和历史（History）窗口。

工作空间显示的是定义的数据集 Data，值 Value 和自定义函数 Function，可以选中双击打开查看。import Dataset 可以快速导入 Excel、CSV、SPSS 等格式的数据。历史窗口显示的是历史操作，可以选中点击上方 To Console 使其进入主控制界面，与重复以前的操作类似。

4. 画图和帮助窗口。

这个窗口的功能容易理解，包括输出图形、显示函数的帮助文件、显示包、帮助文档、观众栏。更多的帮助与信息可以点击 Help->Rstudio Docs，参考 Rstudio 的官方文档。

二、 基于规则的分类

2.1 实验目的

1. 学习决策树分类预测的基本原理，并用其产生 if-then 规则
2. 掌握对决策树进行剪枝的原则与方法，实现对新的数据的分类预测；
3. 学习分类模型的评价方法，在混淆矩阵的基础上学会计算 specificity、sensitivity 等一系列指标
4. 绘制 Gain Chart、Lift Curve 和 ROC 曲线，最终实现对模型的分类性能的评价。

2.2 数据来源与说明

1. 数据来源：UCI ， <http://archive.ics.uci.edu/ml/datasets/Iris>
2. 数据名称：台湾新竹市的输血服务中心输血数据，该数据集共有 748 个实例，包含 5 个属性，通过对以上数据属性的分析预测出在 2007 年的 3 月是否会再次献血。（见附件一）
3. 数据属性与说明：
 - （1） 新进度-距上一次捐赠的时间跨度（Recency）
 - （2） 频率-捐赠总数（Frequency）
 - （3） 总血液捐赠量（Monetary）
 - （4） 时间-距第一次捐赠的时间跨度（Time）
 - （5） 二进制变量（0 表示不献血，1 表示献血）

2.3 算法描述

1. 决策树主要是通过产生“if-then”规则来实现分类预测，有两大核心问题：

(1) 决策树的生长，即利用训练样本集完成决策树的建立过程。

a) 决策树模型一般不建立在全部观测数据上。首先要将全部样本分为训练样本集和测试样本集，主要使用以下两种方法：

➤ 旁置法：将整个样本集随机划分为两个集合，训练样本集通常包含 60% 至 70% 的观测，适合样本量较大的情况。

➤ 留一法：是在包含 n 个观测的样本中，抽出一个观测作为测试样本集，剩余的 $n-1$ 个观测作为训练样本集，这个过程重复 n 次，计算 n 个预测误差的平均值。

b) 分类树生长的本质是如何从众多的输入变量中选择当前的最佳分组变量，而这个选择应该使输出变量的异质性下降最快，主要的测度指标有以下三种：

➤ 信息增益：信息熵是信息量的数学期望，是信源发出信息前的平均不确定性，也称先验熵。其数学定义为：

$$\text{Ent}(U) = \sum_i P(u_i) \log_2 \frac{1}{P(u_i)} = - \sum_i P(u_i) \log_2 P(u_i)$$

进一步，在信宿收到信息 $V = v_j$ 时，此时信源的不确定性修改为：

$$\text{Ent}(U|v_j) = \sum_i P(u_i|v_j) \log_2 \frac{1}{P(u_i|v_j)} = - \sum_i P(u_i|v_j) \log_2 P(u_i|v_j)$$

称为后验信息熵，表示信宿收到 V 后，对信息 U 仍存在的确定性，这是由随机干扰引起的。通常， $\text{Gains}(U,V) = \text{Ent}(U) - \text{Ent}(U|V)$ 称为信息增益，反映的是信息消除随机不确定性的程度。

➤ Gini 系数：分类树采用 Gini 系数测度输出变量的异质性。Gini 的数学定义为： $G(t) = 1 - \sum_{j=1}^k p^2(j|t)$ ，分类树采用 Gini 系数的减少量测度异质性下降的程度，最佳的分组变量是 $\Delta G(t)$ 变化最大。

➤ 信息增益率：它是在信息增益的基础上考虑惩罚参数，偏向取值较少的特征。

(2) 决策树的剪枝，即利用测试样本集对形成的决策树进行精简。

a) 分类回归树采用的后剪枝技术称为最小代价复杂度剪枝法：首先对中间

节点的代价复杂度进行测度，之后计算中间节点的子树的代价复杂度。

- b) 如果前者大于后者，则应该保留子树，否则减掉子树，当两者的代价复杂度相等时，复杂度参数 $\alpha = \frac{R(t)-R(T_t)}{|T_t|-1}$ 小时。

- c) 剪枝的关键问题在于复杂度参数的确定，为此可以采用 N 折交叉验证：

首先将数据集随机近似的等分为不相交的 N 组，然后令其中的 N-1 组为训练样本集用于建立模型，剩余的一组为测试样本集，用于计算误差。

最终建立 N 个预测模型，得出 N 个预测误差的平均值作为模型真实预测误差的估计。

2. 分类器性能的评价

(1) 混淆矩阵

当我们基于训练数据建立好决策树分类模型之后，根据模型对记录的分类正误情况，可以定义四个概念：

表 1 混淆矩阵

		PREDICT	
		P	N
ACTUAL	P	TP	FN
	N	FP	TN

其中，TP 表示实际为 positive 类并且被正确预测，FP 表示实际为 negative 类但被错误预测，并且根据这些值定义出了如下表达式：
 $accuracy=(TP+TN)/(P+N)$,表示模型的准确率； $TPR=TP/(TP+FN)$,称作灵敏度或者召回率； $FPR=FP/(FP+TN)$,也叫假证率； $Specificity=1-FPR$; $Precision=TP/(TP+FP)$,称为命中率，表示被分到正类的例子中哪些是正确的。

(2) ROC 曲线、gain charts 与 lift charts

横轴表示 FPR，纵轴表示 TPR，ROC 曲线下方的面积越大，分类器的性能越好；收益曲线横轴为 RPP，纵轴为 TPR；LIFT 曲线横轴不同的是纵轴为 TPR/RPP。通过作图可以更直观的显示出分类器的性能评价结果。

2.4 实验过程及结果分析

1. 数据的读入与分析

(1) 首先是数据的读入（见图 2-1），需要将数据的路径修改为电脑中保存的路径名。

```
> BloodData<-read.csv("C:/Users/DELL/Documents/Tencent Files/1293604741/FileRecv/输血服务中心数据集.csv", head = TRUE, encoding = 'utf-8')
> BloodData
  Recency Frequency Monetary Time DonatedBloodInMarch2007
1      2         50      12500   98                    1
2      0         13       3250   28                    1
3      1         16       4000   35                    1
4      2         20       5000   45                    1
5      1         24       6000   77                    0
6      4          4       1000    4                    0
7      2          7       1750   14                    1
8      1         12       3000   35                    0
9      2          9       2250   22                    1
10     5         46      11500   98                    1
11     4         23       5750   58                    0
12     0          3        750    4                    0
13     2         10       2500   28                    1
14     1         13       3250   47                    0
15     2          6       1500   15                    1
16     2          5       1250   11                    1
```

图 2-1

(2) 图 2-2 显示了数据的结构与初步的分析结果，可以得出 2007 年 3 月献血与不献血的具体人数。

```
> table(BloodData$DonatedBloodInMarch2007)

 0    1 
570 178
```

图 2-2

(3) 图 2-3 反应了属性之间的的关系，得出不同的捐赠次数下，2007 年 3 月献血与不献血的具体人数，图 2-4 通过图形更加直观的显示出两者之间的关系。

```
> table(BloodData[,c("Frequency","DonatedBloodInMarch2007")])
      DonatedBloodInMarch2007
Frequency    0    1
1      138   20
2       93   19
3       73   14
4       49   13
5       42   20
6       35   17
7       31   12
8       18   13
9       18    6
10        8    6
11       16    6
12       11    3
13        4    5
14        9    4
15        5    1
16        9    4
17        2    2
18        1    0
19        1    1
20        0    2
21        0    2
22        1    1
```

图 2-3

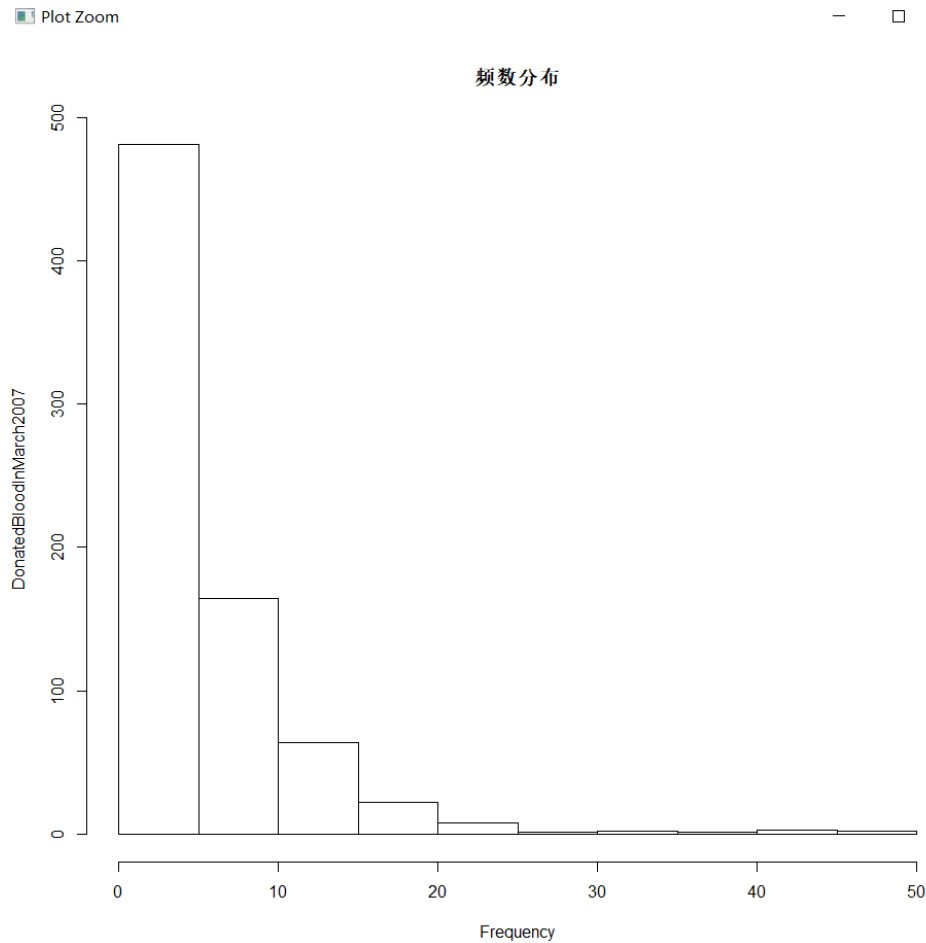


图 2-4

(4) 查看数据的结构，图 2-5 显示了数据的每一个属性的具体取值。

```
> str(BloodData)
'data.frame': 748 obs. of 5 variables:
 $ Recency : int 2 0 1 2 1 4 2 1 2 5 ...
 $ Frequency : int 50 13 16 20 24 4 7 12 9 46 ...
 $ Monetary : int 12500 3250 4000 5000 6000 1000 1750 3000 2250 11500 ...
 $ Time : int 98 28 35 45 77 4 14 35 22 98 ...
 $ DonatedBloodInMarch2007: int 1 1 1 1 0 0 1 0 1 1 ...
```

图 2-5

2. 决策树的建立

(1) 将数据分为训练集与测试集，见图 2-6，训练集数据占 70%，图 2-7 和 2-8 分别罗列出训练集数据和测试集数据以及两者的记录数

[illegible]

图 2-6

```

> TrainData = BloodData[index == 1, ]
> TrainData
  Recency Frequency Monetary Time DonatedBloodInMarch2007
1       2         50    12500   98                      1
2       0         13     3250   28                      1
3       1         16     4000   35                      1
4       2         20     5000   45                      1
6       4          4     1000    4                      0
7       2          7     1750   14                      1
8       1         12     3000   35                      0
9       2          9     2250   22                      1
10      5         46    11500   98                      1
11      4         23     5750   58                      0
12      0          3       750    4                      0
13      2         10     2500   28                      1
15      2          6     1500   15                      1
17      2         14     3500   48                      1
18      2         15     3750   49                      1
19      2          6     1500   15                      1
20      2          3       750    4                      1
21      2          3       750    4                      1

```

图 2-7

```

> nrow(TrainData)
[1] 519
> TestData = BloodData[index == 2,]
> TestData
  Recency Frequency Monetary Time DonatedBloodInMarch2007
5        1         24     6000   77                      0
14       1         13     3250   47                      0
16       2          5     1250   11                      1
26       4         14     3500   40                      0
28       4         12     3000   34                      1
29       4          5     1250   11                      1
36       2          8     2000   28                      1
39       2         14     3500   57                      1
40       4          7     1750   22                      1
50       2          2       500    2                      0
53       2          6     1500   22                      0
58       2          7     1750   28                      1
60       3          6     1500   21                      0

> nrow(TestData)
[1] 229

```

图 2-8

(2) 建立分类回归树的 R 函数是 rpart 包中的 rpart,给出分类模型, 见图 2-

9。

```

> rpart.model
n= 519

node), split, n, loss, yval, (yprob)
* denotes terminal node

1) root 519 131 0 (0.7475915 0.2524085)
 2) Recency>=8.5 249 30 0 (0.8795181 0.1204819) *
 3) Recency< 8.5 270 101 0 (0.6259259 0.3740741)
   6) Frequency< 3.5 111 29 0 (0.7387387 0.2612613) *
   7) Frequency>=3.5 159 72 0 (0.5471698 0.4528302)
    14) Time>=50 59 16 0 (0.7288136 0.2711864)
       28) Frequency< 18 48 8 0 (0.8333333 0.1666667) *
       29) Frequency>=18 11 3 1 (0.2727273 0.7272727) *
    15) Time< 50 100 44 1 (0.4400000 0.5600000)
       30) Frequency< 7.5 64 31 0 (0.5156250 0.4843750)
          60) Time>=23.5 37 13 0 (0.6486486 0.3513514) *
          61) Time< 23.5 27 9 1 (0.3333333 0.6666667) *
          31) Frequency>=7.5 36 11 1 (0.3055556 0.6944444) *

```

图 2-9

- (3) 为更形象直观的展示决策树，还需下载安装 rpart.plot 包，实现决策树的可视化，画出决策树，见图 2-10

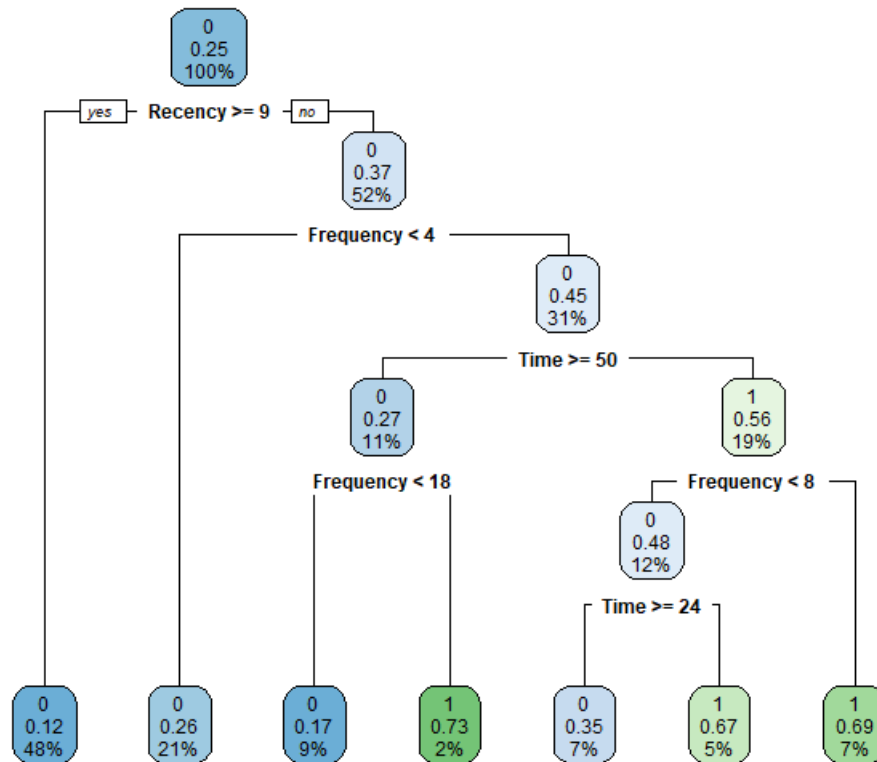


图 2-10

- (4) 对分类的概率进行预测，图 2-11 判断出了每一条记录被分为“0”类和“1”类的概率是多少。

```

> predict(rpart.model)
      0      1
1 0.2727273 0.7272727
2 0.3055556 0.6944444
3 0.3055556 0.6944444
4 0.3055556 0.6944444
6 0.3333333 0.6666667
7 0.3333333 0.6666667
8 0.3055556 0.6944444
9 0.3055556 0.6944444
10 0.2727273 0.7272727
11 0.2727273 0.7272727
12 0.7387387 0.2612613
13 0.3055556 0.6944444
15 0.3333333 0.6666667
17 0.3055556 0.6944444
18 0.3055556 0.6944444
19 0.3333333 0.6666667
20 0.7387387 0.2612613
  
```

图 2-11

(5) 对分类结果进行预测，由上一步的概率值可知，每一条记录的分类结果为概率值较大的那一类，图 2-12 显示出了最终的结果。

[illegible]

图 2-12

(6) 将预测结果与实际结果进行比较, 见图 2-13 最后两列的数值。

```
> train.predict=cbind(TrainData,train_predict) #把两个结果合起来看
> train.predict
```

	Recency	Frequency	Monetary	Time	DonatedBloodInMarch2007	train_predict
1	2	50	12500	98	1	1
2	0	13	3250	28	1	1
3	1	16	4000	35	1	1
4	2	20	5000	45	1	1
6	4	4	1000	4	0	1
7	2	7	1750	14	1	1
8	1	12	3000	35	0	1
9	2	9	2250	22	1	1
10	5	46	11500	98	1	1

图 2-13

(7) 在比较的基础上可以判断出被正确预测和被错误预测的记录个数，由此产生混淆矩阵，见图 2-14

		Actual	
Predicted	0	1	
0	365	80	
1	23	51	

图 2-14

(8) 计算训练集的错误率大致是 19.8% (见图 2-15)

```
> (Error.train=(sum(train_confusion)-sum(diag(train_confusion)))/sum(train_confusion))
[1] 0.1984586
```

图 2-15

(9) 对测试集进行预测，可以清晰地看出每一个记录属于 0 类还是 1 类（见图 2-16）。

```

> test_predict=predict(rpart.model,newdata = TestData,type="class")
> test_predict
 5  14  16  26  28  29  36  39  40  50  53  58  60  61  66  72  74  81
1  1  1  1  1  1  1  1  0  1  0  1  0  1  0  0  1  0  1
149 154 156 158 169 171 173 176 185 187 190 192 194 195 196 197 198 199
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
272 274 277 283 284 285 290 293 296 302 305 308 310 311 320 322 330 331
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
389 390 392 393 396 403 404 407 411 412 413 415 418 422 425 432 434 438
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
502 503 506 507 508 510 512 513 516 521 524 526 534 536 539 543 545 547
1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  1
612 615 619 621 624 626 628 631 635 636 638 644 648 650 653 657 660 662
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
726 730 731 736 738 739 740
0  0  0  0  0  0  0
Levels: 0 1

```

图 2-16

图 2-17 显示了测试集的混淆矩阵

```

          Actual
Predicted  0   1
          0 165  31
          1  17  16

```

图 2-17

计算测试集的错误率，见图 2-18

```

> (Error.rpart=(sum(test_confusion)-sum(diag(test_confusion)))/sum(test_confusion))
[1] 0.209607

```

图 2-18

图 2-19 呈现了测试集和训练集的一些参数，灵敏度与真负率。

```

> sensitivity(train_predict,TrainData$DonatedBloodInMarch2007)
[1] 0.9407216
> specificity(train_predict,TrainData$DonatedBloodInMarch2007)
[1] 0.389313
> sensitivity(test_predict,TestData$DonatedBloodInMarch2007)
[1] 0.9065934
> specificity(test_predict,TestData$DonatedBloodInMarch2007)
[1] 0.3404255

```

图 2-19

3. 决策树的剪枝

(1) 复杂度参数 CP 是决策树剪枝的关键参数。需要进一步了解 CP 对模型

预测误差的影响，并以此判断用户指定的初始 CP 值是否合理，可通过函数

printcp 和 plotcp 浏览（图 2-20）与可视化 CP 值（图 2-21）。

```
> printcp(rpart.model)

Classification tree:
rpart(formula = DonatedBloodInMarch2007 ~ ., data = TrainData,
      method = "class", parms = list(split = "information"))

Variables actually used in tree construction:
[1] Frequency Recency Time

Root node error: 131/519 = 0.25241

n= 519

      CP nsplit rel error xerror  xstd
1 0.030534      0  1.00000 1.0000 0.075543
2 0.010000      6  0.78626 1.0076 0.075733
```

