第三节课 有监督学习

李毅

山西财经大学 统计学院

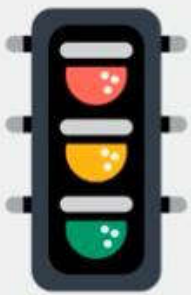
目录

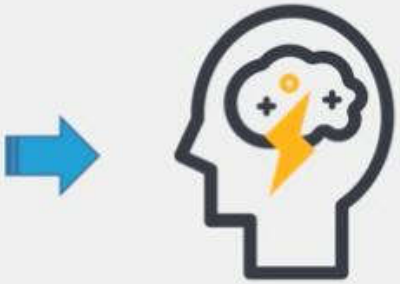
一、有监督学习概论
二、最小二乘线性回归
三、Logistic回归
四、决策树及其组合方法
五、支持向量机
六、人工神经网络
七、朴素贝叶斯
八、K最近邻方法
九、有监督学习模型比较案例

一、有监督学习概论

1.1“学习”的概念

学习：人在生活过程中，通过获得经验而产生的行为或行为潜能的相对持久的行为方式





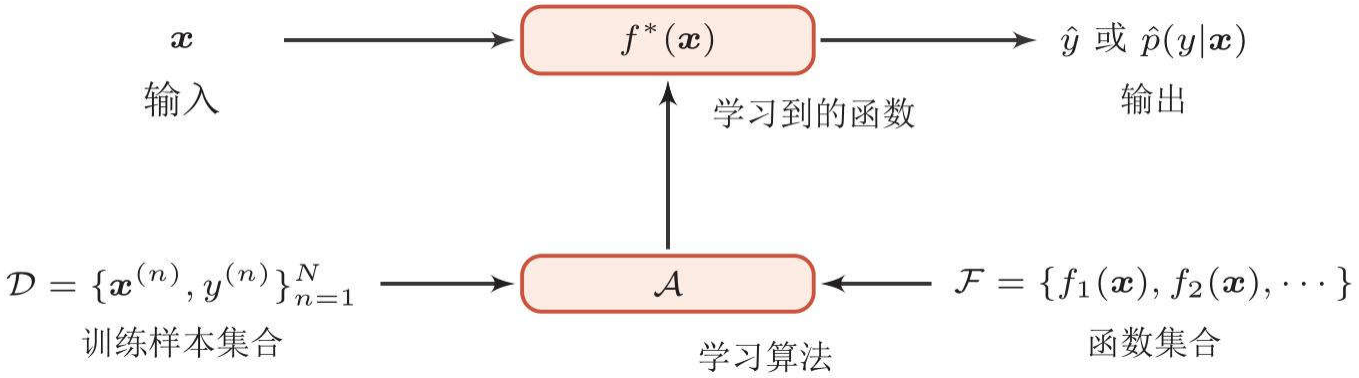
红
绿
灯

环境 → 人脑学习 → 经验

机器学习：计算机模拟人类的学习行为，以获取新的知识或技能，重新组织已有的知识结构不断改善自身的性能

“假设用P来评估计算机程序在某任务类T上的性能，若一个程序通过利用经验E在T中任务上获得了性能改善，则我们就说关于T和P，该程序对E进行了学习”