图 2-20

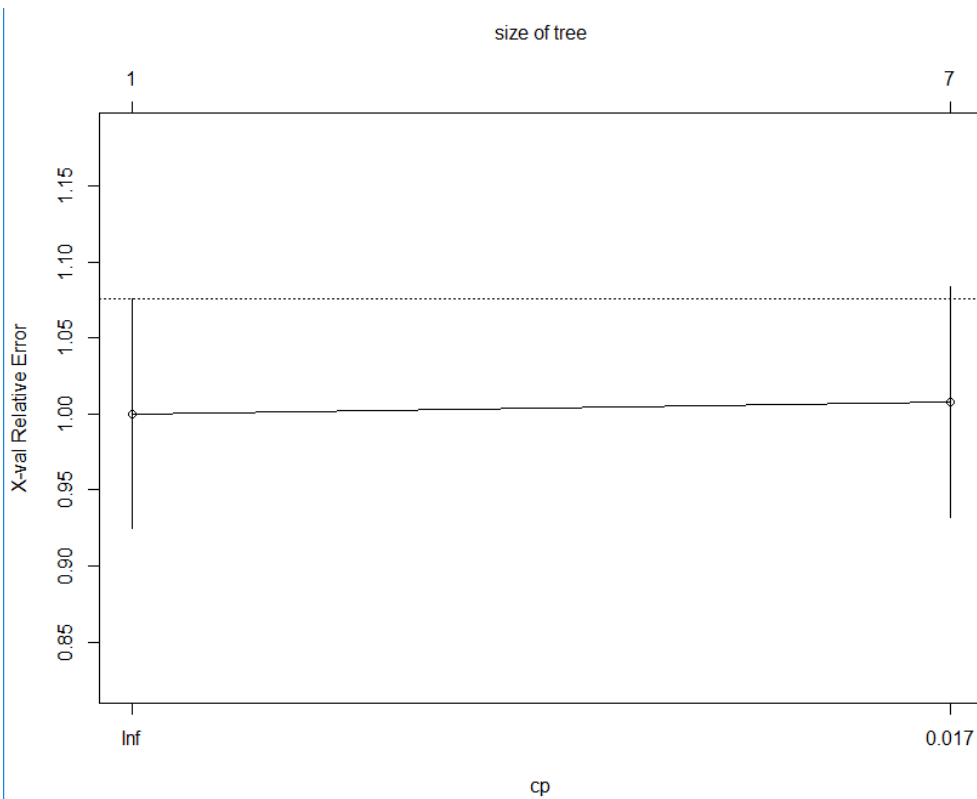


图 2-21

- (2) 设置随机数种子使剪枝结果可以重现，图 2-22 显示了剪枝后的决策树。

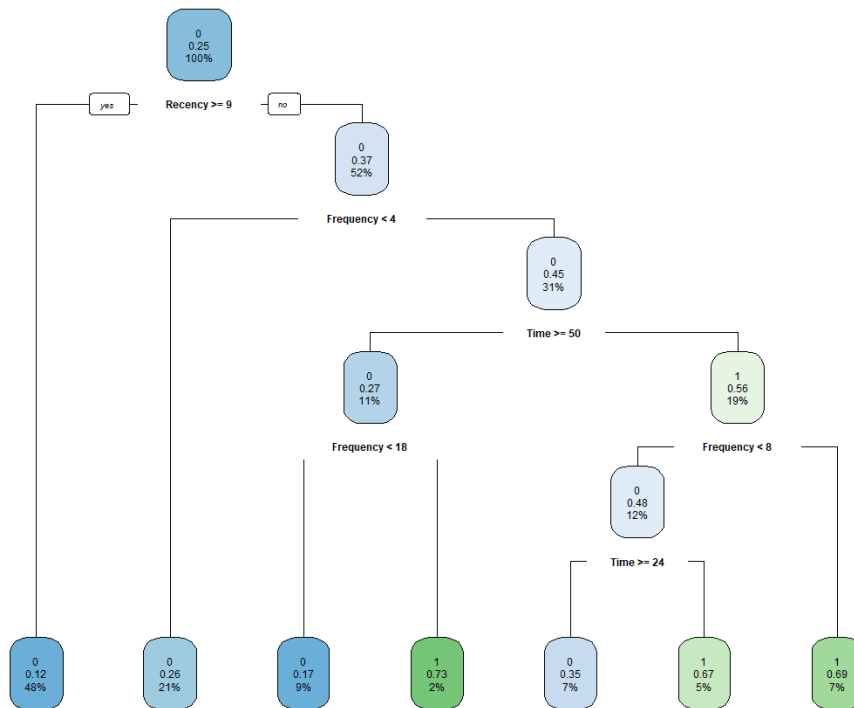


图 2-22

(3) 用剪枝之后的模型，预测训练集数据的分类结果（见图 2-23）和测试集数据的分类结果（见图 2-24）

```
> train.predict
1  2  3  4  6  7  8  9 10 11
1  1  1  1  1  1  1  1  1  1
32 33 34 35 37 38 41 42 43 44
1  1  1  0  1  1  0  1  1  1
65 67 68 69 70 71 73 75 76 77
0  1  0  0  0  1  0  0  0  1
97 98 99 101 102 103 104 105 106 107
0  1  1  1  0  1  0  0  0  0
133 134 136 138 139 141 143 144 145 146
0  0  0  0  0  0  0  0  0  0
167 168 170 172 174 175 177 178 179 180
0  0  0  0  0  0  0  0  0  0
208 209 211 212 213 215 217 219 221 222
0  0  0  0  0  0  0  0  0  0
```

图 2-23

```
> test.predict
5  14  16  26  28  29  36  39  40
1  1  1  1  1  1  1  0  1
117 120 121 122 123 124 131 135 137
0  0  0  0  0  0  0  0  0
195 196 197 198 199 204 206 210 214
0  0  0  0  0  0  0  0  0
274 277 283 284 285 290 293 296 302
0  0  0  0  0  0  0  0  0
355 356 358 360 365 366 369 370 371
0  0  0  0  0  0  0  0  0
425 432 434 438 439 441 442 443 445
0  0  0  0  0  0  0  0  0
506 507 508 510 512 513 516 521 524
1  1  1  1  1  1  1  1  1
```

图 2-24

(4) 计算剪枝之后测试集的混淆矩阵以及测试集的错误率（见图 2-25）

```
> (test_confusion=table(actual=TestData$DonatedBloodInMarch2007,predictedclass=test_predict))
      predictedclass
actual 0 1
      0 165 17
      1 31 16
> (Error.rpart=(sum(test_confusion)-sum(diag(test_confusion)))/sum(test_confusion))
[1] 0.209607
```

图 2-25

生成剪枝之后训练集和测试集的灵敏度与真负率（见图 2-26）

```
> sensitivity(train.predict,TrainData$DonatedBloodInMarch2007)
[1] 0.9407216
> specificity(train.predict,TrainData$DonatedBloodInMarch2007)
[1] 0.389313
> sensitivity(test.predict,TestData$DonatedBloodInMarch2007)
[1] 0.9065934
> specificity(test.predict,TestData$DonatedBloodInMarch2007)
[1] 0.3404255
```

图 2-26

(5) 然后我们就可以利用这个模型对新的数据进行预测，图 2-27 显示了对测试集数据的预测结果。

```
> predict(rpart.prune, TestData)
      0      1
5    0.2727273 0.7272727
14   0.3055556 0.6944444
16   0.3333333 0.6666667
26   0.3055556 0.6944444
28   0.3055556 0.6944444
29   0.3333333 0.6666667
36   0.3055556 0.6944444
39   0.8333333 0.1666667
40   0.3333333 0.6666667
50   0.7387387 0.2612613
53   0.3333333 0.6666667
58   0.6486486 0.3513514
60   0.3333333 0.6666667
```

图 2-27

(6) 分类器性能评价之 ROC 曲线的绘制

- a) 计算 TPR 和 FPR，ROC 包中完成相关计算的函数是 prediction 和 performance, prediction 函数的主要目的是将概率值和类别值组织成 performance 函数要求的对象格式
- b) 画图，画图函数为 plot,图 2-28 中蓝色表示训练集，红色表示测试集，还可以对曲线的线型与宽度进行设定，再加上一条对角线表示随机模

型的预测效果并以此作为一个参考值，若 ROC 曲线位于其下，则该模型便可以舍弃。

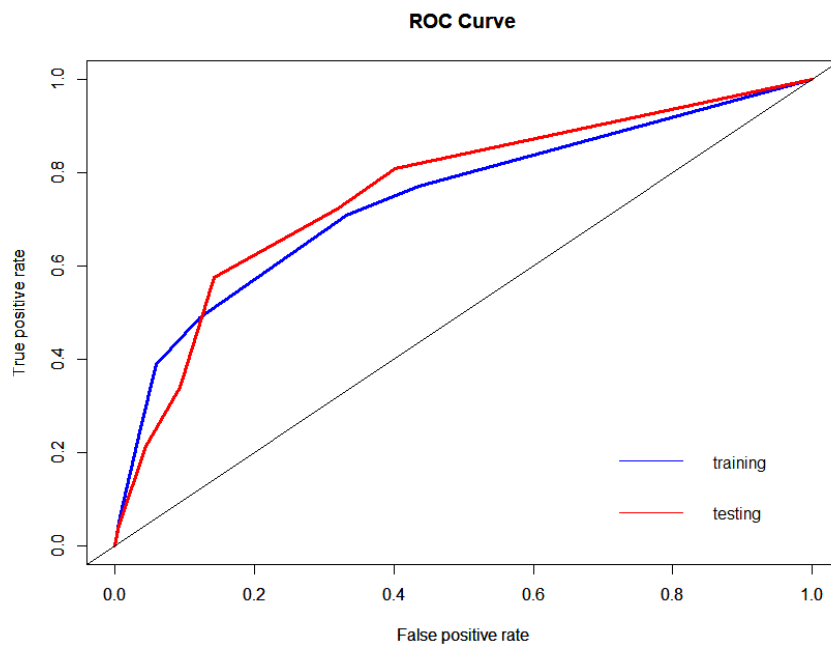


图 2-28

1. 分类器性能评价之 gain chart 的绘制

利用 performance 函数设定曲线的纵坐标为 TPR，横坐标为 RPP，再利用 plot 函数画图，图 2-29 中蓝色为训练集，红色为测试集，同样的绘制对角线表示随机模型。

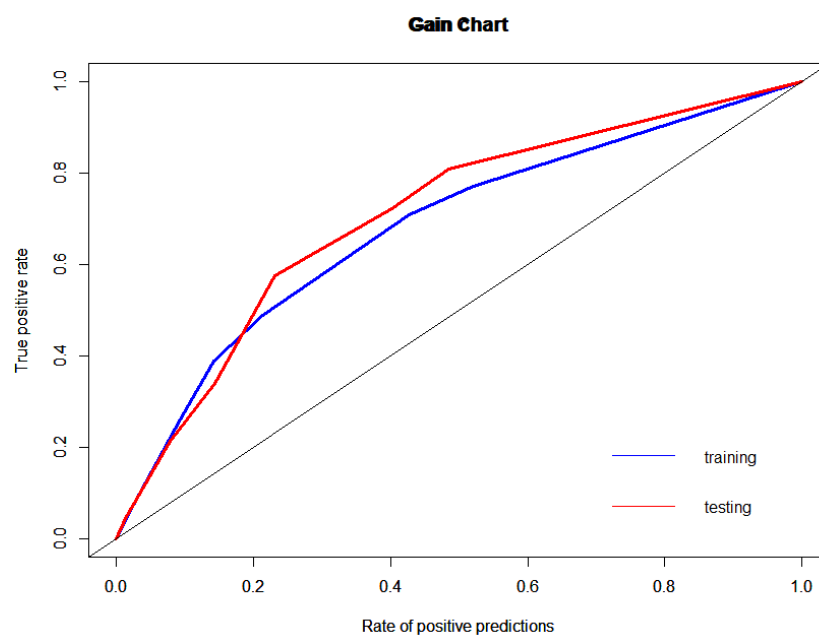


图 2-29

2. 分类器性能评价之 lift chart 的绘制

利用 performance 函数设定曲线的纵坐标为 LIFT，横坐标为 RPP，再利用 plot 函数画图，图 2-30 中蓝色为训练集，红色为测试集。

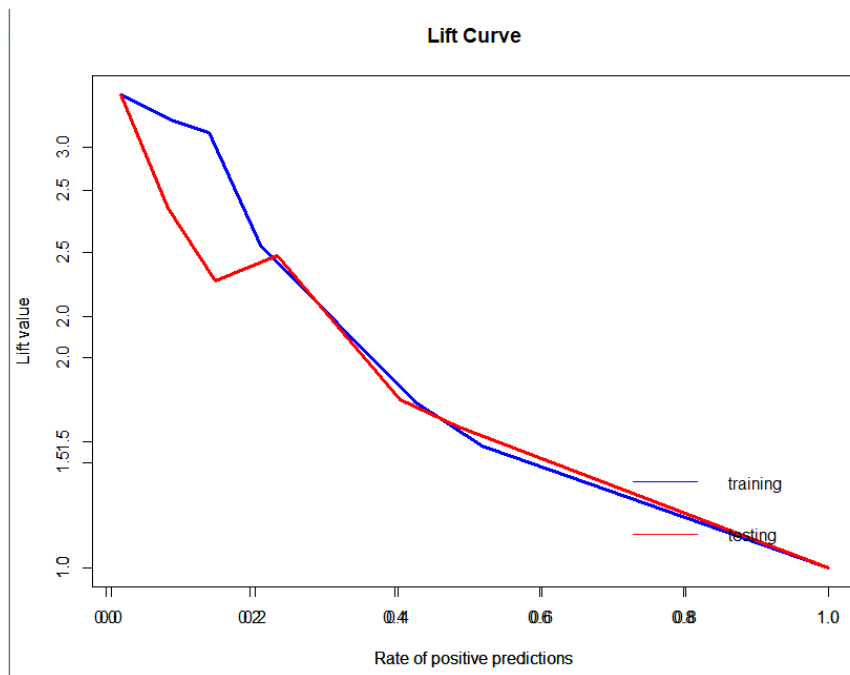


图 2-30

三、 基于规则组合预测

3.1 实验目的

1. 学习使用集成的方法进行预测，也就是分类回归树的组合预测模型，主要是掌握其中的袋装技术、推进技术与随机森林三种集成方法。
2. 提高分类回归树预测的稳健性，并实现在训练样本中存在大量噪声数据的情况下提高模型的预测精度。

3.2 数据来源与说明

1. 数据来源：UCI ， <http://archive.ics.uci.edu/ml/datasets/Iris>
2. 数据名称：台湾新竹市的输血服务中心输血数据，该数据集共有 748 个实例，包含 5 个属性，通过对以上数据属性的分析预测出在 2007 年的 3 月是否会再次献血。（见附件一）
3. 数据属性与说明：
 - 1) 新进度-距上一次捐赠的时间跨度（Recency）
 - 2) 频率-捐赠总数（Frequency）
 - 3) 总血液捐赠量（Monetary）
 - 4) 时间-距第一次捐赠的时间跨度（Time）
 - 5) 二进制变量（0 表示不献血，1 表示献血）

3.3 算法描述

1. Adaboosting:
 - (1) 建模阶段：AdaBoost 技术通过对加权样本的有放回随机抽样，获得训练样本集。
 - 第一次建模时，对样本量为 n 的原始样本集 S ，进行有放回的随机抽样，此时得到的随机样本中的观测含有相同的权重，每个观测进入训练样本集的概率是相等的，在此基础上建立模型，之后重新调整各个观测的权重，对模型正确预测的样本给予较低的权重，错误预测的观测权重不变。
 - 第二次建模时，权重较大的观测进入样本集的概率较大，重点关注的是第一次建模未被正确分类的样本
 - 将以上过程重复 k 次，会得到 k 个模型。每次建模时权重的调整方式如

下：第一次建模时，各个观测的权重 $w_j(i) = 1/n$ ， $w_j(i)$ 表示第 j 个观测在第 i 次迭代中的权重。之后对于第 i 次迭代过程：根据样本的权重有放回的随机抽取观测形成训练样本集，建立模型并计算预测误差；根据 $e(i)$ 更新权重：正确预测的观测权重调整为 $w_j(i+1) = w_j(i) \times \beta(i)$ ，其中 $\beta(i)$ 为 $e(i)/(1-e(i))$ ，最后对权重进行归一化处理，使得各个观测的权重值和为 1。

(2) 预测阶段

对于每一个分类模型权重为： $\alpha = \frac{1}{2} \ln \frac{1-e(i)}{e(i)}$ ，对于一个新的观测，依照预测类别分别计算权重的总和，权重总和最高的类别即为该观测的最终预测类别。

2. 随机森林

随机森林是用随机方式建立一片森林，森林中包含众多有较高预测精度且弱相关甚至不相关的决策树，并形成组合预测模型，后续众多的预测模型将共同参与对新观测输出变量取值的预测。

(1) 构建随机森林的样本随机性

训练样本是对原始样本的重抽样自举，训练样本具有随机性，使得每次参与建模的训练样本存在一定的随机性差异，所获得决策树模型也因随机性而具有一定差异。

(2) 构建随机森林的变量随机性

对于随机森林，在第 i 颗决策树建立过程中，首先通过随机方式选取少量几个输入变量构成候选变量子集，只有进入变量子集的输入变量才有机会通过“竞争获胜”成为最佳分组变量，通常选择 k 个输入变量， $k = \sqrt{P}$ ， P 表示输入变量的个数，这样根据变量子集建立的充分生长的决策树，无需剪枝以减少预测偏差。

(3) 随机森林对输入变量重要性的测度

基本思路：若某输入变量对输出变量的预测有重要作用，对袋外观测 OOB 的该输入变量的取值上添加随机噪声，将显著影响输出变量的预测结果。R 中添加随机噪声的方法是随机打乱袋外观测 OOB 在该输入变量上的取值顺序。

(4) 随机森林模型评价

优点：仅需要两个简单的参数，决策树的棵树以及输入变量候选子集 k ；它是一项性能最好的通用分类器，计算效率高；能够处理数以万计的输入变量而不用删除数据。

3.4 实验过程及结果分析

1. AdaBoosting

实现推进技术的 R 函数是 adabag 包中的 boosting 函数。首次使用时应下载安装 adabag 包。

- (1) 首先进行数据的读入与 boosting 函数的参数设置，boosting 函数的基础学习器为分类树，control 参数应为 rpart 函数的参数（见图 3-1）。

```
ctrl<-rpart.control(minisplit=10,maxcompete=4, maxdepth=30,cp=0.01,xval=10)
set.seed(1234)
BloodData$DonatedBloodInMarch2007<-as.factor(BloodData$DonatedBloodInMarch2007)
```

图 3-1

- (2) 设置种子，按照 boosting 函数的基本书写格式设置函数，并采用 boosting 函数进行预测（见图 3-2）

pre.boost	List of 6
values	
bag.pred	Factor w/ 2 levels "0","1": 2 2 2 2 2 1 2 2 2 2 ...
confusion.bag	'table' int [1:2, 1:2] 536 34 98 80
confusion.one	'table' int [1:2, 1:2] 540 108 30 70
err1	0.184491978609626
err2	0.176470588235294
pred	Factor w/ 2 levels "0","1": 2 2 2 2 1 1 2 2 2 2 ...

图 3-2

- (3) 根据预测结果，计算混淆矩阵（见图 3-3）与错误率（见图 3-4）

```
> (confusion.boost<-pre.boost$confusion) #混淆矩阵存放在一个叫confusion 的子对象中
      Observed Class
Predicted Class  0    1
               0 546 105
               1  24  73
```

图 3-3

```
> (error.boost<-(sum(confusion.boost)-sum(diag(confusion.boost)))/sum(confusion.boost))
[1] 0.1724599
> |
```

图 3-4

- (4) 进行变量重要性的测度，并用图形显示（见图 3-5）。

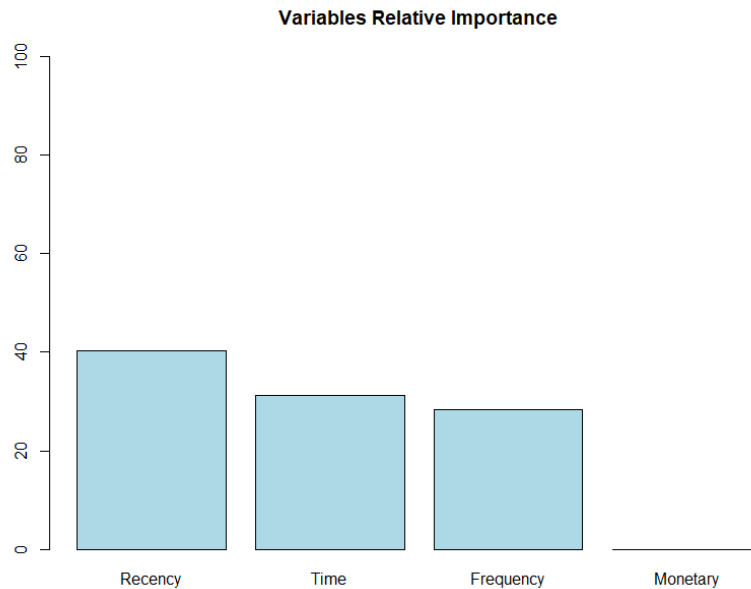


图 3-5

(5) 对模型进行十折交叉验证（见图 3-6）并计算混淆矩阵（见图 3-7）

```
> boostcvModel <- boosting.cv(DonatedBlood:
= ctrl)
i: 1 Tue Nov 06 20:05:47 2018
i: 2 Tue Nov 06 20:05:51 2018
i: 3 Tue Nov 06 20:05:59 2018
i: 4 Tue Nov 06 20:05:59 2018
i: 5 Tue Nov 06 20:06:03 2018
i: 6 Tue Nov 06 20:06:07 2018
i: 7 Tue Nov 06 20:06:11 2018
i: 8 Tue Nov 06 20:06:15 2018
i: 9 Tue Nov 06 20:06:19 2018
i: 10 Tue Nov 06 20:06:23 2018
```

图 3-6

```
> boostcvModel$confusion
      Observed Class
Predicted Class  0   1
      0  513 121
      1   57  57
```

图 3-7

2. 随机森林

建立随机森林的 R 函数是 randomForest 包中的 randomForest 函数，首先将其加载到 R 的工作空间中。

(1) 首先进行 random Forest 函数的参数设置，该函数的返回值为列表，包含一下成分：predicted、confusion、votes、oob . times、err. rate、importance 等等，在这里我们选择生成 votes(见图 3-8)，它适用于分类树，给出各预测类别的概率值，即随机森林中有多少比例的分类树投票给第 i 个

类别；`oob.times`（见图 9），它表示各个观测作为袋外观侧的次数，即在重抽样自举中有多少次未进入自举样本，它会影响基于袋外观测的预测误差结果。

```
> head(randomforesModel$votes) #predicted probability list of observations
      0      1
1 0.4869110 0.5130890
2 0.6666667 0.3333333
3 0.4486486 0.5513514
4 0.0500000 0.9500000
5 0.5089820 0.4910180
6 0.1797753 0.8202247
```

图 3-8

```
> head(randomforesModel$oob.times)#各观测作为oob的次数
[1] 191 183 185 180 167 178
```

图 3-9

- (2) 将随机森林模型的错判率与决策树的颗数之间的关系用图形直观的展示出来（见图 3-10），其中黑色线表示整体错判率，红色线表示对 NO 类的错判率，绿色线表示对 YES 类的错判率，可以发现模型对 YES 类的错判率高于整体与 NO 类。

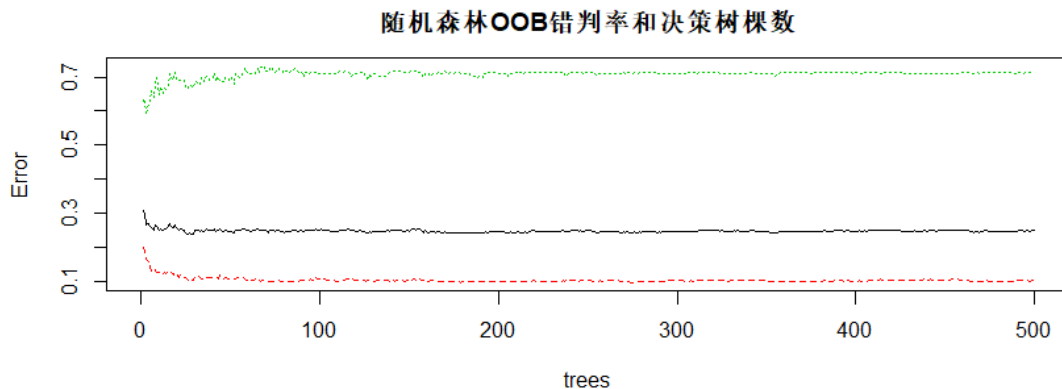


图 3-10

- (3) 将边界附近的点与错判率两者的关系用图像表示出来（见图 3-11），考察是否处于边界的原则是：计算投票给正确类别（该观测所属的实际类别）与投票给众数类（出正确类别以外的其他众数类别）的比率之差，之差为正表示预测正确，为负表示预测错误，差的绝对值越小，越接近于零，表明该观测处在分类边界上。

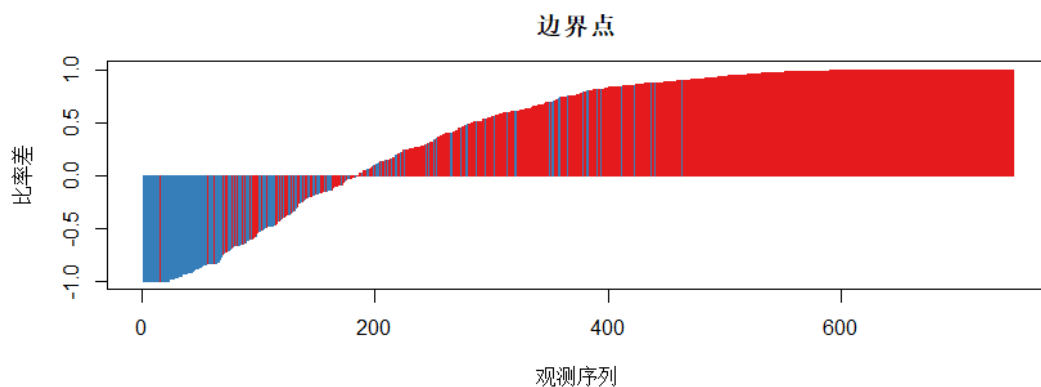


图 3-11

(4) 浏览各个树的叶节点个数 (见图 3-12)

```
> head(treesize(randomforesModel))#浏览各个树的叶节点个数
[1] 106 124 117 128 133 125
> |
```

图 3-12

(5) 用随机森林模型进行预测，并计算出相应的混淆矩阵 (见图 3-13) 与错判率 (见图 3-14)

```
> (confusion.random<-table(BloodData$DonatedBloodInMarch2007,pre.random))
pre.random
  0  1
0 561  9
1  57 121
~ |
```

图 3-13

```
> (error.random<-(sum(confusion.random)-sum(diag(confusion.random)))/sum(confusion.random))#计算错误率
[1] 0.08823529
> |
```

图 3-14

(6) 进行变量重要性的判别，有两种方法：1 表示使用精度平均减少值作为度量标准；2 表示采用节点不纯度的平均减少值作为度量标准，值越大说明变量的重要性越强，这里采用第一种方法 (见图 15)，可以看到 Time 这一变量对输出结果影响最大。

```
> importance(randomforesModel)# type可以是1，也可以是2， 计算gini系数
      MeanDecreaseGini
Recency             53.57788
Frequency           31.09616
Monetary            31.14587
Time                74.23479
```

图 3-15

(7) 将输入变量的重要性用柱形图更加直观的表现出来 (见图 16)，重要性是根据预测精度来判定的，以及变量重要性的散点图 (见图 17)

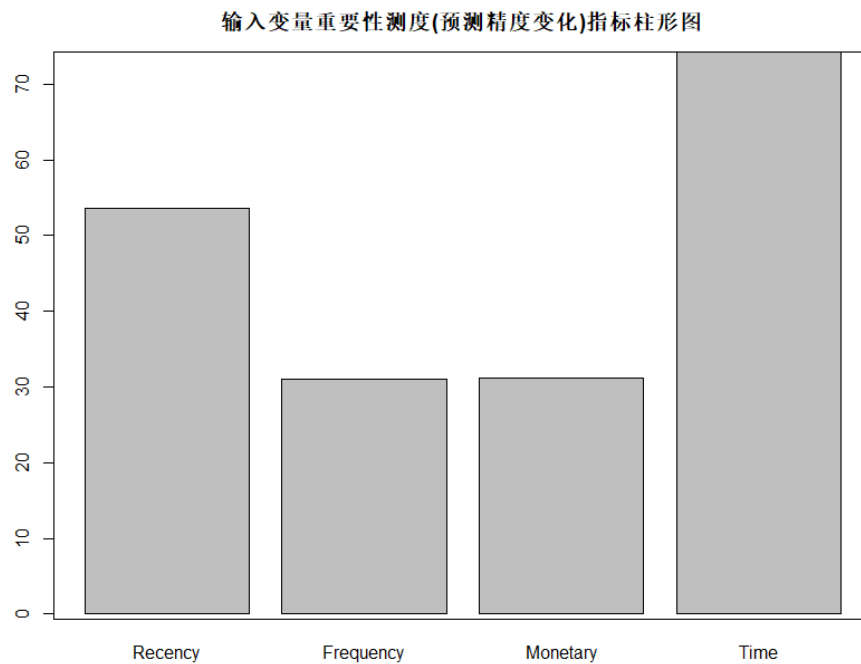


图 3-16

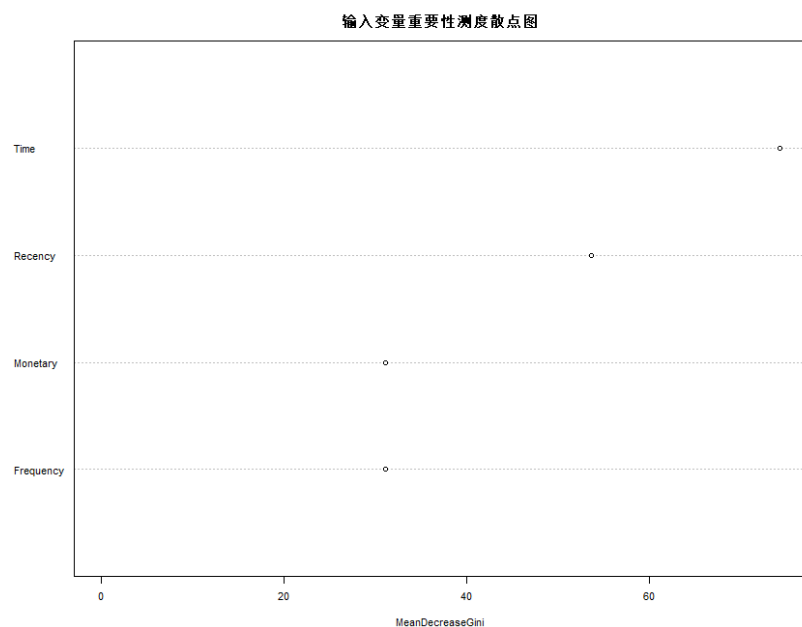


图 3-17

(8) 最后生成 randomForest 函数的 importance 返回值列表，可以观察到它是一个矩阵（见图 18）。

```
> randomforesModel$importance
      MeanDecreaseGini
Recency      53.57788
Frequency    31.09616
Monetary     31.14587
Time         74.23479
> is.matrix(randomforesModel$importance)
[1] TRUE
```

图 3-18

- (9) 寻找最优的数据变量的个数，设置数据变量从 1 到 $n-1$ (n 为变量的总个数)，计算出每次的错误率，以此为依据进行选择（见图 19）

```
> n=length(names(BloodData))
> set.seed(100)
> for(i in 1:(n-1)){mtryFit<-randomForest(DonatedBloodInMarch2007~.,data=BloodData,mtry=i)
+ err=mean(mtryFit$err.rate)
+ print(err)}
[1] 0.3368465
[1] 0.3563012
[1] 0.360929
[1] 0.371712
> |
```

图 3-19

3. 三种方法分类性能的比较

分别生成三种集成模型下的 ROC 曲线，观察三种方法的分类性能。利用 performance 函数计算纵坐标 TPR 与横坐标 FPR，最终将三条曲线呈现在一张图中以便于进行对比（见图 20）。其中，蓝色表示袋装技术，红色表示推进技术，绿色表示随机森林方法，黑色表示随机模型。对比可以发现，模型性能从高到低依次为：随机森林、推进技术、袋装技术。

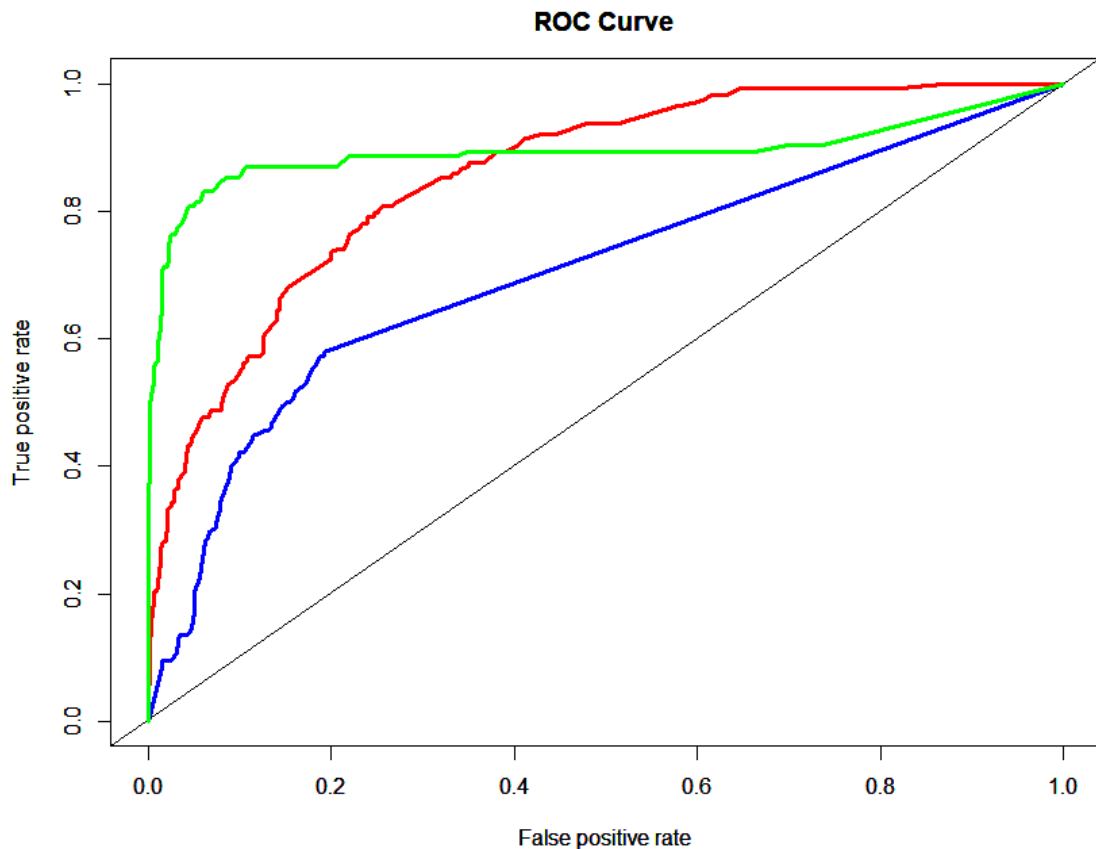


图 3-20

四、 懒惰学习——基于近邻的分类预测

4.1 实验目的

1. 理解 K-近邻法方法的原理和适用性，掌握基于变量重要性的加权 K-近邻法以及基于观测相似性的加权 K-近邻法算法的原理和使用特点；
2. 能够掌握 R 近邻分析的函数、应用和结果解读，基本熟练运用 KNN 算法来分析数据，锻炼分析问题、解决问题并动手实践的能力。

4.2 数据来源与说明

1. 数据来源：UCI ， <http://archive.ics.uci.edu/ml/datasets/Iris>
2. 数据名称：Iris Data Set 鸢尾植物数据库

这可能是模式识别文献中最著名的数据库。费希尔的论文是该领域的经典之作，至今仍被频繁引用。数据集包含 3 个类别，每个类别 50 个实例，其中每个类别指的是一种鸢尾植物。（见附件二）

3. 数据属性与说明：
 - （1） SepalLength：萼片长度（单位：cm）
 - （2） SepalWidth：萼片宽度（单位：cm）
 - （3） PetalLength：花瓣长度（单位：cm）
 - （4） PetalWidth：花瓣宽度（单位：cm）
 - （5） IrisPlantClass：鸢尾植物分类
 - Iris Setosa 山鸢尾
 - Iris Versicolour 杂色鸢尾
 - Iris Virginica 维吉尼亚鸢尾

4.3 算法描述

1. K-近邻法

K 最近邻(k-Nearest Neighbor, KNN)分类算法, 是一个理论上比较成熟的方法, 也是最简单的机器学习算法之一。该方法的思路是: 如果一个样本在特征空间中的 k 个最相似(即特征空间中最邻近)的样本中的大多数属于某一个类别, 则该样本也属于这个类别。K 值的选择, 距离度量和分类决策规则是该算法的三个基本要素:

(1) K 值的选择

K 值较小意味着只有与输入实例较近的训练实例才会对预测结果起作用, 但容易发生拟合; 如果 K 值较大, 优点是可以减少学习的估计误差, 但缺点是学习的近似误差增大, 这时与输入实例较远的训练实例也会对预测起作用, 使预测发生错误。

在实际应用中, K 值一般选择一个较小的数值, 通常采用交叉验证的方法来选择最优的 K 值。随着训练实例数目趋向于无穷和 $K=1$ 时, 误差率不会超过贝叶斯误差率的 2 倍, 如果 K 也趋向于无穷, 则误差率趋向于贝叶斯误差率。

(2) 分类决策规则

该算法中的分类决策规则往往是多数表决, 即由输入实例的 K 个最临近的训练实例中的多数类决定输入实例的类别

(3) 距离度量

一般采用 L_p 距离, 当 $p=2$ 时, 即为欧氏距离。在度量之前, 应该将每个属性的值规范化(归一化), 这样有助于防止具有较大初始值域的属性比具有较小初始值域的属性的权重过大。

2. 基于变量重要性的加权 K-近邻法

由于我们计算 K-近邻法默认输入变量在距离测度中有“同等重要”的贡献, 但情况并不总是如此。不同的变量对我们所要预测的变量的作用是不一定一样的, 所以找出对输出变量分类预测有意义的重要变量对数据预测具有重要作用。同时也可以减少那些对输出变量分类预测无意义的输入变量, 减少模型的变量。为此, 采用基于变量重要性的 K-近邻法, 计算加权距离, 给重要的变量赋予较高的权重, 不重要的变量赋予较低的权重是必要的。

(1) 算法思路:

引进 w_i 为第 i 个输入变量的权重, 是输入变量重要性(也称特征重要性), FI 函数, 定义为: $w_i = FI_i / \sum FI_i$ 。其中 FI_i 为第 i 个输入变量的特征重要性, $w_i < 1, \sum w_i = 1$, 这里, FI_i 依第 i 个输入变量对预测误差的影响定义。

设输入变量集合包含 p 个变量: x_1, x_2, \dots, x_p 。剔除第 i 个变量后计算剩余输入变量的误判率, 记为 e_i 。若第 i 个变量对预测有重要作用, 剔除变量后的

预测误差 e_i 应较大。于是，第 i 个变量的重要性定义为： $FI_i = e_i + 1/p$ 。可见，变量越重要，在计算距离时的权重越高。

(2) 算法步骤：

- step. 1——求解出错判率最低的 K 值；
- step. 2——求解出第 i 个变量的 FI_i 。

3. 基于观测相似性的加权 K -近邻法

(1) 算法思路：

K -近邻法预测时，默认 K 个近邻对观测结果又“同等力度“的影响。事实上，据 x_0 的远近观测对预测贡献的大小是有影响的，距离越近对预测的贡献大于距离较远的预测贡献。

将相似性定义为各观测与 x_0 距离的某种非线性函数，且距离越近，相似性越强，权重越高，预测时的重要性越大。

设观测 x 与 x_0 的距离为 $d(d \geq 0, d \in \partial)$ 。若采用函数 $K(d)$ 将距离 d 转换成 x 与 x_0 的相似性，则函数 $K(d)$ 应有如下特性：

- $K(d) \geq 0, d \in \partial$
- $d=0$ 时， $K(d)$ 获得最大值，即距离最近时相似性越大
- $K(d)$ 是 d 的单调减函数，即距离越远相似性越小

通常，核函数是符合上述特征的函数。有均匀核函数、三角核函数和高斯核函数等。

(2) 算法步骤：

- step. 1——求解误判率最低的 k 值；
- step. 2——加权 K -近邻法与 K -近邻法比较。

4.4 实验过程及结果分析

1. K -近邻法

(1) 读取数据得到训练集和测试集（如图 4-1），由于数据比较少，就直接用的训练集做测试了。共 150 个实例，5 个变量类型；并将第 5 列的属性（IrisPlantClass）赋予类标号（见图 4-2）。



Data		
▶ Iris_test	150 obs. of 5 variables	
▶ Iris_train	150 obs. of 5 variables	

图 4-1

Values	
Iris_test.lab...	Factor w/ 3 levels "Iris-setosa",...: 1...
Iris_train.la...	Factor w/ 3 levels "Iris-setosa",...: 1...