机器学习致力于研究如何通过计算的手段，利用经验来改善系统自身的性能，从而在计算机上从数据中产生“模型”，用于对新的情况给出判断。



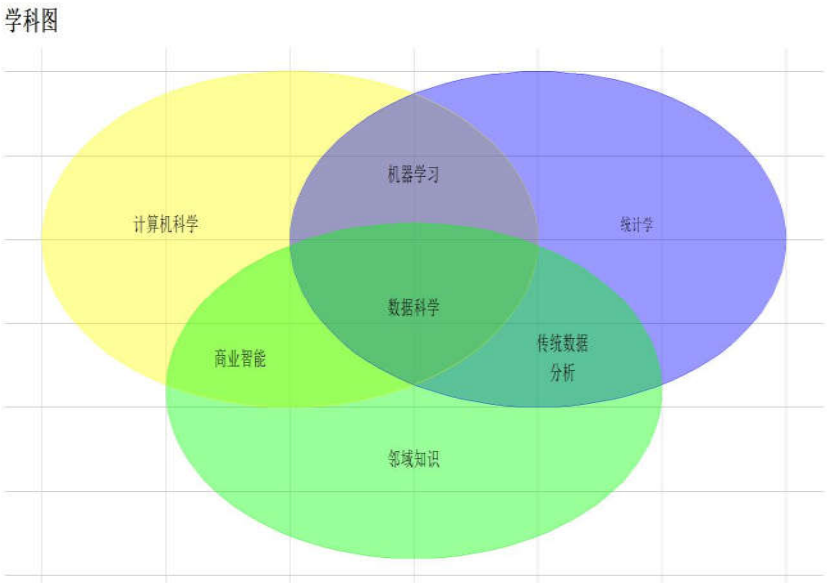
独立同分布 p(x,y)

机器学习的优势

- 能够处理大量数据，远远超过人工的处理能力
- 解决无法通过清晰地定义的问题
- 借助机器在短时间内做出快速、清晰的判断
- 为海量用户或者使用提供个性化服务
- 对问题的判断不受环境干扰
- .....

维基百科：

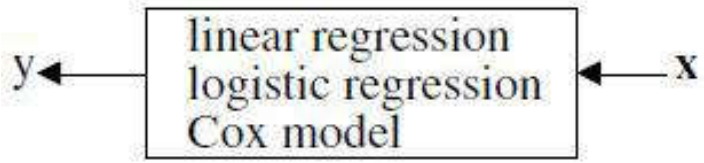
机器学习是近20多年兴起的一门**多领域交叉学科，涉及概率论、统计学、逼近论、凸分析、算法复杂度理论**等多门学科。机器学习理论主要是设计和分析一些让计算机可以**自动“学习”**的算法。**机器学习算法**是一类从数据中自动分析获得规律，并利用规律对未知数据进行预测的算法。因为学习算法中涉及了大量的统计学理论，**机器学习与统计推断学联系尤为密切，也被称为统计学习理论。**算法设计方面，机器学习理论关注可以实现的，行之有效的学习算法。很多推论问题属于无程序可循难度，所以部分的机器学习研究是开发容易处理的近似算法。



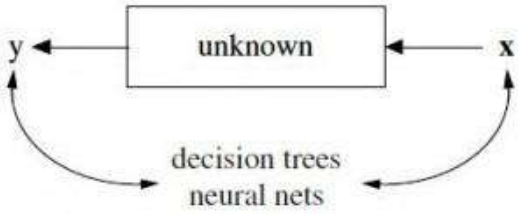
论文：STATISTICAL MODELING: THE TWO CULTURES

Leo Breiman大神2001年发表在Statistical Science上的一篇老文，他将统计科学分为两个分支：**Data Modeling**和**Algorithm Modeling**。

Data Model直接假设数据服从一定的分布和随机噪声，数据均是由这些分布产生。这种模型的框架图为：



而Algorithm Modeling认为框架内部非常复杂，他们只是寻找一个函数*f(x)*，用*x*做输入来预测*y*。其框架图如下：



文章认为统计学过分依赖了Data Modeling，而机器学习主要依赖模型预测精度（*predictive accuracy* of models），从而取得了更多进步。

Brendan O'Connor的博文Statistics vs. Machine Learning, fight!