图 4-2

- (2) 由于变量取值存在数量级差异，因此需要对输入的数据进行标准化，利用极差法进行归一化，可以得到图 4-3 中四个变量的最大值/最小值、分位数等。

```
> summary(Iris_train.n)
SepalLength      Sepalwidth      PetalLength      Petalwidth
Min.      :0.0000    Min.      :0.0000    Min.      :0.0000    Min.      :0.00000
1st Qu.:0.2222    1st Qu.:0.3333    1st Qu.:0.1017    1st Qu.:0.08333
Median :0.4167    Median :0.4167    Median :0.5678    Median :0.50000
Mean    :0.4287    Mean    :0.4406    Mean    :0.4675    Mean    :0.45806
3rd Qu.:0.5833    3rd Qu.:0.5417    3rd Qu.:0.6949    3rd Qu.:0.70833
Max.    :1.0000    Max.    :1.0000    Max.    :1.0000    Max.    :1.00000
```

图 4-3

- (3) 依据旁置法计算当 K 取 [1, 30]，不同取值下的预测误差（错判概率），如图 4-4 所示为错误率的值：

```
> errRatio
[1] 0.000000 1.333333 3.333333 4.000000 4.000000 4.000000 3.333333 2.666667 2.666667 2.666667 4.000000 5.333333
[13] 2.666667 3.333333 2.666667 2.666667 3.333333 3.333333 3.333333 4.000000 3.333333 4.000000 4.666667 4.000000
[25] 4.666667 4.000000 4.000000 4.000000 4.000000 4.000000
```

图 4-4

同时可以绘制样本集的错误率随近邻个数 K 变量的折线图（见图 4-5）：

鸢尾植物分类预测中的近邻数K与错判率

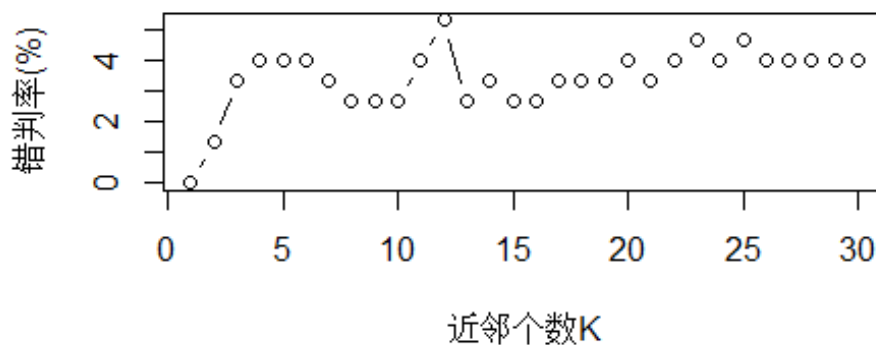


图 4-5

结合图 4-5 并兼顾近邻分析的稳健性等考虑，本例采用 K=7 的分析结论，测试样本集的错误率为 3.33%。

- (4) 调用 `kkn` 函数进行模型训练与预测，当 $K=7$ 时分类结果如图 4-6 所示，可以从下图看出，只有 6 个记录被分错。

```
> fit <- fitted(knnm(fit))
> table(Iris_test$IrisPlantClass, fit)
      fit
      Iris-setosa Iris-versicolor Iris-virginica
Iris-setosa      50              0              0
Iris-versicolor  0              47              3
Iris-virginica   0              3              47
```

图 4-6

2. 基于变量重要性的加权 K-近邻法

- (1) 利用普通 KNN 方法确定参数 K ，测试样本的错判率随 K 值变量的曲线如图 4-7 所示，可见， $K=7$ 时错判率较低为 3.33%且能保证预测稳定性。

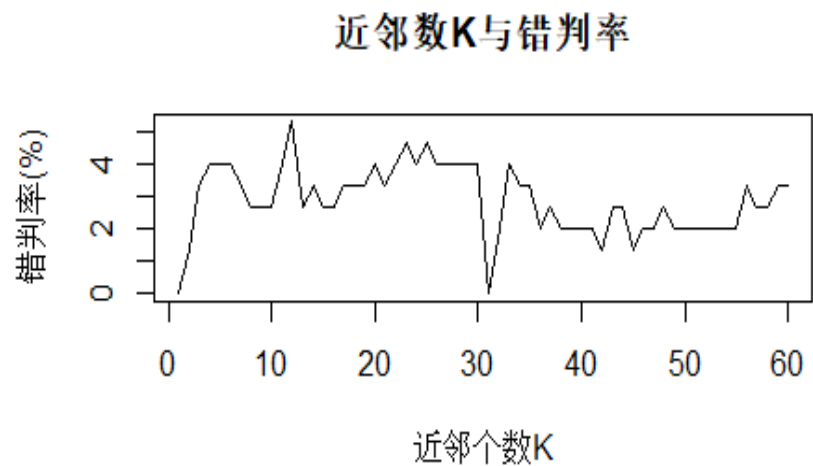


图 4-7

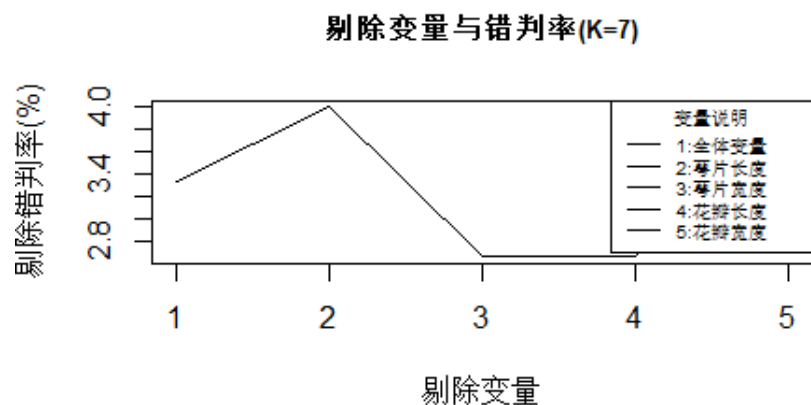


图 4-8

(2) 逐个剔除输入变量，剔除后的错判率曲线如图 4-8 所示，图中，横坐标为 1 是 4 个输入变量参与 K-近邻分析的错判率。

可以看出，剔除萼片长度和花瓣宽度后错判率明显增加，说明这两者对预测的影响比较大。

(3) 依据 FI 的定义计算各个输入变量的重要性，以此确定权重。计算得到四个变量权重为：0.2965116, 0.2034884, 0.2034884, 0.2965116。各个输入变量的权重分配如图 4-9 的饼图所示，四个变量权重相差不是特别大，萼片长度和花瓣宽度的权重较高。

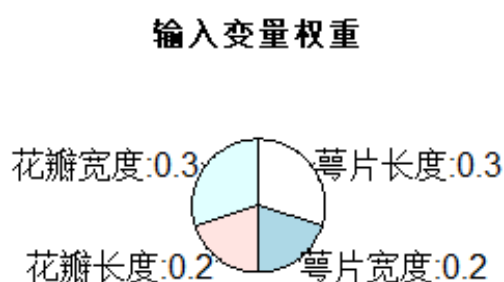


图 4-9

(4) 图 4-10 是萼片长度和花瓣宽度特征空间中观测点的分布情况，其中三种不同的颜色代表三种鸢尾植物的品种，可以看出，这两个特征的分类能力还是很强的。

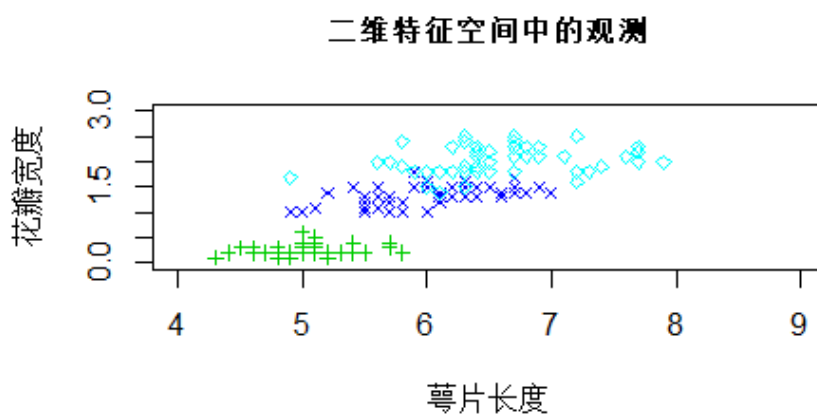


图 4-10

综上所述，在 K=7 时，普通 KNN 对测试集的预测错判率仅为 3.33%，效果较为理想。其次，大部分鸢尾植物 setosa 处于花瓣宽度和萼片长度较低的位置，可为今后判断鸢尾植物的种类提供依据。

3. 基于观测相似性的加权 K-近邻法

(1) 求解误判率最低的 k 值

调用 `kknn` 函数，对比三种核函数（均匀核函数、三角核函数和高斯核函数）下，近邻个数 K 取 $[1, 11]$ 时留一法的错判率曲线，如图 4-11 所示，图中黑色实线、黑色长虚线、红色短虚线分别代表均匀核函数、三角核函数和高斯核函数的错判率曲线。

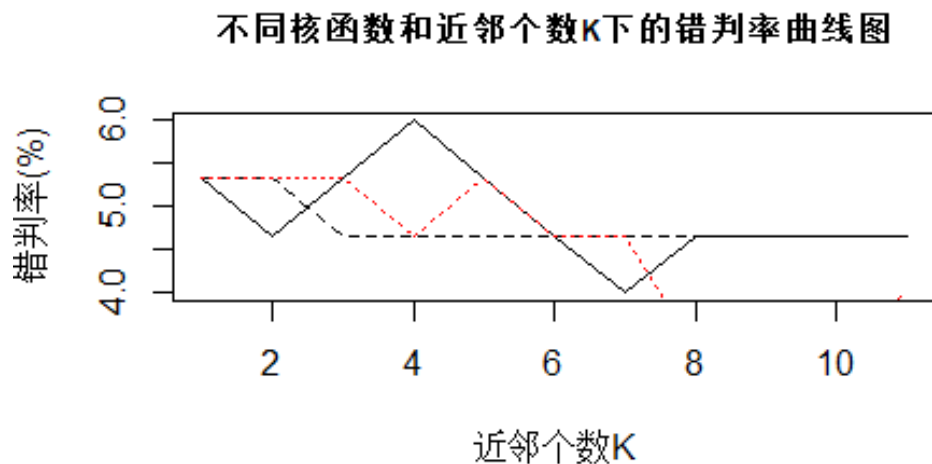


图 4-11

可见，均匀核函数（即不加权）的错判率在 $K=4$ 在附近高于其他两种核。

(2) 选择高斯核函数，设置近邻个数为 7，得到 CT（见图 4-12）如下，只有 3 个记录被分错，此时错判率为 2%。

```
> (CT<-table(Iris_test[,5],fit$fitted.values))
```

	Iris-setosa	Iris-versicolor	Iris-virginica
Iris-setosa	50	0	0
Iris-versicolor	0	49	1
Iris-virginica	0	2	48

```
> (errRatio<-(1-sum(diag(CT))/sum(CT))*100)
[1] 2
```

图 4-12

(3) 利用柱形图对比加权 KNN 与一般 KNN 在测试集上的错判率，如图 4-13 所示。这里，加权 KNN 的错误率较低，为 2%。

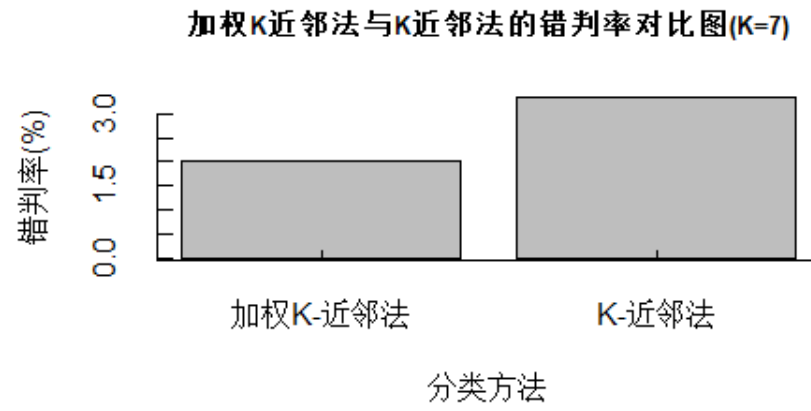


图 4-13

五、 基于神经网络的分类预测

5.1 实验目的

1. 理解 K-近邻法方法的原理和适用性，掌握基于变量重要性的加权 K-近邻法以及基于观测相似性的加权 K-近邻法算法的原理和使用特点；
2. 能够掌握 R 近邻分析的函数、应用和结果解读，基本熟练运用 KNN 算法来分析数据，锻炼分析问题、解决问题并动手实践的能力。

5.2 数据来源与说明

neuralnet、nnet:

Iris Data Set（同第四章、见附件二）

AMORE:

1. 数据来源：UCI ，
[http://archive.ics.uci.edu/ml/datasets/statlog+\(german+credit+data\)](http://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data))
2. 数据名称：Statlog (German Credit Data) Data Set
Statlog（德国信用数据）数据集（见附件三）

此数据集将由一组属性描述的人员分类为良好或不良信用风险，共 20 组属性。

3. 数据属性与说明：

(1) account_check_status : (定性) 现有支票账户的状态

A11: ... <0 DM ; A12: 0 <= ... <200 DM ; A13: ...> = 200 DM /工资分配至少 1 年 ; A14: 否支票账户

(2) duration_in_month: (数字) 月份持续时间

(3) credit_history : (定性) 信用记录

A30: 未获得信用额度/所有信用额已正式支付 ; A31: 该银行的所有信用均已正式偿还 ; A32: 现有信用额已到期支付至现在 ; A33 : 过去延迟付款 ; A34: 现有关键账户/其他信用（不在此银行）

(4) purpose : (定性) 目的

A40: 汽车（新）； A41: 汽车（二手）； A42: 家具/设备； A43: 广播/电视； A44: 家用电器； A45: 维修； A46: 教育； A47: (假期)； A48: 再培训； A49: 业务； A410: 其他

(5) credit_amount : (数字) ; savings : (定性) ; job (定性) ;
present_emp_since (定性) ; installment_as_income_perc (数字) ;
personal_status_sex (定性) ; other_debtors (定性) ;
present_res_since (数字) ; property (定性) ; age (数字) ;
other_installment_plans (定性) ; housing (定性) ;
credits_this_bank (数字) ; people_under_maintenance (数字) ;
telephone(数字)

(6) foreign_worker (定性) 外国工人
——是
——否

5.3 算法描述

神经网络是一种运算模型，由大量的节点（或称神经元）之间相互联接构成。每个节点代表一种特定的输出函数，称为激励函数。每两个节点间的连接都代表一个对于通过该连接信号的加权值，称之为权重，这相当于人工神经网络的记忆。网络的输出则依网络的连接方式，权重值和激励函数的不同而不同。而网络自身通常都是对自然界某种算法或者函数的逼近，也可能是对一种逻辑策略的表达。

神经网络的优势：

- (1) 可以检测因变量与自变量之间的非线性关系
- (2) 可以利用算法并行化实现对大数据的高效训练
- (3) 属于无参模型，能够避免参数估计过程中产生的错误

神经网络的不足：

- (1) 容易陷入局部最优得到不到全局最优解
- (2) 算法训练时间过长，容易导致过度适应

R 语言中关于神经网络的包（package）：nnet、AMORE、neuralnet 等：

- (1) AMORE 包进一步提供了更为丰富的控制参数，并可以增加多个隐藏层。
- (2) neuralnet 包的改进在于提供弹性反向传播算法和更多的激活函数形式。
- (3) nnet 包提供了最常见的前馈反向传播神经网络算法。

5.4 实验过程及结果分析

1. AMORE

(1) 安装并加载 AMORE 包

```
Console Terminal x
C:/Users/dell/Learning/3.1/信息分析与预测/实验5 ANN/
> library(AMORE)
> |
```

图 5-1

(2) 读取数据集（如图 5-1），共 1000 个实例，20 个变量类型：

Data	
data	1000 obs. of 20 variables

检验是否有缺省值，结果显示每个属性下都为 0，则无缺省值。

```
> apply(data,2,function(x) sum(is.na(x))) #列进行检验,如果数据没有缺失值,则每个属性下的值都为0
account_check_status      duration_in_month      credit_history
0                          0                          0
purpose                   credit_amount             savings
0                          0                          0
present_emp_since installment_as_income_perc personal_status_sex
0                          0                          0
other_debtors             present_res_since           property
0                          0                          0
age                       other_installment_plans      housing
0                          0                          0
credits_this_bank                job people_under_maintenance
0                          0                          0
telephone                      foreign_worker
0                          0
```

图 5-2

(3) 由于变量取值存在类型差异，因此需要把 1 到 20 列都转换成数值型。

```
> data
account_check_status duration_in_month credit_history purpose credit_amount savings
1 1 6 2 5 1169 5
2 3 48 4 5 5951 2
3 4 12 2 1 2096 2
4 1 42 4 8 7882 2
5 1 24 2 2 4870 2
```

图 5-3

(4) 划分训练数据集以及测试数据集：

german_test	500 obs. of 20 variables
german_train	500 obs. of 20 variables

图 5-3

- (5) 进行训练。设定输入层输出层，创建多层前馈神经网络，这里我设定输入层 19 个属性，2 个隐藏层分别是 8 和 2 个节点（可以改动），输出层 1 个节点，利用最小均方算法 LMS 求得 5 次结果如图

```
index.show: 1 LMS 0.0379998675142953
index.show: 2 LMS 0.0379998889727415
index.show: 3 LMS 0.0379998988835212
index.show: 4 LMS 0.0379999048046855
index.show: 5 LMS 0.037999908826192
```

图 5-4

- (6) 开始测试。这里规定输出结果小于 1.5 划分为 1（即不是外国工人），大于等于 1.5 划分为 2（即使外国工人）。

```
output=sim(result$net,testdata)#仿真得到输出结果 simulate
output[which(output<1.5)]<-1
output[which(output>=1.5)]<-2
```

图 5-5

- (7) 计算正确率。通过测试我们得到有 482 个被正确分类，正确率达到 0.964:

```
> sum
[1] 482
> cat("正确率",sum/nr
正确率 0.964
```

图 5-6

2. neuralnet

- (1) 安装并加载 neuralnet 包如图 4-7 所示

```
> install.packages("neuralnet")
Error in install.packages : Updating loaded packages
> library(neuralnet)
```

图 5-7

- (2) 读取数据集并划分训练集（0.7）和测试集（0.3）：

▶ Iris	150 obs. of 5 variables
▶ testset	55 obs. of 5 variables
▶ trainset	95 obs. of 8 variables

图 5-8

- (3) 根据数据集在 Class 列取值不同，为训练集新增三种数列

```
trainset$setosa = trainset$species == "setosa"
trainset$virginica = trainset$species == "virginica"
trainset$versicolor = trainset$species == "versicolor"
```

图 5-9

结果如下

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species	setosa	virginica	versicolor
3	4.7	3.2	1.3	0.2	setosa	TRUE	FALSE	FALSE
4	4.6	3.1	1.5	0.2	setosa	TRUE	FALSE	FALSE
6	5.4	3.9	1.7	0.4	setosa	TRUE	FALSE	FALSE
8	5.0	3.4	1.5	0.2	setosa	TRUE	FALSE	FALSE
9	4.4	2.9	1.4	0.2	setosa	TRUE	FALSE	FALSE
10	4.9	3.1	1.5	0.1	setosa	TRUE	FALSE	FALSE
12	4.8	3.4	1.6	0.2	setosa	TRUE	FALSE	FALSE
16	5.7	4.4	1.5	0.4	setosa	TRUE	FALSE	FALSE
18	5.1	3.5	1.4	0.3	setosa	TRUE	FALSE	FALSE
19	5.7	3.8	1.7	0.3	setosa	TRUE	FALSE	FALSE

图 5-10

- (4) 调用 `neuralnet` 函数创建一个包括 3 个隐藏层的神经网络，训练结果有可能随机发生变化，所以得到的结果可能不同。

输出构建好的神经网络模型的结果矩阵（见图），可以看出本次训练执行了 14543 次，结束条件为误差函数的绝对偏导数小于 0.01：

```
> BPnet1$result.matrix #weights and other information
error 1.059731231994
reached.threshold 0.009740011633
steps 14543.000000000000
Intercept.to.1layhid1 -1.372024310694
Sepal.Length.to.1layhid1 -0.441741752311
Sepal.Width.to.1layhid1 -2.581129046591
Petal.Length.to.1layhid1 1.056728687553
```

图 5-11

画出神经网络图（网络及权值参数的可视化）如图

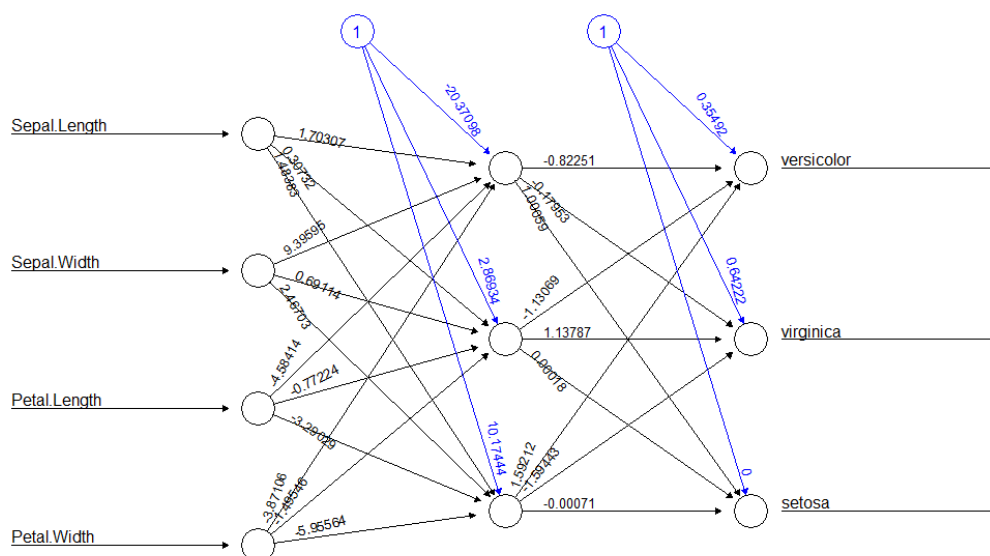


图 5-12

4 个输入变量重要性的可视化如图所示

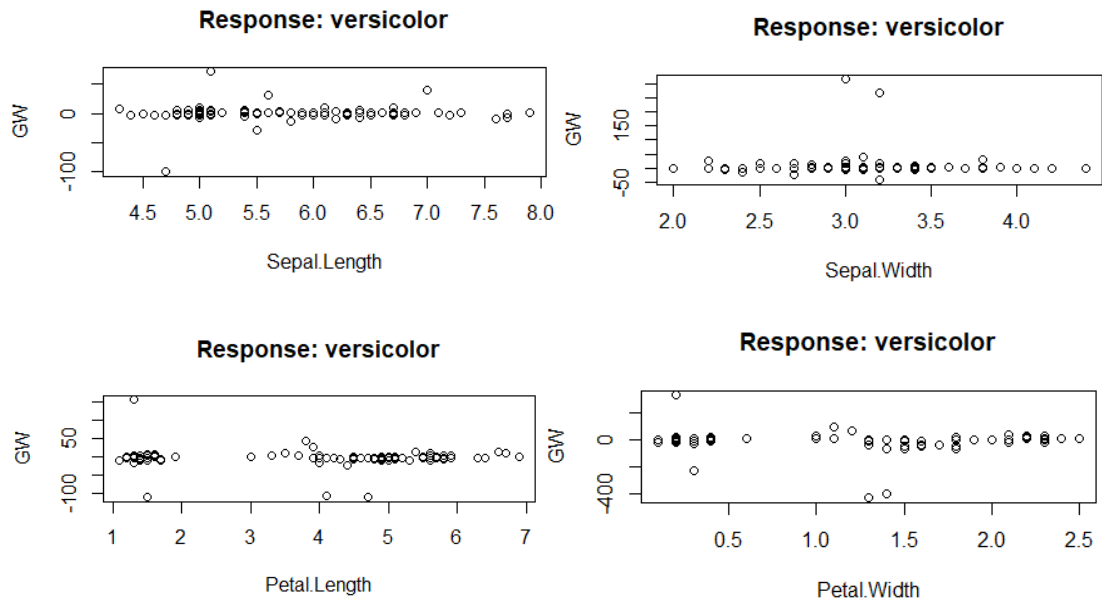


图 5-13

泛化权值图展示了四个协变量以 versicolor 的响应。如图所示的泛化值中，Sepal.Length 和 Petal.Width 都接近于 0，则说明协变量对分类影响不大，然而总体方差大于 1，则意味协变量对方差结果对方差存在非线性影响。

3. nnet

- (1) 安装并加载 nnet 包，数据导入与分类与上述一致，不再重复。
- (2) 使用 nnet 包训练神经网络

```
> BPnet2<-nnet(Species ~ .,data = trainset,size = 2,rang = 0.1,decay = 5e-4,maxit = 200)## ? ? buy
# weights: 25
initial value 115.425443
iter 10 value 48.188602
iter 20 value 1.980196
iter 30 value 1.614000
iter 40 value 1.408036
iter 50 value 1.333348
iter 60 value 1.246535
iter 70 value 1.188874
```

图 5-14

在应用函数时可以实现分类观测，数据源，隐蔽单元个数 (size 参数) 为 2，初始随机数权值 (rang 参数)，权值衰减参数 (decay 参数)，最大迭代次数 (maxit)，整个过程会一直重复直至拟合准则值与衰减项收敛。

- (3) 调用 summary() 输出训练好的神经网络:

```
> summary(BPnet2)
a 7-2-3 network with 25 weights
options were - softmax modelling decay=0.0005
b->h1 i1->h1 i2->h1 i3->h1 i4->h1 i5->h1 i6->h1 i7->h1
-0.06 0.01 -0.08 0.00 0.15 -5.45 5.46 -0.06
b->h2 i1->h2 i2->h2 i3->h2 i4->h2 i5->h2 i6->h2 i7->h2
0.19 0.91 0.57 0.32 0.06 0.15 -0.02 0.06
b->o1 h1->o1 h2->o1
5.14 -26.28 5.11
b->o2 h1->o2 h2->o2
1.91 0.88 1.94
b->o3 h1->o3 h2->o3
-7.05 25.40 -7.05
> |
```

图 5-15

(4) 使用模型 BPnet2 模型完成对测试数据集的预测，得到混淆矩阵

	setosa	versicolor	virginica
setosa	17	0	0
versicolor	0	13	1
virginica	0	2	13

图 5-16

```
Accuracy : 0.9348
95% CI : (0.821, 0.9863)
No Information Rate : 0.3696
P-Value [Acc > NIR] : 1.019e-15

Kappa : 0.9019
```

图 5-17

六、 贝叶斯分类

6.1 实验目的

理论方面，理解贝叶斯分类的基本原理，结合概率论知识进行所属类别的计算，以及拉普拉斯校准的应用；

实践方面，熟悉自然语言处理，可以对数据进行初步处理，形成待分类样本。

6.2 数据来源与说明

1. 数据来源：UCI，
<http://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection>
2. 数据名称：SMS Spam Collection，垃圾短信邮件集合，该语料库是从互联网上免费获取的，其中包含有 5574 条邮件与短信记录。（见附件四）
3. 数据属性与说明：
 - （1） Type：包含两个属性值，spam（垃圾邮件）；ham（正常邮件）
 - （2） Text：短信或邮件的内容

6.3 算法描述

朴素贝叶斯分类指的是需要满足一定的假设条件，在此基础上利用全概率公式与条件概率的相关知识，计算每一个样本被分到 A 类与 B 类的概率，概率大的那一类即为样本最终所属的那一类。

6.4 实验过程及结果分析

1. 读入数据
 - （1） 修改好文件的路径之后，可以看到图 6-1 显示出每一条记录是否为垃圾邮件。

```

3      spam
4      spam
5      ham
6      spam
7      ham
8      ham
9      spam
10     spam
11     ham
12     spam
13     spam
14     ham

```

图 6-1

- (2) 查看数据的结构，图 6-2 显示了数据的属性信息

```

> str(sms_raw)
'data.frame': 5573 obs. of 3 variables:
 $ type: chr "ham" "spam" "spam" "spam" ...
 $ text: chr "Lol your always so convincing."
        eply END SPTV" "Free entry in 2 a wkly comp to

```

图 6-2

- (3) 将种类这一属性用 1/2 分别进行标示并显示出每一种类的记录个数（见图 6-3）

```

> sms_raw$type <- factor(sms_raw$type)
> str(sms_raw$type)
Factor w/ 2 levels "ham","spam": 1 2 2 2 1 2 1 1 2 2 ...
> table(sms_raw$type)

ham spam
4818 755

```

图 6-3

- (4) 图 6-4 显示出了两个类别所占的百分比以及前三条消息的详细内容

```

> prop.table(table(sms_raw$type))#百分数
      ham      spam
0.8645254 0.1354746
> sms_raw$text[1:3]#前三条信息
[1] "Lol your always so convincing."
[2] "SMS. ac Sptv: The New Jersey Devils ar
[3] "Free entry in 2 a wkly comp to win FA
r18's"

```

图 6-4

2. 生成语料库

- (1) 处理数据的第一步是建立一个语料库，为此先进行“tm”包的安装，以及 NLP 自然语言处理，之后将 text 属性下的内容生成一个向量传递给语料库生成函数 VCorpus()，图 6-5 显示出了语料库的结果

```

> library(NLP)#自然语言处理
> library(tm)
> sms_corpus <- VCorpus(VectorSource(sms_raw$text))
> print(sms_corpus)
<<VCorpus>>
Metadata: corpus specific: 0, document level (indexed): 0
Content: documents: 5573

```

图 6-5

- (2) 查看语料库的前两条记录，并以字符串的形式显示出来（见图 6-6）

```
> inspect(sms_corpus[1:2])
<<VCorpus>>
Metadata: corpus specific: 0, document level (indexed): 0
Content: documents: 2

[[1]]
<<PlainTextDocument>>
Metadata: 7
Content: chars: 30

[[2]]
<<PlainTextDocument>>
Metadata: 7
Content: chars: 120

> as.character(sms_corpus[[1]])
[1] "lol your always so convincing."
```

图 6-6

3. 根据语言的特点对语料库进行清洗，主要是清除掉标点符号、数字、大小写转换等等。

- (1) 建立映射函数，首先将所有字母用小写展示，图 6-7 显示了转换后的第一条记录。

```
> sms_corpus_clean <- tm_map(sms_corpus, tolower) #映射函数
> as.character(sms_corpus_clean[[1]])
[1] "lol your always so convincing."
```

图 6-7

- (2) 接着是去除数字、停用词与标点符号
- (3) 进行词干的提取，这里需要进行包的安装。示例是给出了 learn 的不同形式，最终都返回了 learn，图 6-8 显示了提取词干之后的前三条记录。

```
> wordstem(c("learn", "learned", "learning", "learns")) #示例：执行完之后全是learn
[1] "learn" "learn" "learn" "learn"
> sms_corpus_clean <- tm_map(sms_corpus_clean, stemDocument)
> sms_corpus_clean <- tm_map(sms_corpus_clean, stripwhitespace)
> inspect(sms_corpus_clean[1:3])
<<VCorpus>>
Metadata: corpus specific: 0, document level (indexed): 0
Content: documents: 3

[[1]]
<<PlainTextDocument>>
Metadata: 7
Content: chars: 17

[[2]]
<<PlainTextDocument>>
Metadata: 7
Content: chars: 102

[[3]]
<<PlainTextDocument>>
Metadata: 7
Content: chars: 104
```

图 6-8

4. 准备数据，也就是产生稀疏矩阵

- (1) 数据清洗完成后就可以产生稀疏矩阵，它是将文本文档拆分成词语，显示出文本中出现的所有单词以及每一条记录中该单词是否出现。图 6-9 显示了 1-5 这 5 个文本中是否出现排在 100-105 位置上的这五个词。

```
> sms_tdm <- DocumentTermMatrix(sms_corpus_clean)#转换形式
> inspect(sms_tdm[1:5, 100:105])
<<DocumentTermMatrix (documents: 5, terms: 6)>>
Non-/sparse entries: 0/30
Sparsity           : 100%
Maximal term length: 11
weighting           : term frequency (tf)
Sample             :
  Terms
Docs affidavit afford afghanistan afraid africa african
1         0      0      0          0      0      0      0
2         0      0      0          0      0      0      0
3         0      0      0          0      0      0      0
4         0      0      0          0      0      0      0
5         0      0      0          0      0      0      0
```

图 6-9

- (2) 我们还可以查看特定的单词是否出现，图 6-10 显示了以下四个单词的出现情况。

```
> d<-c("price","free","call","urgent")#字符型向量，看这几个关键字的情况
> sparsematrix<-DocumentTermMatrix(sms_corpus_clean,control=list(dictionary=d))
> inspect(sparsematrix)
<<DocumentTermMatrix (documents: 5573, terms: 4)>>
Non-/sparse entries: 919/21373
Sparsity           : 96%
Maximal term length: 6
weighting           : term frequency (tf)
Sample             :
  Terms
Docs call free price urgent
1008  2   3   0   0
1848  2   3   1   0
2044  1   2   1   0
2693  2   2   0   2
3265  1   3   0   0
3384  2   2   0   0
390   1   2   1   0
43    2   2   0   0
711   1   2   1   0
871   1   2   1   0
```

图 6-10

5. 产生词云，函数 wordcloud 设定了单词在语料库中出现的最少次数，以及产生词云的字号范围，6-11 显示出了词云图。

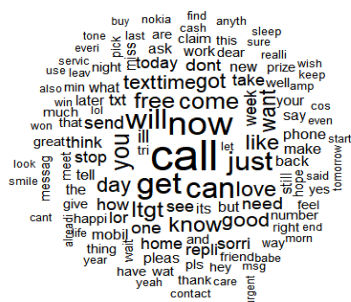


图 6-11

6. 图 6-12 与图 6-13 分别显示出了垃圾邮件和正常邮件中词云的分布。



图 6-12

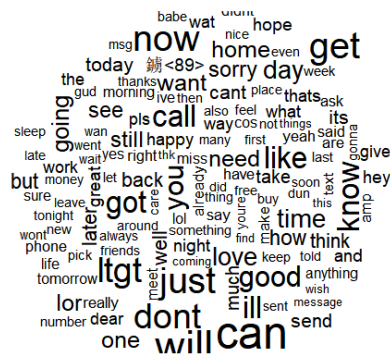


图 6-13

7. 减少单词的个数，设定这个单词至少出现在 5 封邮件中，图 6-14 显示了删减之后的单词

```
> sms_freq_words<-findFreqTerms(sms_tdm_train,5) #at least 5 times , 这个单词至少出现在5条短信里
> str(sms_freq_words)
chr [1:1213] "abiola" "abl" "abt" "accept" "access" "account" "across" "activ" "actual" "actualli"
> findFreqTerms(sms_tdm_train,5)
[1] "abiola" "abl" "abt" "accept" "access" "account"
[10] "actualli" "add" "address" "admir" "adult" "advanc"
[19] "after" "age" "ago" "ahead" "aight" "aint"
[28] "all" "almost" "alon" "alreadi" "alright" "alrite"
[37] "and" "angri" "ani" "announc" "anoth" "answer"
[46] "anyth" "anytim" "anyway" "apart" "app" "appli"
[55] "ard" "are" "area" "argument" "around" "arrang"
[64] "ask" "askd" "asleep" "ass" "attempt" "auction"
[73] "await" "award" "away" "awesom" "babe" "babi"
[82] "bak" "balanc" "bank" "bare" "bath" "batteri"
[91] "beauti" "becaus" "becom" "bed" "bedroom" "begin"
[100] "better" "bid" "big" "bill" "bird" "birthday"
```

图 6-14

- 将删减之后的单词分为测试集与训练集，并用 yes/no 代替出现的次数和没有出现。
- 用训练集数据建立贝叶斯模型，并用测试集数据对模型进行评估，图 6-15 显示了预测之后的分类

```
> sms_test_pred
[1] ham ham ham ham ham ham ham ham ham
[30] spam ham ham ham ham ham spam ham ham
[59] ham ham ham ham ham ham ham spam spam
[88] spam ham ham ham ham ham ham ham ham
[117] ham ham ham ham ham ham ham ham spam
[146] ham ham ham ham ham ham ham ham ham
[175] ham ham ham spam ham ham ham ham ham
[204] ham ham spam spam ham ham ham ham ham
```

图 6-15

10. 图 6-16 将预测之后的结果，对角线之和表示预测正确或错误，发现这是有 30 个预测错误。

Cell Contents			
			N
	N / Row Total		
	N / Col Total		

Total Observations in Table: 1402

predicted	actual		Row Total
	ham	spam	
ham	1213	23	1236
	0.981	0.019	0.882
	0.994	0.126	
spam	7	159	166
	0.042	0.958	0.118
	0.006	0.874	
column Total	1220	182	1402
	0.870	0.130	

图 6-16

11. 通过拉普拉斯方法提升模型的准确率, 分别设置 `laplace=1` 和 `laplace=0.5`, 通过图 6-17 可以看到提升并不是很大。

predicted	actual		Row Total
	ham	spam	
ham	1213	22	1235
	0.982	0.018	0.881
	0.994	0.121	
spam	7	160	167
	0.042	0.958	0.119
	0.006	0.879	
column Total	1220	182	1402
	0.870	0.130	

图 6-17

七、 基于支持向量机的分类预测

7.1 实验目的

1. 理论方面：理解支持向量分类和支持向量回归的基本原理、适用性和方法特点。
2. 实践方面：掌握 R 的支持向量预测、应用及结果解读，能够正确运用支持向量法实现数据的分类预测。

7.2 数据来源与说明

1. 数据来源：UCI
2. 数据名称：输血服务中心数据集（同第二、三章、见附件一）

7.3 算法描述

在机器学习中，支持向量机(SVM)是与相关的学习算法有关的监督学习模型，可以分析数据，识别模式，用于分类和回归分析。给定一组训练样本，每个标记为属于两类，一个 SVM 训练算法建立了一个模型，分配新的实例为一类或其他类，使其成为非概率二元线性分类。除了进行线性分类，支持向量机可以使用所谓的核技巧，它们的输入隐含映射成高维特征空间中有效地进行非线性分类。

1. 线性分类器：

首先给出一个非常非常简单的分类问题（线性可分），我们要用一条直线，将下图中黑色的点和白色的点分开，很显然，图 7-1 上的这条直线就是我们要求的直线之一（可以有无数条这样的直线）

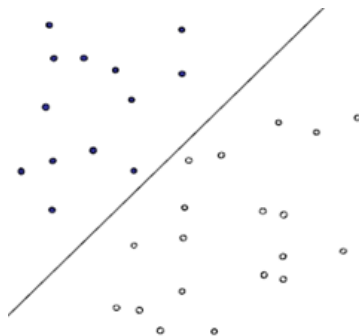


图 7- 1
49

令黑色的点 $= -1$ ，白色的点 $= +1$ ，直线 $f(x) = w \cdot x + b$ ，这儿的 x 、 w 是向量，其实写成这种形式也是等价的 $f(x) = w_1x_1 + w_2x_2 \cdots + w_nx_n + b$ ，当向量 x 的维度=2 的时候， $f(x)$ 表示二维空间中的一条直线，当 x 的维度=3 的时候， $f(x)$ 表示 3 维空间中的一个平面，当 x 的维度= $n > 3$ 的时候，表示 n 维空间中的 $n-1$ 维超平面。

当有一个新的点 x 需要预测属于哪个分类的时候，我们用 $\text{sgn}(f(x))$ ，就可以预测了， sgn 表示符号函数，当 $f(x) > 0$ 的时候， $\text{sgn}(f(x)) = +1$ ，当 $f(x) < 0$ 的时候 $\text{sgn}(f(x)) = -1$ 。

怎样才能取得一个最优的划分直线 $f(x)$ 呢？图 7-2 的直线表示几条可能的 $f(x)$

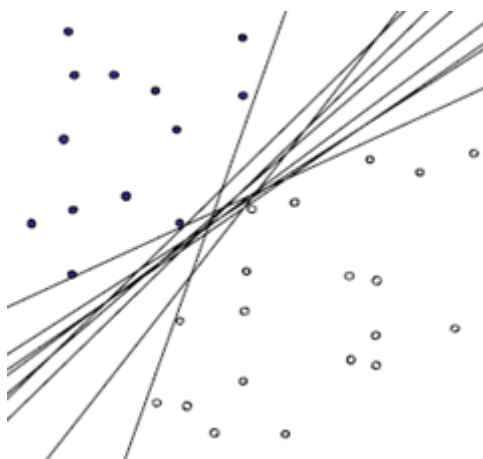


图 7- 2

一个很直观的感受是，让这条直线到给定样本中最近的点最远，从直观上来说，就是分割的间隙越大越好，把两个类别的点分得越开越好。在 SVM 中叫做 Maximum Marginal 是 SVM 的一个理论基础之一。选择使得间隙最大的函数作为分割平面是由很多道理的，比如说从概率的角度上来说，就是使得置信度最小的点置信度最大。

这里给出 M 的式子：

$$M = \frac{2}{\sqrt{w \cdot w}}$$

另外支持向量位于 $wx + b = 1$ 与 $wx + b = -1$ 的直线上，我们在前面乘上一个该点所属的类别 y ，就可以得到支持向量的表达式为： $y(wx + b) = 1$ ，这

样就可以更简单的将支持向量表示出来了。

当支持向量确定下来的时候，分割函数就确定下来了，两个问题是等价的。得到支持向量，还有一个作用是，让支持向量后方那些点就不用参与计算了。

最后，给出要优化求解的表达式：

$$\max \frac{1}{\|w\|} \rightarrow \min \frac{1}{2} \|w\|^2$$

$\|w\|$ 的意思是 w 的二范数，跟上面的 M 表达式的分母是一个意思，之前得到， $M = 2 / \|w\|$ ，最大化这个式子等价于最小化 $\|w\|$ ，另外由于 $\|w\|$ 是一个单调函数，我们可以对其加入平方，和前面的系数，这个式子是为了方便求导。

这个优化问题可以用拉格朗日乘子法去解，使用了 KKT 条件的理论，这里直接作出这个式子的拉格朗日目标函数：

$$L(w, b, a) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i (y_i (w^T x_i + b) - 1)$$

新问题加上其限制条件是（对偶问题）：

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \\ \text{s.t.}, \quad & \alpha_i \geq 0, i = 1, \dots, n \\ & \sum_{i=1}^n \alpha_i y_i = 0 \end{aligned}$$

这个就是我们需要最终优化的式子。至此，得到了线性可分问题的优化式子。

2. 线性不可分的情况：

实际中，我们会经常遇到线性不可分的样例，此时，我们的常用做法是把样例特征映射到高维空间中去(如图 7-3)；

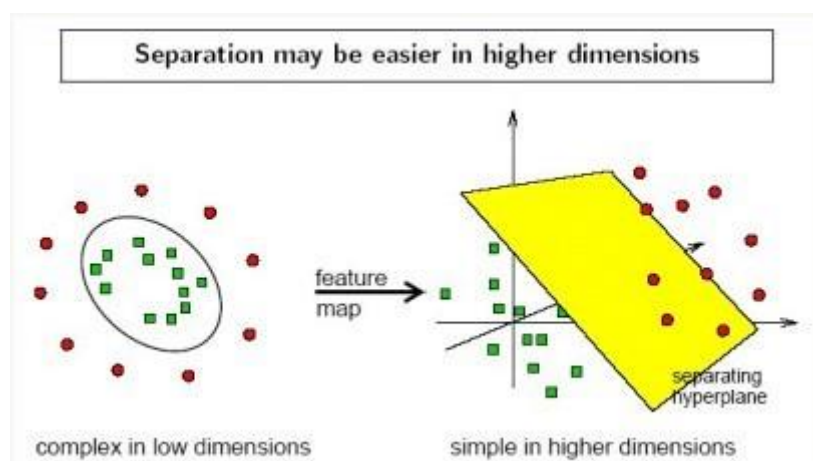


图 7- 3

线性不可分映射到高维空间，可能会导致维度大小高到可怕的(19 维乃至无穷维的例子)，导致计算复杂。核函数的价值在于它虽然也是讲特征进行从低维到高维的转换，但核函数绝就绝在它事先在低维上进行计算，而将实质上的分类效果表现在了高维上。

7.4 实验过程及结果分析

1. 模拟线性可分下的支持向量分类

(1) 构建数据集

在线性可分的原则下，随机生成训练样本集和测试样本集。其中的输入变量有 2 个，输出变量类别为-1 和+1

可以看到训练样本集和测试样本集的部分数据如图 7-4 所示：

```
> data_train
      Fx1      Fx2 Fy
1  0.58552882  0.70946602 -1
2 -0.10930331 -0.45349717 -1
3  0.60588746 -1.81795597 -1
4  0.63009855 -0.27618411 -1
5 -0.28415974 -0.91932200 -1
6 -0.11624781  1.81731204 -1
7  0.37062786  0.52021646 -1
8 -0.75053199  0.81689984 -1
9 -0.88635752 -0.33157759 -1
10 1.12071265  0.29872370 -1
11 0.77962192  1.45578508 -1
12 -0.64432843 -1.55313741 -1
13 -1.59770952  1.80509752 -1

> data_test
      Fx1      Fx2 Fy
1  2.1453831  2.5431436  1
2  1.1956309  3.9771109  1
3  0.9712207  1.8670992 -1
4  2.1720425  1.1920466  1
5  2.0365237  2.3248701  1
6  0.5360985  0.6449175  1
7  1.8869469 -0.3918194 -1
8 -0.9806329  0.6873321 -1
9  0.9949565  3.6577198  1
10 0.9002024  0.8054533  1
```

图 7-4

可以看到（见图 7-5），训练样本集有 40 个记录，测试样本集有 10 个记录：

Data		
data_test	10 obs. of 3 variables	
data_train	40 obs. of 3 variables	
x	num [1:10, 1:2] 2.145 1.196 0.971 2....	
values		
y	num [1:10] 1 1 -1 1 1 1 -1 -1 1 1	