初稿是08年写的，或许和作者的机器学习背景有关，他在初稿中主要是贬低了统计学，思想和[1]有点类似，认为机器学习比统计学多了些Algorithm Modeling方面内容，比如SVM的Max-margin，决策树等，此外他认为机器学习更偏实际。但09年十月的时候他转而放弃自己原来的观点，认为统计才是real deal: Statistics, not machine learning, is the real deal, but unfortunately suffers from bad marketing。

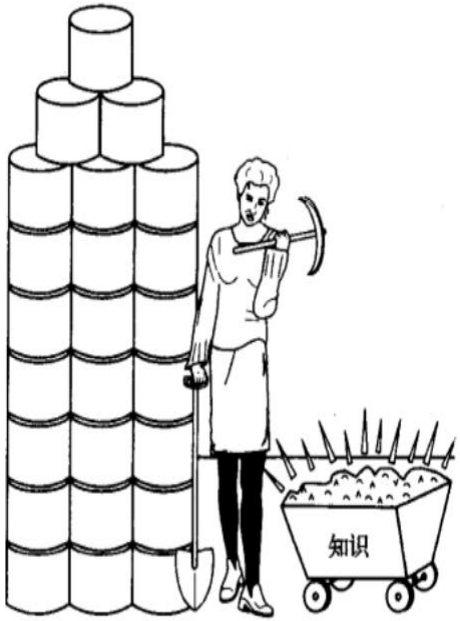
他的博文中还引用了大神Robert Tibshiriani的一张对比表：

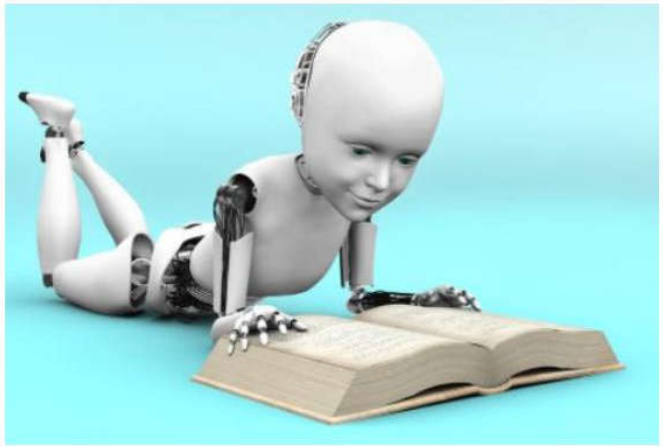
Glossary

Machine learning	Statistics
network, graphs	model
weights	parameters
learning	fitting
generalization	test set performance
supervised learning	regression/classi?cation
unsupervised learning	density estimation, clustering
large grant = $1,000,000	large grant = $50,000
nice place to have a meeting: Snowbird, Utah, French Alps	nice place to have a meeting: Las Vegas in August

机器学习的任务

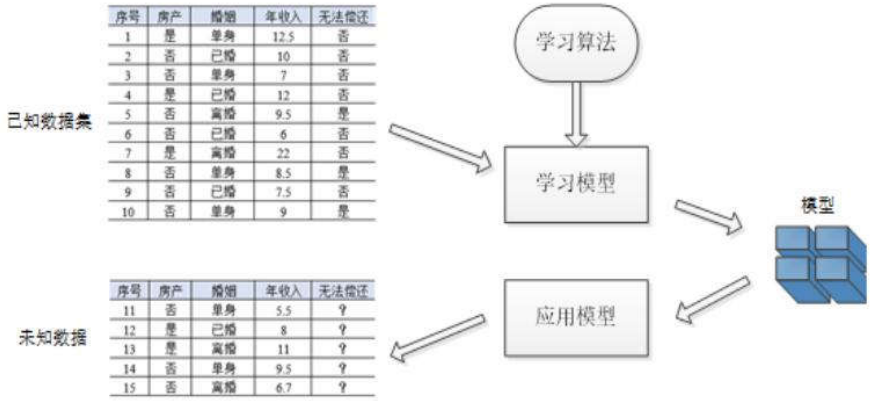
- 机器学习方法利用既有的经验，完成某种既定任务，且在此过程不断改善自身性能
- 按照机器学习的任务分为两大类方法
  - 有监督的学习（Supervised Learning）
  - 无监督的学习（Unsupervised Learning）