图 7-5

(2) 画出数据集的情况

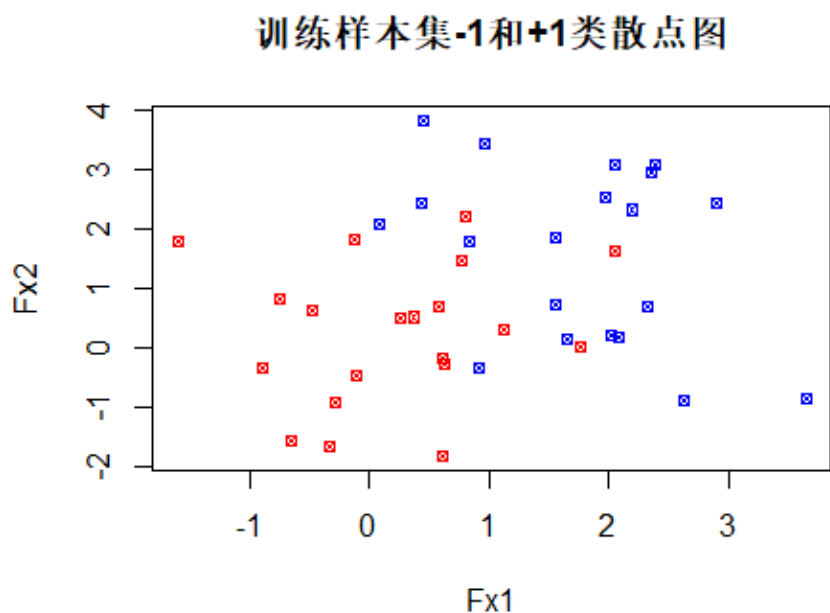


图 7-6

(3) 构建支持向量机

采用线性核函数，比较当损失惩罚参数较大和较小时的支持向量个数和最大边界超平面：

```
svm(formula = Fy ~ ., data = data_train, type = "c-classification", kernel = "linear",
    cost = 10, scale = FALSE)
```

```
Parameters:
  SVM-Type:  C-classification
  SVM-Kernel: linear
    cost:  10
   gamma:  0.5
```

```
Number of Support Vectors:  16
```

```
( 8 8 )
```

```
Number of Classes:  2
```

```
Levels:
-1  1
```

图 7-7

可以看到（见图 7-7），当损失函数参数 $C=10$ 时，有 16 个支持向量（两类各有 8 个），此处使用的核函数为 linear（高斯）核函数， γ 取值 0.5，分类图像如图 7-8 所示：

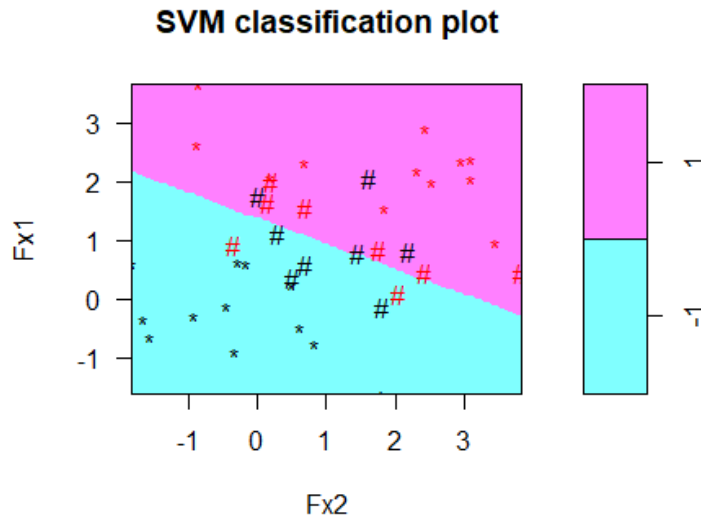


图 7-8

可以看到（见图 7-9），当损失惩罚参数 $C=0.1$ 时，由于惩罚程度降低，所有包含的支持向量增加到了 25 个。

```
svm(formula = Fy ~ ., data = data_train, type = "c-classification", kernel = "linear",
     cost = 0.1, scale = FALSE)

Parameters:
  SVM-Type:  C-classification
 SVM-kernel: linear
    cost:   0.1
   gamma:   0.5

Number of Support Vectors: 25
( 12 13 )

Number of classes: 2

Levels:
-1 1
```

图 7-9

(4) 利用 10 折交叉验证找到预测误差最小时的损失惩罚参数

```

- sampling method: 10-fold cross validation
- best parameters:
  cost
  1
- best performance: 0.15
- Detailed performance results:
  cost error dispersion
1 1e-03 0.575 0.3545341
2 1e-02 0.325 0.2058182
3 1e-01 0.200 0.1581139
4 1e+00 0.150 0.1290994
5 5e+00 0.175 0.1687371
6 1e+01 0.175 0.1687371
7 1e+02 0.175 0.1687371
8 1e+03 0.175 0.1687371

Parameters:
SVM-Type: C-classification
SVM-Kernel: linear
cost: 1
gamma: 0.5

Number of Support Vectors: 17
( 8 9 )

Number of classes: 2

Levels:
-1 1

```

图 7-10

可以看到（见图 7-10），采用 10 折交叉验证的预测误差最低时的 C 等于 1，平均误判率为 0.15。该参数下的模型为最优模型，找到了 17 个支持向量。

（5） 利用最优模型对测试样本集

```

> (ConfM<-table(yPred,data_test$Fy))

yPred -1 1
      -1  1 2
      1  2 5
> (Err<-(sum(ConfM)-sum(diag(ConfM)))/sum(ConfM))
[1] 0.4

```

图 7-11

可以看到（见图 7-11），将-1 预测为+1 的数量为 2，将+1 预测到-1 的数量为 2，故误判率为 40%。

2. 模拟线性不可分下的支持向量分类

（1） 构造数据集

在线性不可分的原则下，随机生成训练样本集合测试样本集。类似线性可分情况下的其中的输入变量有 2 个，输出变量类别为 1 和 2，训练样本散点图如图 7-12：

训练样本集散点图

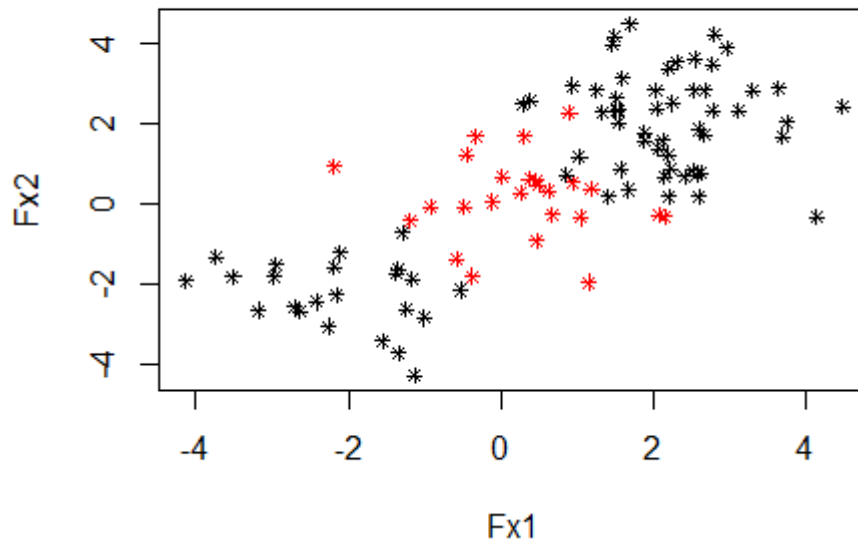


图 7-12

可见，样本集是线性不可分的。

- (2) 采用径向基核函数，利用 10 折交叉验证找到预测误差最小下的最优参数和最优模型，不同参数组合下的预测错误率如图 7-13 所示：

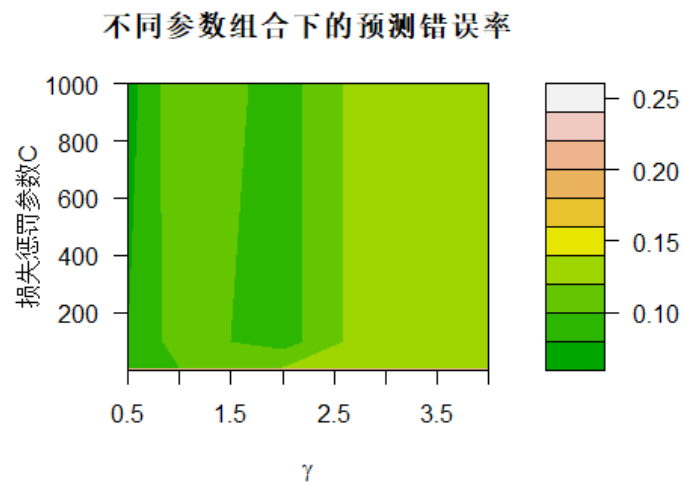


图 7-13

可以看到，颜色越深预测误差越小；即当 gamma 值等于 0.5, C 等于 1000 时，模型取最优，从 summary 结果（如图 7-14）也可以看出：


```

- sampling method: 10-fold cross validation

- best parameters:
  gamma cost
  0.5 1000

- best performance: 0.07

- Detailed performance results:
  gamma cost error dispersion
1    0.5 1e-03 0.25 0.1354006
2    1.0 1e-03 0.25 0.1354006

```

图 7-14

(3) 可视化超平面

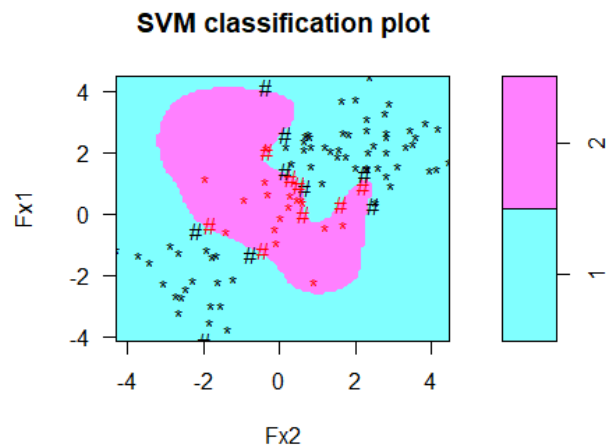


图 7-15

(4) 利用 10 折交叉验证训练的最优模型对测试集做预测

```

> (confM<-table(yPred,data_test$Fy))

yPred  1  2
      1 71  4
      2  4 21
> (Err<-(sum(confM)-sum(diag(confM)))/sum(confM))
[1] 0.08

```

图 7-16

可以看到，预测误差率仅为 0.08，说明训练所得模型效果的表现不错。

3. 模拟多分类的支持向量分类

- (1) 在线性不可分的原则下，随机生成训练样本集；其中，输入变量有 2 个，输出变量有 0, 1 和 2 三类。训练集总共包含 250 个观测，总共有 3 个类。

训练样本集散点图

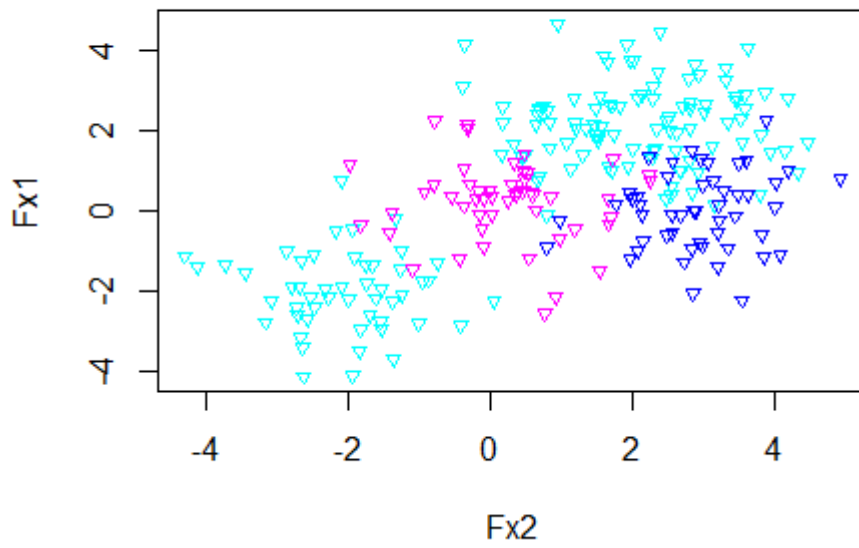


图 7-17

- (2) 采用径向基核函数, 利用 10 折交叉验证找到预测误差最小下的最优参数和最优模型, 最优模型时当 $C=5$, $\gamma=1$ 时的模型, 模型将训练集划分结果如下:

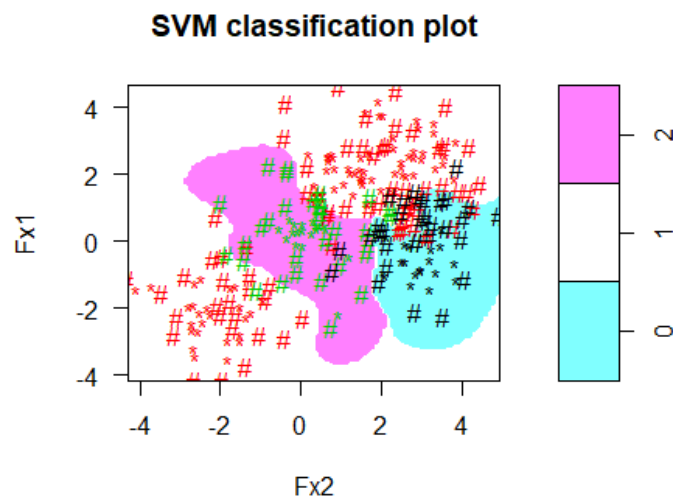


图 7-18

- (3) 利用最优模型对训练样本做预测, 观测多类别预测的依据。

```

> yPred<-predict(svmFit,data)
> (ConfM<-table(yPred,data$Fy))

yPred   0   1   2
  0  42   3   0
  1   6 143   6
  2   2   4  44
> (Err<-(sum(ConfM)-sum(diag(ConfM)))/sum(ConfM))
[1] 0.084

```

图 7-19

可以看到，0 类误判为 1 类和 2 类的数量分别为 3 和 0；1 类误判为 0 类和 2 类的数量分别为 6 和 6；2 类误判为 0 类和 1 类的数量分别为 2 和 4。总体预测误差为 0.084，说明模型训练效果表现良好。

4. e1071 binary classes

(1) 读取数据

读取输血服务中心数据集，并划分训练数据集（2/3）以及测试数据集（1/3）

● Blood_test	249 obs. of 5 variables
● Blood_train	499 obs. of 5 variables
● BloodData	748 obs. of 5 variables

图 7-20

以 Monetary 和 Time 为例，画出数据集的情况：

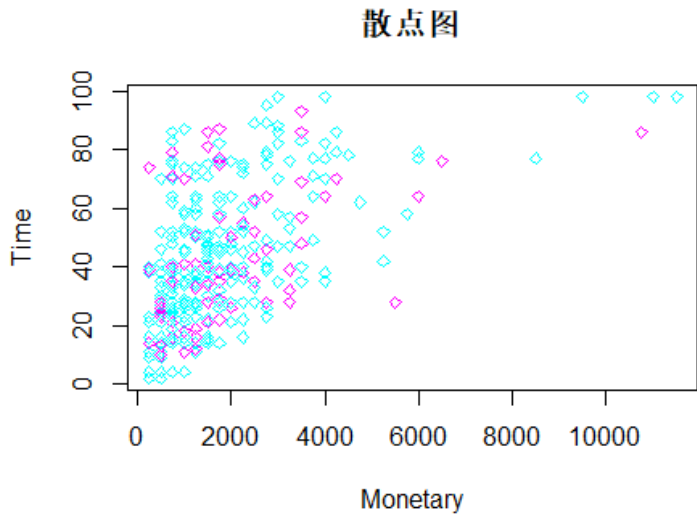


图 7-21

可见，样本集是线性不可分的。

(2) 采用径向基核函数，利用 10 折交叉验证找到预测误差最小下的最优参数和最优模型

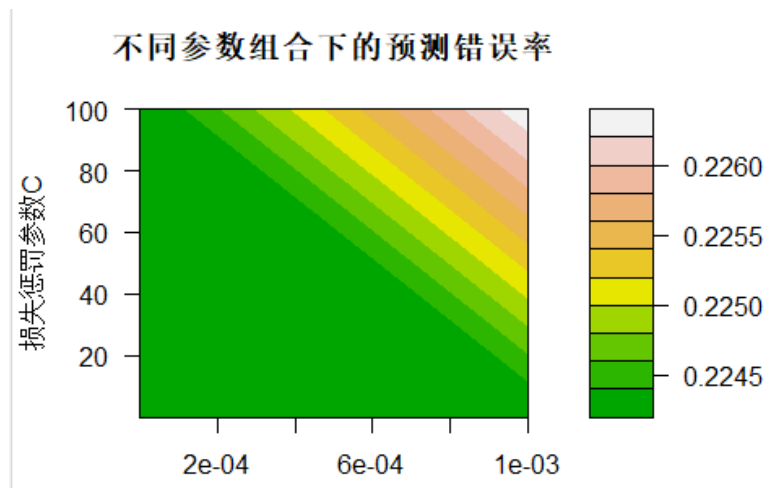


图 7-22

可以看到，颜色越深预测误差越小；即当 γ 值等于 0.000001，C 等于 0.001 时，模型取最优，从 summary 结果也可以看出：

```
cost = 10^(-3:2), type = "C-classification"

Parameters:
  SVM-Type:  C-classification
  SVM-Kernel: radial
    cost:  0.001
   gamma: 1e-06

Number of Support Vectors: 224

( 112 112 )
```

图 7-23

(3) 利用 10 折交叉验证训练的最优模型对测试集做预测

```
> yPred<-predict(BestSvm,Blood_test)
> (ConfM<-table(yPred,Blood_test$DonatedBloodInMarch2007))

yPred    0    1
    0 183   66
    1   0    0
> (Err<-(sum(ConfM)-sum(diag(ConfM)))/sum(ConfM))
[1] 0.2650602
```

图 7-24

可以看到，预测误差率为 0.265，说明训练所得模型效果的表现略差。

八、 常规聚类

8.1 实验目的

1. 理解聚类分析的目标和意义，掌握 K-means 聚类和层次聚类的原理、适用性和方法特点。
2. 掌握 R 的各种聚类方法的实现、应用以及结果解读，能够正确运用聚类方法解决实际应用中的数据全方位自动分组问题。

8.2 数据来源与说明

1. 数据来源：UCI ， <http://archive.ics.uci.edu/ml/datasets/Iris>
2. 数据名称：Iris Data Set 鸢尾植物数据库（见附件二）

8.3 算法描述

1. 基于质心的分类：K-means 聚类

第一步：指定聚类数目 K，既要考虑最终的聚类效果，也要根据研究问题的实际需要来确定；

第二步：确定 K 个初始类质心，这将直接影响到聚类算法收敛的速度；

第三步：根据最近原则进行聚类，以此计算每个观测点到 K 个类质心的距离，将每个观测分派到最近的类中，形成 K 个类；

第四步：重新计算 K 个类的质心点，确定原则是：依次计算各类中所有观测点在各个变量上的均值，并以均值点作为新的类质心点。

第五步：判断是否满足终止聚类算法的条件，一般是指定迭代次数，若不满足，返回第三步不断重复上述过程。

2. 层次聚类—基于连通性的聚类

(1) 基本过程：层次聚类是将各个观测逐步合并成小类，再将小类逐步合并的过程。

➤ 首先，每个观测点自成一类

➤ 然后计算所有观测点彼此之间的距离，并将其中距离最近的观测点聚成

一个小类，形成 $n-1$ 个类

- 接下来，再次度量剩余观测点和小类之间的距离，并将当前距离最近的观测点或小类再聚成一类

重复上述过程，直至所有的观测点聚到一起形成一个最大的类。

(2) 层次聚类中距离的测度

- Single-linkage: 观测点联通一个小类所需的距离长度，是该观测与小类中所有观测距离中最小的值
- Complete-linkage: 观测点联通一个小类所需的距离长度，是该观测与小类中所有观测距离中最大的值
- 质心法: 观测点联通一个小类所需的距离长度，是该观测与小类质心点的距离
- Average linkage: 观测点联通一个小类所需的距离长度，是该观测与小类中所有观测距离的平均值