有监督学习

- 利用经验（数据），学习表示事物的模型，关注利用模型预测未来数据
  - 分类问题（Classification）
    - 对事物所属类型的判别，类别的数量是已知的
    - 例：鸟类型识别，垃圾邮件分类
  - 回归问题（Regression）
    - 预测目标是连续变量
    - 例：根据父母身高预测孩子身高



序号	房产	婚姻	年收入	无法偿还
1	是	单身	12.5	否
2	否	已婚	10	否
3	否	单身	7	否
4	是	已婚	12	否
5	否	离婚	9.5	是
6	否	已婚	6	否
7	是	离婚	22	否
8	否	单身	8.5	是
9	否	已婚	7.5	否
10	否	单身	9	是

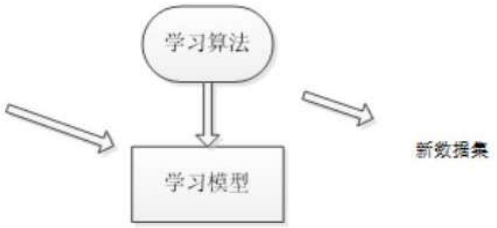
序号	房产	婚姻	年收入	无法偿还
11	否	单身	5.5	?
12	是	已婚	8	?
13	是	离婚	11	?
14	否	单身	9.5	?
15	否	离婚	6.7	?

无监督学习

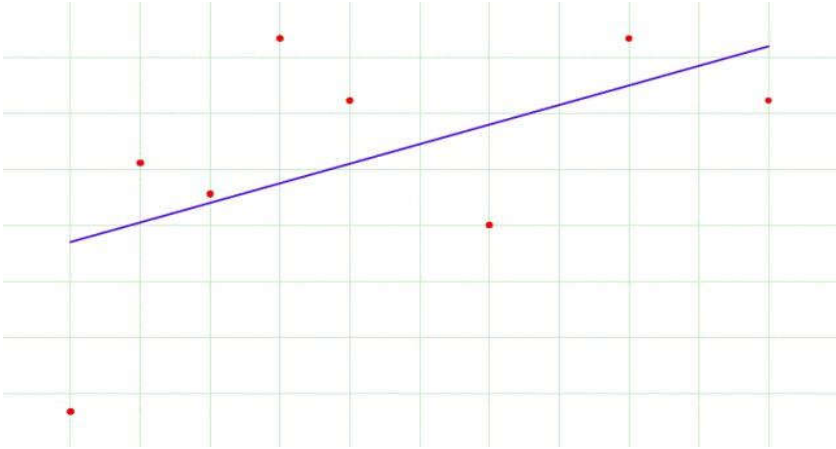
- 倾向于对事物本身特性的分析，常见问题包括
  - 数据降维（Dimensionality Reduction）
    - 对描述事物的特征数量进行压缩的方法
    - 例：从已有的100个特征中选取部分特征表示音乐信号
  - 聚类问题（Clustering）
    - 将事物划分成不同的类别，但事先不知道类别的数量，根据事物之间的相似性，将相似的事物归为一簇
    - 例：电子商务网站将具有类似背景与购买习惯的用户自动聚为一类

已知数据集

序号	房产	婚姻	年收入	无法偿还
1	是	单身	12.5	否
2	否	已婚	10	否
3	否	单身	7	否
4	是	已婚	12	否
5	否	离婚	9.5	是
6	否	已婚	6	否
7	是	离婚	22	否
8	否	单身	8.5	是
9	否	已婚	7.5	否
10	否	单身	9	是



1.2 模型和拟合



- 模型（model）对所关心正式世界问题的一个近似描述。
- 拟合（fit）：通过数据来训练模型的行为。

编号	色泽	根蒂	敲声	好瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	沉闷	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否
1	青绿	蜷缩	沉闷	?

训练集←

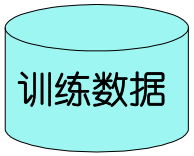
测试集←

过拟合与欠拟合的判断标准

训练集上的表现	测试集上的表现	结论
不好	不好	欠拟合
好	不好	过拟合
好	好	适度拟合

典型的机器学习过程

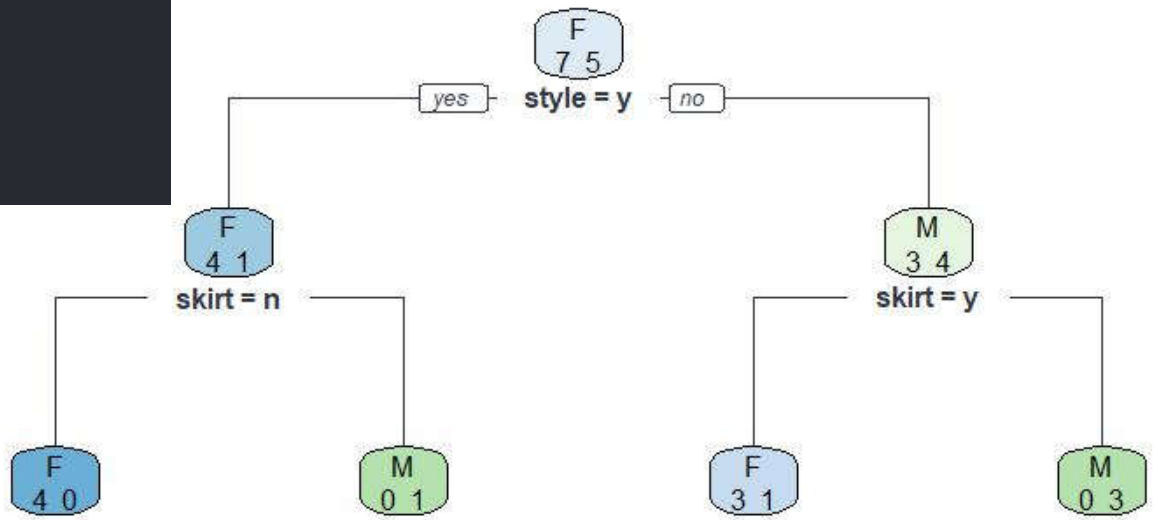
【案例】训练决策树模型分类：



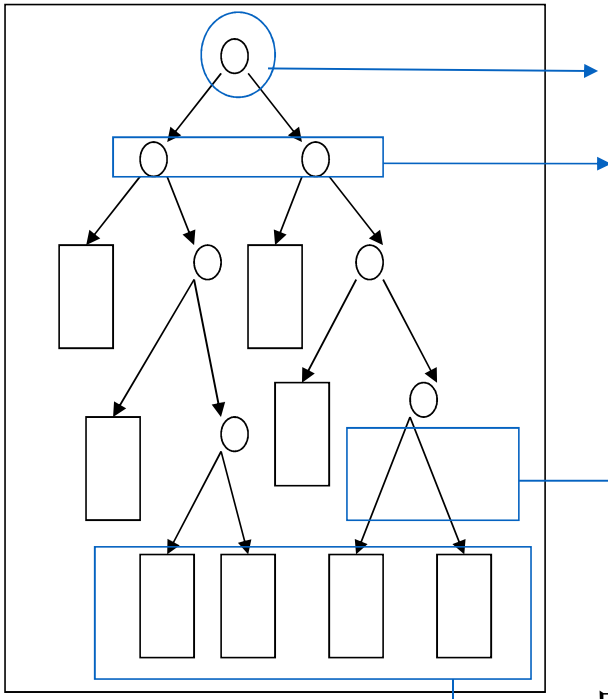
	sex	haircol	skirt	style
1	F	yellow	y	n
2	F	black	y	n
3	F	yellow	n	y
4	F	black	n	y
5	M	yellow	n	n
6	M	black	n	n
7	M	yellow	y	n
8	F	black	n	y
9	M	yellow	n	n
10	F	black	y	n
11	M	black	y	y
12	F	yellow	n	y