8.4 实验过程及结果分析

1. 简单原理解解

- (1) 首先随机生成正态分布的数据点，图 8-1 显示了数据点的分布

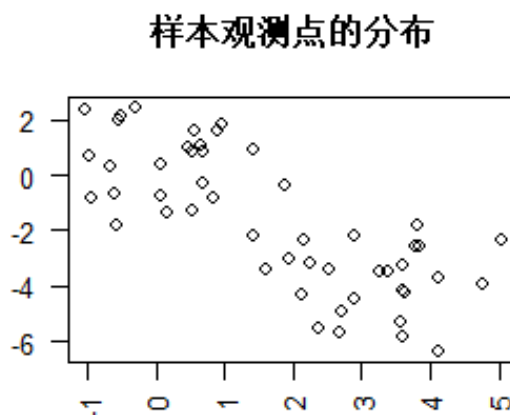


图 8-1

- (2) 对这些数据进行聚类，这里设定有两个类别，也就产生两个均值点，不同的 pch 代表用不同的符号表示不同类别，图 8-2 显示了两类数据点

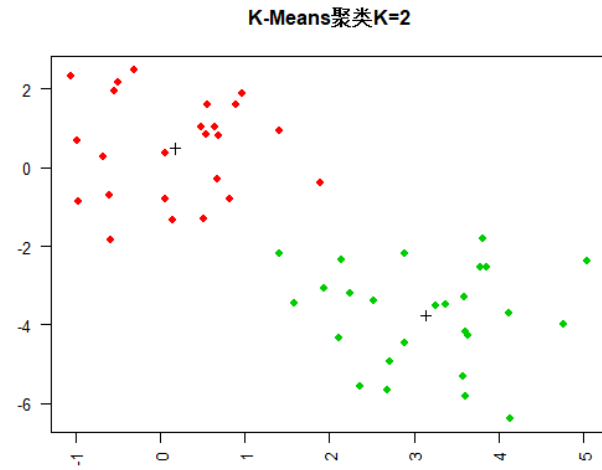


图 8-2

(3) 图 8-3、8-4 显示了当 $K=4$ 时的聚类情况，并分别生成 $nstart$ 不同取值之下的结果。

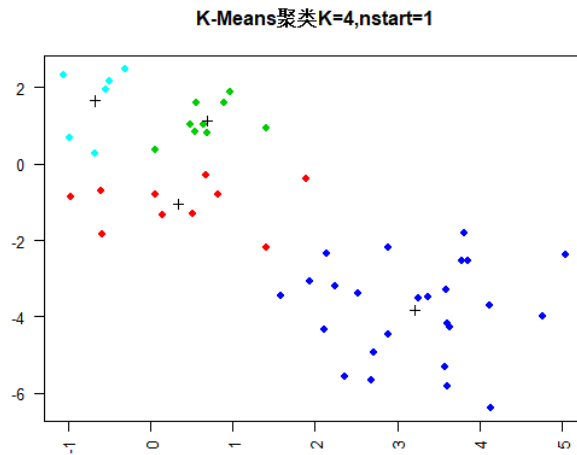


图 8-3

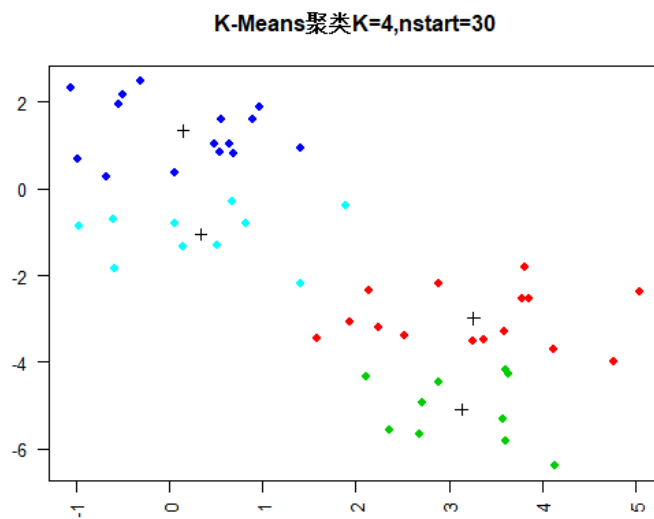


图 8-4

2. 接下来是对实例数据进行聚类

- (1) 我们首先设定有三个类别，也就产生三个均值点，图 8-5 显示了每个类别中有几个记录以及每一个类的在四个属性上的均值点

```
> set.seed(12345)
> CluR<-kmeans(x=CluData,centers=3,iter.max=10,nstart=30)
> CluR$size##每一个类中有几个值
[1] 8 9 13
> CluR$centers##四个均值点
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1    6.750000    2.975000    5.887500    2.012500
2    4.877778    3.255556    1.377778    0.200000
3    5.876923    2.915385    4.561538    1.623077
```

图 8-5

- (2) 进行绘图。可视化操作，并标注横纵坐标所代表的意义，以及图例，如图 8-6 所示，

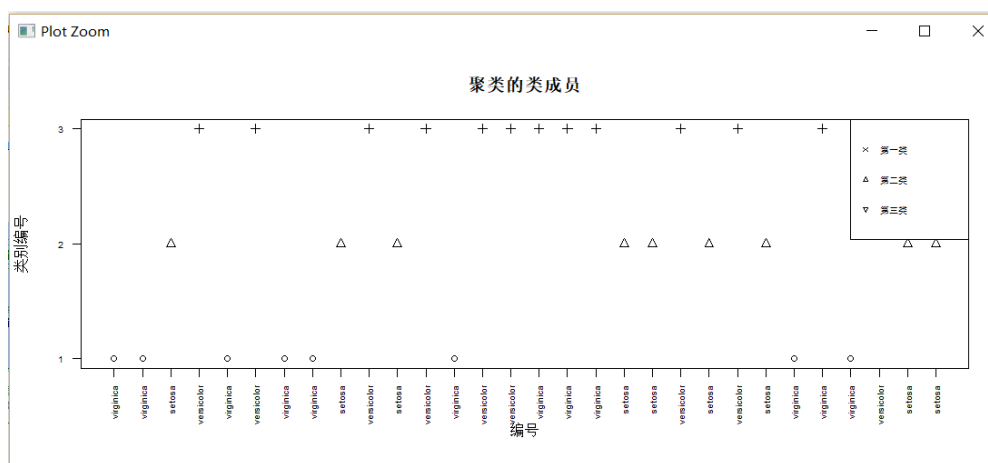


图 8-6

- (3) 进行聚类特征的可视化，图 8-7 显示了每一个类均值点的变化

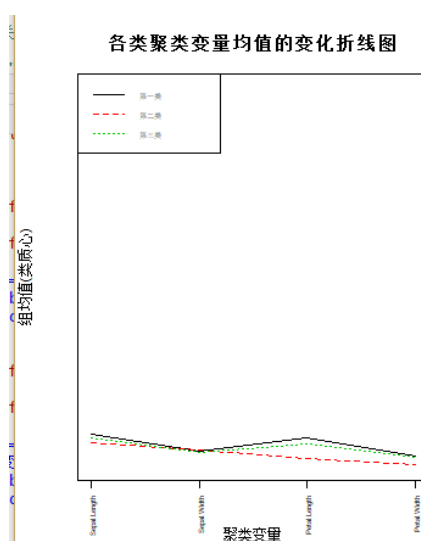


图 8-7

(4) 之后是聚类效果的可视化评价，可以显示变量两两之间的关系，如图 8-8

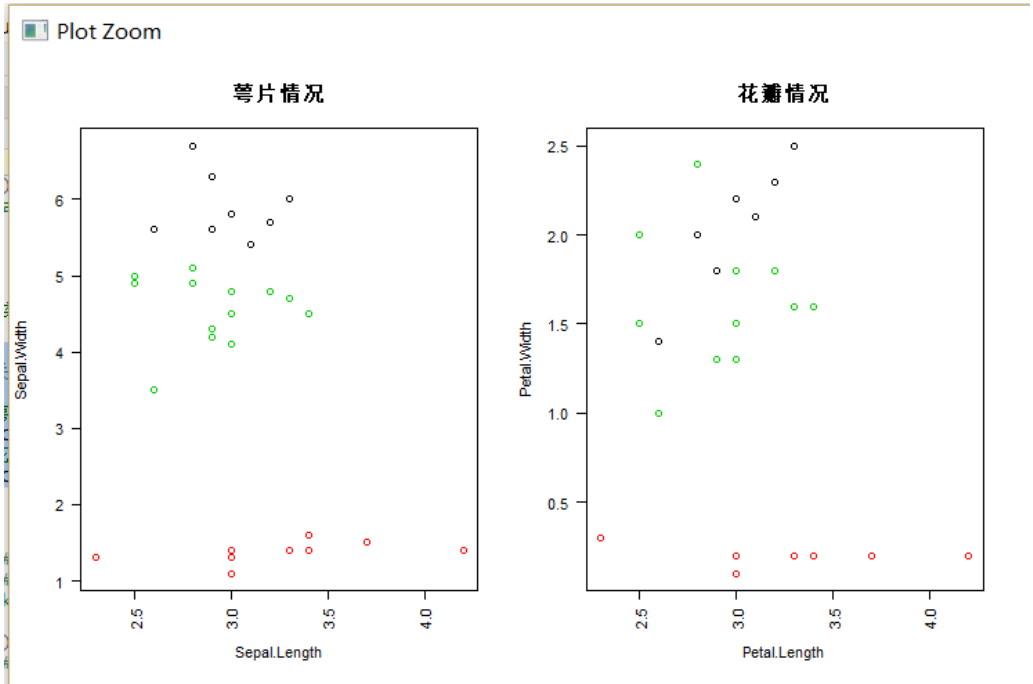


图 8-8

3. 层次聚类法

(1) 首先画出层次聚类的树形图，如图 8-9

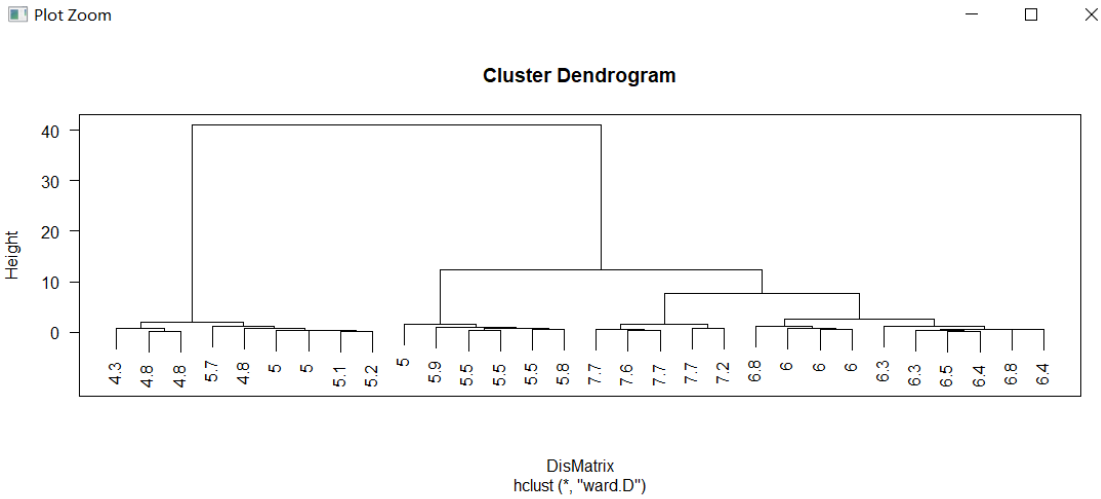
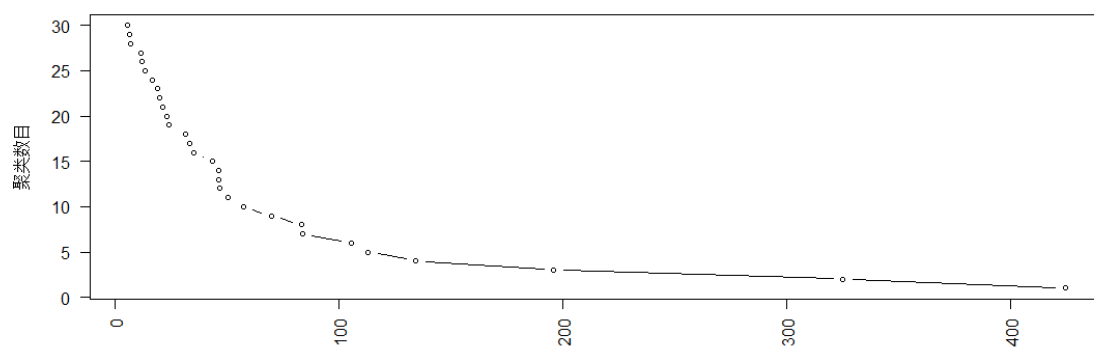
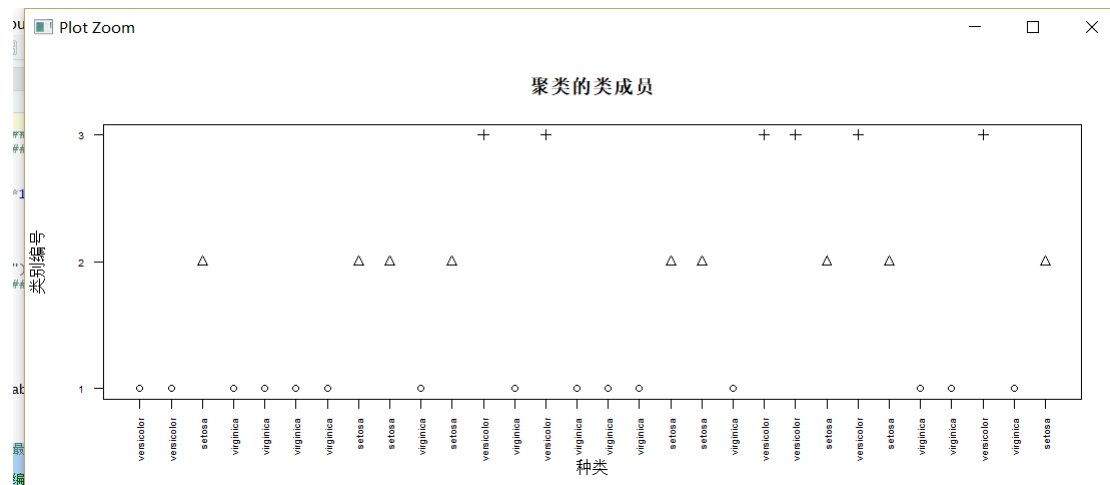


图 8-9

(2) 图 8-10 显示了层次聚类的碎石图



(3) 最后是通过层次聚类形成的三个类别，见图 8-11



九、 特色聚类 kohonen 网络聚类

9.1 实验目的

1. 理论方面：理解特色聚类 kohonen 网络聚类的基本原理、适用性和方法特点。
2. 实践方面：掌握 R 的 SOM 聚类方法实现、应用及结果解读，能够正确运用特色聚类方法实现数据的全方位自动分类。

9.2 数据来源与说明

1. 数据来源：UCI
2. 数据名称：iris 鸢尾花数据集（见附件二）

9.3 算法描述

自组织映射（Self Organization Map, SOM）神经网络是较为广泛应用于聚类的神经网络，它是由 Kohonen 提出的一种无监督学习的神经元网络模型。主要功能是将输入的 n 维空间数据映射到一个较低的维度（通常是一维或者二维）输出，同时保持数据原有的拓扑逻辑关系。

SOM 指的竞争网络，顾名思义就是网络节点相互竞争。对于每一个输入的数据点，网络节点都要进行竞争，最后只有一个节点获胜。获胜节点会根据赢得的数据点进行演化，变得与这个数据点更匹配。如果数据可以明显地分为多个 cluster，节点变得跟某个 cluster 内的数据点更为匹配，一般而言就会跟其它 cluster 不太匹配，这样其它节点就会赢得了其它 cluster 的数据点。如此反复学习，每个节点就会变得只跟特定的一个 cluster 匹配，这样就完成了数据点的聚类。

9.4 实验过程及结果分析

1. 输入数据

在 R 中直接获取 iris 数据集，去掉最后一列，提取输入变量得到见图 9-1

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2
6	5.4	3.9	1.7	0.4
7	4.6	3.4	1.4	0.3
8	5.0	3.4	1.5	0.2

图 9-1

2. 构建 SOM 模型

邻域半径默认为获胜节点与最远节点距离的 2/3，见图 9-2，当网络结构为 1 × 3 时，进行 200 次迭代，得到终止迭代时的损失函数值为 0.933，这时各观测距离各自簇质心的距离为 0.0922，还是比较合理的。

```
> summary(irisdata.som)
SOM of size 1x3 with a rectangular topology and a bubble neighbourhood function.
The number of data layers is 1.
Distance measure(s) used: sumofsquares.
Training data included: 150 objects.
Mean distance to the closest unit in the map: 0.933.
> mean(irisdata.som$distances)# distance 各观测与各自簇质心的距离，距离越近，越合理。
[1] 0.9327065
```

图 9-2

各输出节点的连接权重和聚类类别所含样本个数如图 9-3 所示

```
> irisdata.som$code #各输出节点$的连接权重
[[1]]
      Sepal.Length Sepal.Width Petal.Length Petal.Width
v1 -0.02710202 -0.90837909   0.3617976   0.2600298
v2 -1.00944733  0.89181996  -1.3027051  -1.2545201
v3  1.09084286 -0.02581185   0.9645360   0.9362333

> table(irisdata.som$unit.classif) # unit.classif#各观测所属聚类类别编号
 1  2  3
52 50 48
```

图 9-3

3. 预览模型

(1) SOM 网络输出层示意图

图 9-4 是通过 mapping 可视化输出层，各输出节点以不同符号的点表示观测点与簇对应关系，是各簇内密度的粗略表现，可大致看出每个类的样本个数是差不多的。

SOM 网络输出层示意图

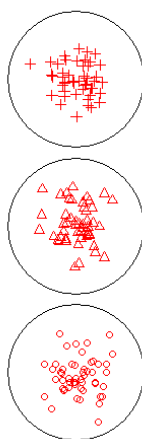


图 9-4

(2) SOM 网络聚类评价图

图 9-5 中显示迭代达 25 次后，输入与输出的类中心不再移动，说明迭代是充分的，各质心位置偏移的平均值稳定在 0.04 左右

SOM 网络聚类评价图

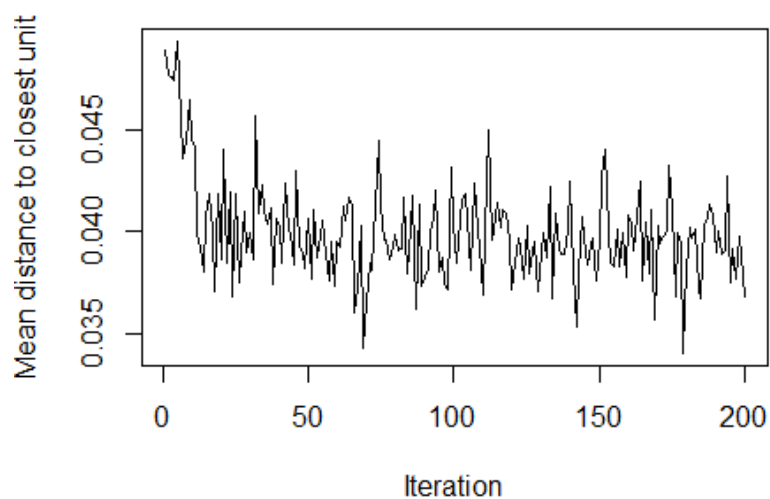


图 9-5

(3) SOM 聚类样本量分布情况

图 9-6 中颜色深浅表示簇所含样本的多少，其实三个样本之间的数量差异并不明显，第一类样本最多。

SOM聚类样本量分布情况

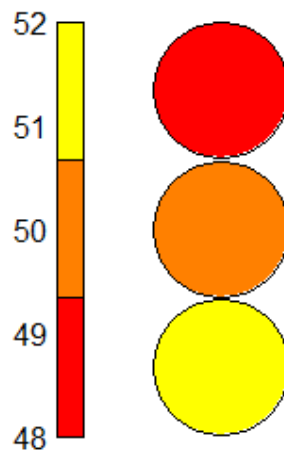


图 9-6

(4) SOM 聚类类内差异情况图

图 9-7 是比 mapping 的输出层可视化，更精确深浅表示簇内观测与质心距离的平均值大小。三个类的效果都较为理想，在 1 以下。

SOM聚类类内差异情况图

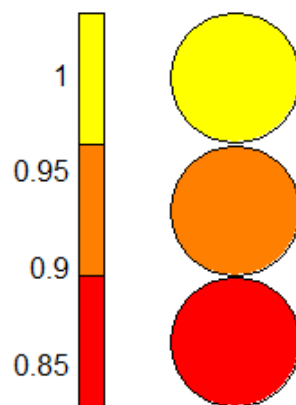


图 9-7

十、发现数据中的关联特征

10.1 实验目的

理论上，理解简单关联规则的含义，掌握 Apriori 算法的核心原理与基本实现思路，以及他的优缺点。

实践方面，掌握算法实现以及结果解读，解决实际问题中的关联规则。

10.2 数据来源与说明：

数据名称：购物篮数据，共有 1000 条记录，显示了这 1000 个人的年龄、性别以及都购买了一些什么物品。（见附件五）

数据属性说明：该数据集记录了信用卡 ID、购买者年龄、性别以购买的产品种类。

10.3 算法描述：

Apriori 算法包括以下两部分：

- （1） 搜索频繁项集，在生成候选项集时，需要设定支持度，频繁项集的产生需要依据算法的性质以及连接的规则
- （2） 5 依据频繁项集产生关联规则，在确定最大频繁项集后，需要写出他的非空子集，进行他们之间的两两组合，分别作为前件和后件，在此过程中还可以加入 Lift 限制，保证规则的有效性。

10.4 实验过程及结果分析：

1. 进行购物篮数据的导入，将其转换为矩阵形式，去掉前边年龄性别等属性列，只保留物品的信息
2. 将这些数据转变为 transaction 对象，进行汇总查看，可以看到每个商品

的购买。见图 10-1

```
> summary(data.trans)
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 fruitveg
2 freshmeat
3 dairy
> |
```

图 10-1

3. 设置支持度为 0.1，置信度为 0.5，进行规则的生成与筛选，图 10-2 显示了规则的置信度与支持度

```
data.trans      1000      0.1      0.5
> inspect(rules.apriori)
lhs      rhs      support confidence lift      count
[1] {confectionery} => {wine}      0.144 0.5217391 1.817906 144
[2] {wine} => {confectionery} 0.144 0.5017422 1.817906 144
[3] {beer} => {frozenmeal} 0.170 0.5802048 1.921208 170
[4] {frozenmeal} => {beer} 0.170 0.5629139 1.921208 170
[5] {beer} => {cannedveg} 0.167 0.5699659 1.881075 167
[6] {cannedveg} => {beer} 0.167 0.5511551 1.881075 167
[7] {frozenmeal} => {cannedveg} 0.173 0.5728477 1.890586 173
[8] {cannedveg} => {frozenmeal} 0.173 0.5709571 1.890586 173
[9] {frozenmeal,beer} => {cannedveg} 0.146 0.8588235 2.834401 146
[10] {cannedveg,beer} => {frozenmeal} 0.146 0.8742515 2.894873 146
[11] {cannedveg,frozenmeal} => {beer} 0.146 0.8439306 2.880309 146
```

图 10-2

4. 进行规则的排序，图 10-3 显示了按支持度的降序排列，10-4 显示了按 Lift 值降序排列，10-5 是精确匹配，显示特定规则


```
> inspect(sort(x=rules.apriori,by="support",decreasing=TRUE))#排序
  lhs      rhs      support confidence lift      count
[1] {frozenmeal} => {cannedveg} 0.173 0.5728477 1.890586 173
[2] {cannedveg} => {frozenmeal} 0.173 0.5709571 1.890586 173
[3] {beer}      => {frozenmeal} 0.170 0.5802048 1.921208 170
[4] {frozenmeal} => {beer}      0.170 0.5629139 1.921208 170
[5] {beer}      => {cannedveg} 0.167 0.5699659 1.881075 167
[6] {cannedveg} => {beer}      0.167 0.5511551 1.881075 167
[7] {frozenmeal,beer} => {cannedveg} 0.146 0.8588235 2.834401 146
[8] {cannedveg,beer} => {frozenmeal} 0.146 0.8742515 2.894873 146
[9] {cannedveg,frozenmeal} => {beer} 0.146 0.8439306 2.880309 146
[10] {confectionery} => {wine} 0.144 0.5217391 1.817906 144
[11] {wine}      => {confectionery} 0.144 0.5017422 1.817906 144
```

图 10-3

```
> inspect(sort(x=rules.apriori,by="lift",decreasing=TRUE))
  lhs      rhs      support confidence lift      count
[1] {cannedveg,beer} => {frozenmeal} 0.146 0.8742515 2.894873 146
[2] {cannedveg,frozenmeal} => {beer} 0.146 0.8439306 2.880309 146
[3] {frozenmeal,beer} => {cannedveg} 0.146 0.8588235 2.834401 146
[4] {beer}      => {frozenmeal} 0.170 0.5802048 1.921208 170
[5] {frozenmeal} => {beer}      0.170 0.5629139 1.921208 170
[6] {frozenmeal} => {cannedveg} 0.173 0.5728477 1.890586 173
[7] {cannedveg} => {frozenmeal} 0.173 0.5709571 1.890586 173
[8] {beer}      => {cannedveg} 0.167 0.5699659 1.881075 167
[9] {cannedveg} => {beer}      0.167 0.5511551 1.881075 167
[10] {confectionery} => {wine} 0.144 0.5217391 1.817906 144
[11] {wine}      => {confectionery} 0.144 0.5017422 1.817906 144
```

图 10-4

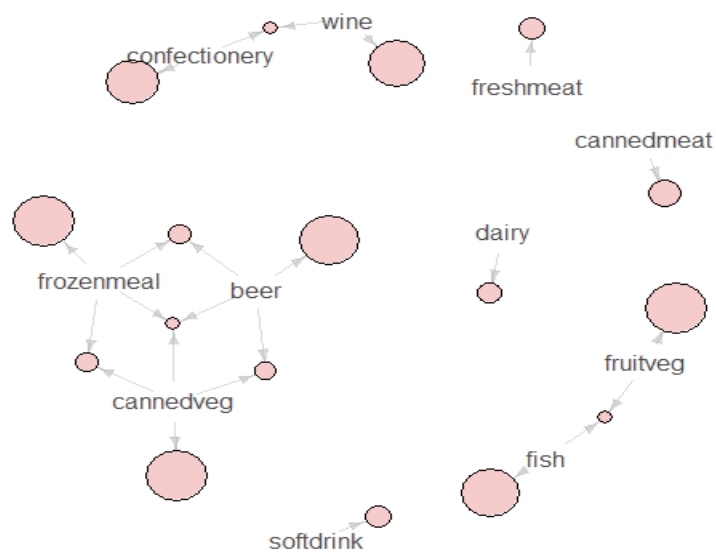
```
> inspect(subset(x=rules.apriori,subset=rhs%in%"beer"&lift>=2.2))
  lhs      rhs      support confidence lift      count
[1] {cannedveg,frozenmeal} => {beer} 0.146 0.8439306 2.880309 146
> inspect(subset(x=rules.apriori,subset=rhs%in%"beer"&lift>=2.2))
  lhs      rhs      support confidence lift      count
[1] {cannedveg,frozenmeal} => {beer} 0.146 0.8439306 2.880309 146
> inspect(subset(x=rules.apriori,subset=size(rules.apriori)==2))
  lhs      rhs      support confidence lift      count
[1] {confectionery} => {wine} 0.144 0.5217391 1.817906 144
[2] {wine}      => {confectionery} 0.144 0.5017422 1.817906 144
[3] {beer}      => {frozenmeal} 0.170 0.5802048 1.921208 170
[4] {frozenmeal} => {beer}      0.170 0.5629139 1.921208 170
[5] {beer}      => {cannedveg} 0.167 0.5699659 1.881075 167
[6] {cannedveg} => {beer}      0.167 0.5511551 1.881075 167
[7] {frozenmeal} => {cannedveg} 0.173 0.5728477 1.890586 173
[8] {cannedveg} => {frozenmeal} 0.173 0.5709571 1.890586 173
```

图 10-5

- 可视化频繁项集与规则。图 10-6 显示了支持度为 0.1，置信度为 0.5 时的频繁项集，图 10-7 和 10-8 是相应的规则可视化。

频繁项集可视化

size: support (0.144 - 0.303)



关联规则可视化

size: support (0.144 - 0.173)
color: lift (1.818 - 2.895)

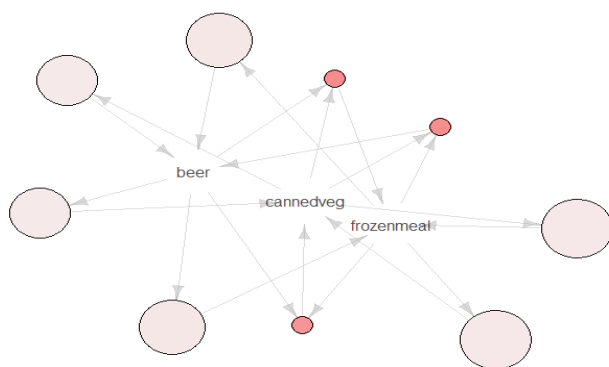
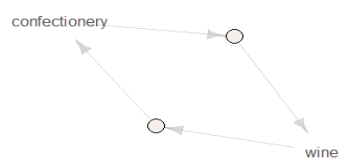


图 10-7

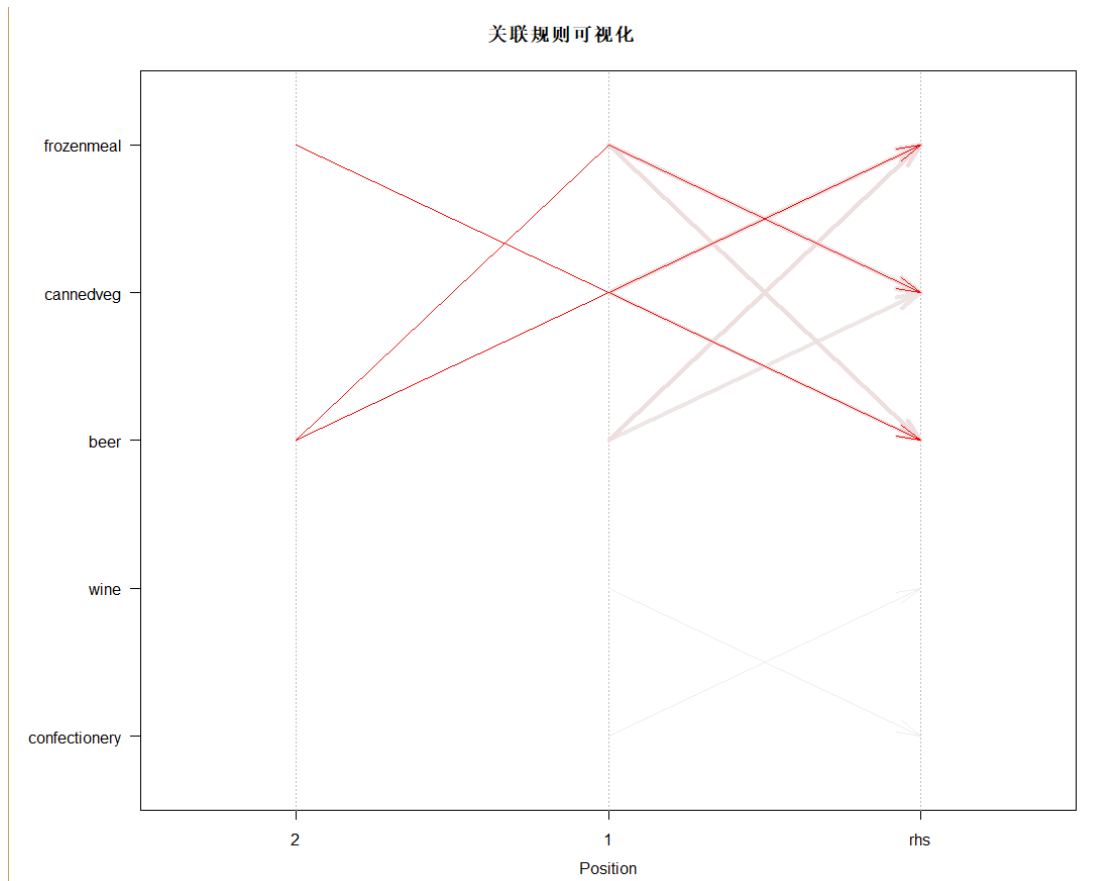


图 10-8

6. 之后是对顾客的选择倾向进行分析，比较不同年龄与性别的顾客购买啤酒的可能性大小，所以将性别、年龄与啤酒购买提取出来。如图 10-9.

```
> data.rule
      sex age beer
1      M  46   0
2      F  28   0
3      M  36   1
4      F  26   0
5      M  24   0
6      F  35   0
7      F  30   0
8      M  22   1
9      F  46   0
10     M  22   0
11     F  18   0
12     F  48   0
13     M  43   0
14     F  43   0
15     F  24   0
```

图 10-9

7. 将数据转换为 transaction 对象，设置支持度为 0.01，置信度为 0.2，进

行规则的生成，如图 10-10，显示了规则的置信度与支持度

```
creating 34 object ... done [0.003].
> inspect(rules)
```

	lhs	rhs	support	confidence	lift	count
[1]	{age=1}	=> {beer=1}	0.124	0.2890443	0.9864993	124
[2]	{sex=M}	=> {beer=1}	0.196	0.4016393	1.3707827	196
[3]	{age=2}	=> {beer=1}	0.164	0.3009174	1.0270219	164
[4]	{sex=M,age=1}	=> {beer=1}	0.086	0.3981481	1.3588674	86
[5]	{sex=F,age=2}	=> {beer=1}	0.058	0.2049470	0.6994778	58
[6]	{sex=M,age=2}	=> {beer=1}	0.106	0.4045802	1.3808196	106

```
> |
```

图 10-10

8. 选择出 LIFT 值大于 1 的规则进行可视化。如图 10-11

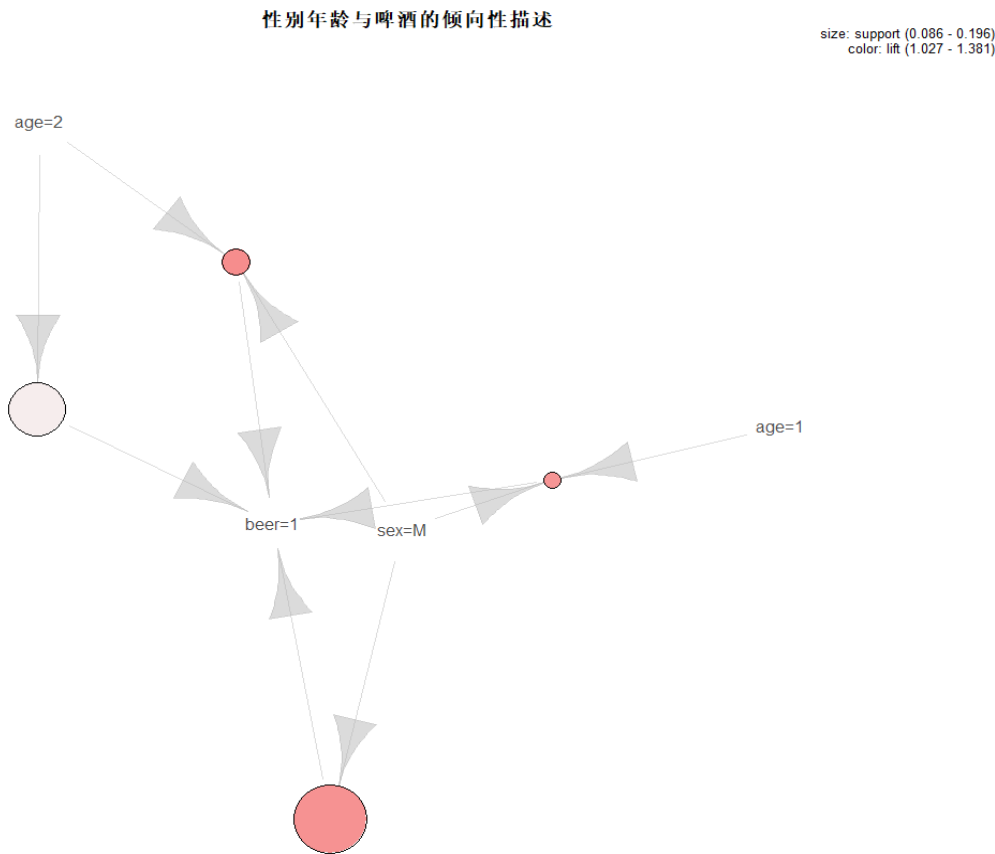


图 10-11

十一、总结

大数据不仅意味着数据的积累、存储与管理，更意味着大数据的分析。数据挖掘无可争议地成为当今大数据分析的核心利器。R 语言因彻底的开放性策略业已跻身数据挖掘工具之首列。

通过本次实验，我们了解了数据挖掘的理论和应用轮廓，明确了 R 语言入门的必备知识和学习路线，并展示了数据挖掘的初步成果。经过接近一个学期的学习，从对 R 语言的完全陌生，到现在对其有了一些粗浅的认识，其中经历了遇到困难苦思冥想的艰辛，也有解决问题以后豁然开朗的畅快。在学习的过程中，以前掌握的编程基础给我们带来了不少便利，而认真地态度和踏实的性格也使我们获益匪浅。围绕数据挖掘应用的核心方面，通过决策树、随机森林、KNN、ANN、贝叶斯分类、SVM、聚类、关联规则的一系列实践应用，更深刻地理解了理论知识和基本原理。在这个学期中，我学会了 R 语言的基本操作和语法，以及针对具体的信息分析问题相应的解决方法。并按时完成老师布置的课后作业，以达到学以致用目的，也加强了对 R 语言操作的熟练度。在今后的学习生活中，能够掌握初步通过 R 语言进行数据挖掘和信息分析与预测。

附件:

一、 输血服务中心数据集

Recency	Frequency	Monetary	Time	DonatedBloodInMarch2007
2	50	12500	98	1
0	13	3250	28	1
1	16	4000	35	1
2	20	5000	45	1
1	24	6000	77	0
4	4	1000	4	0
2	7	1750	14	1
1	12	3000	35	0
2	9	2250	22	1
5	46	11500	98	1
4	23	5750	58	0
0	3	750	4	0
2	10	2500	28	1
1	13	3250	47	0
2	6	1500	15	1
2	5	1250	11	1
2	14	3500	48	1

二、 鸢尾植物数据库

SepalLength,SepalWidth,PetalLength,PetalWidth,IrisPlantClass

5.1,3.5,1.4,0.2,Iris-setosa

4.9,3.0,1.4,0.2,Iris-setosa

4.7,3.2,1.3,0.2,Iris-setosa

4.6,3.1,1.5,0.2,Iris-setosa

5.0,3.6,1.4,0.2,Iris-setosa

5.4,3.9,1.7,0.4,Iris-setosa

4.6,3.4,1.4,0.3,Iris-setosa

5.0,3.4,1.5,0.2,Iris-setosa

4.4,2.9,1.4,0.2,Iris-setosa

4.9,3.1,1.5,0.1,Iris-setosa

5.4,3.7,1.5,0.2,Iris-setosa

4.8,3.4,1.6,0.2,Iris-setosa

4.8,3.0,1.4,0.1,Iris-setosa

4.3,3.0,1.1,0.1,Iris-setosa

5.8,4.0,1.2,0.2,Iris-setosa

5.7,4.4,1.5,0.4,Iris-setosa

5.4,3.9,1.3,0.4,Iris-setosa

5.1,3.5,1.4,0.3,Iris-setosa

5.7,3.8,1.7,0.3,Iris-setosa

三、 Statlog（德国信用数据）数据集

account _check_ status	duratio n_in_m onth	credit_history	purpos e	credi t_am ount	savings	presen t_emp_ since
< 0 DM	6	critical account/ other credits existing (not at this bank)	domest ic applian ces	1169	unknown/ no savings account	.. >= 7 years
0 <= ... < 200 DM	48	existing credits paid back duly till now	domest ic applian ces	5951	... < 100 DM	1 <= ... < 4 years

installment _as_income _perc	personal_stat us_sex	other _debt ors	presen t_res_s ince	pro per ty	a g e	other_ins tallment_ plans	ho usi ng	credits _this_b ank
4	male : single	none	4	real esta te	6 7	none	ow n	2
2	female : divorced/sepa rated/married	none	2	real esta te	2 2	none	ow n	1

job	people_under_mai ntenance	telephone	foreign_w orker
skilled employee / official	1	yes, registered under the customers name	yes
skilled employee / official	1	none	yes

四、 SMS Spam Collection

type	text								
type	text								
ham	Lol your always so convincing.								
spam	SMS. ac Sptv: The New Jersey Devils and the Detroit Red Wings play Ice Hockey. Correct or Inc								
spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive								
spam	Valentines Day Special! Win over 撙 1000 in our quiz and take your partner on the trip of a life								
ham	Nah I don't think he goes to usf, he lives around here though								
spam	FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up								
ham	Even my brother is not like to speak with me. They treat me like aids patent.								
ham	As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as you								
spam	WINNER!! As a valued network customer you have been selected to receivea 撙 900 prize rew								
spam	Had your mobile 11 months or more? U R entitled to Update to the latest colour mobiles with c								
ham	I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried								
spam	SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 87575. Cost 150								
spam	URGENT! You have won a 1 week FREE membership in our 撙 100,000 Prize Jackpot! Txt the w								
ham	I've been searching for the right words to thank you for this breather. I promise i wont take you								
ham	I HAVE A DATE ON SUNDAY WITH WILL!!								
spam	XXXMobileMovieClub: To use your credit, click the WAP link in the next txt message or click he								
ham	Oh k...i'm watching here:)								
ham	Eh u remember how 2 spell his name... Yes i did. He v naughty make until i v wet.								
ham	Fine if that's the way u feel. That's the way its gota b								
spam	England v Macedonia - dont miss the goals/team news. Txt ur national team to 87077 eg ENGL								
ham	Is that seriously how you spell his name?								
ham	Iam going to try for 2 months ha ha only joking								
spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive								
ham	Aft i finish my lunch then i go str down lor. Ard 3 smth lor. U finish ur lunch already?								
ham	Lol your always so convincing.								
spam	SMS. ac Sptv: The New Jersey Devils and the Detroit Red Wings play Ice Hockey. Correct or Inc								
spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive								
spam	Valentines Day Special! Win over 撙 1000 in our quiz and take your partner on the trip of a life								
ham	Nah I don't think he goes to usf, he lives around here though								
spam	FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up								
ham	Even my brother is not like to speak with me. They treat me like aids patent.								

五、 购物篮数据

cardid,value,pmethod,sex,homeown,income,age,fruitveg,freshmeat,dairy,canned

veg,cannedmeat,frozenmeal,beer,wine,softdrink,fish,confectionery

39808,42.7123,CHEQUE,M,NO,27000,46,0,1,1,0,0,0,0,0,0,0,1

67362,25.3567,CASH,F,NO,30000,28,0,1,0,0,0,0,0,0,0,0,1

10872,20.6176,CASH,M,NO,13200,36,0,0,0,1,0,1,1,0,0,1,0

26748,23.6883,CARD,F,NO,12200,26,0,0,1,0,0,0,0,1,0,0,0

91609,18.8133,CARD,M,YES,11000,24,0,0,0,0,0,0,0,0,0,0,0

26630,46.4867,CARD,F,NO,15000,35,0,1,0,0,0,0,0,1,0,1,0

62995,14.0467,CASH,F,YES,20800,30,1,0,0,0,0,0,0,0,1,0,0

38765,22.2034,CASH,M,YES,24400,22,0,0,0,0,0,0,0,1,0,0,0

28935,22.975,CHEQUE,F,NO,29500,46,1,0,0,0,0,1,0,0,0,0,0

41792,14.5692,CASH,M,NO,29600,22,1,0,0,0,0,0,0,0,0,1,0

59480,10.3282,CASH,F,NO,27100,18,1,1,1,1,0,0,0,1,0,1,0

60755,13.7796,CASH,F,YES,20000,48,1,0,0,0,0,0,0,0,0,1,0

70998,36.509,CARD,M,YES,27300,43,0,0,1,0,1,1,0,0,0,1,0

80617,10.2011,CHEQUE,F,YES,28000,43,0,0,0,0,0,0,0,0,1,1,0

61144,10.3736,CASH,F,NO,27400,24,1,0,1,0,0,0,0,0,1,1,0

36405,34.8222,CHEQUE,F,YES,18400,19,0,0,0,0,0,1,1,0,1,0,0

76567,42.248,CARD,M,YES,23100,31,1,0,0,1,0,0,0,0,0,1,0

85699,18.1688,CASH,F,YES,27000,29,0,0,0,0,0,0,0,0,0,1,0