结果模型

```
> library(rpart.plot)
载入需要的程辑包：rpart
> a=rpart(sex~.,df,minsplit=2,model = TRUE) #产生决策树
> rpart.plot(a,extra = 1) #画图
> a #输出放在a中的决策树细节
n= 12

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

1) root 12 5 F (0.5833333 0.4166667)
  2) style=y 5 1 F (0.8000000 0.2000000)
    4) skirt=n 4 0 F (1.0000000 0.0000000) *
    5) skirt=y 1 0 M (0.0000000 1.0000000) *
  3) style=n 7 3 M (0.4285714 0.5714286)
    6) skirt=y 4 1 F (0.7500000 0.2500000) *
    7) skirt=n 3 0 M (0.0000000 1.0000000) *
```



决策树的结构



根部节点(root node)

中间节点(non-leaf node)

(代表测试的条件)

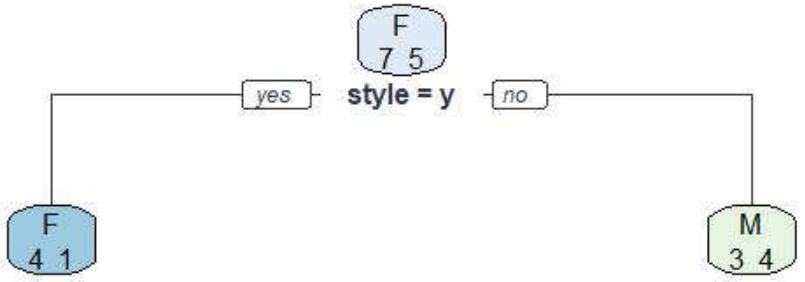
分支(branches)

(代表测试的结果)

叶节点(leaf node)

(代表分类后所获得的分类标记)

决策树的直观过程：



1、根节点（1号节点，全部12个观测值：7个F及5个M）拆分变量的竞争

（1）考虑haircol（头发颜色），用代码：

```
> table(df[,c(1,2)])
   haircol
sex black yellow
  F     4      3
  M     2      3
```

误判：5个

（2）考虑skirt（是否穿裙子），用代码：

```
> table(df[,c(1,3)])
   skirt
sex n y
  F 4 3
  M 3 2
```

误判：5个

（3）考虑styple（是否有某种走路姿势），用代码：

```
> table(df[,c(1,4)])
   style
sex n y
  F 3 4
  M 4 1
```

误判：4个

用haircol为 2个误判;
用skirt为1个误判

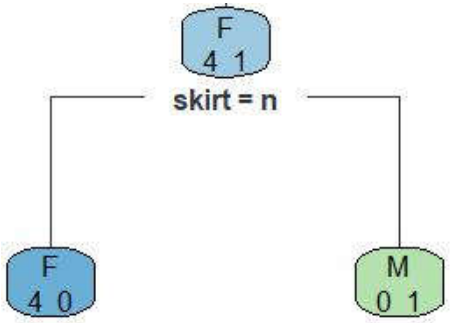
2、左下节点（命名2号节点，全部5个观测值：4个F及1个M）拆分变量的竞争

（1）考虑haircol（头发颜色），用代码：

```
> table(df[df[,4]=="y",][,c(1,2)])
   haircol
sex black yellow
  F     2      2
  M     1      0
```

（2）考虑skirt（是否穿裙子），用代码：

```
> table(df[df[,4]=="y",][,c(1,3)])
   skirt
sex n y
  F 4 0
  M 0 1
```



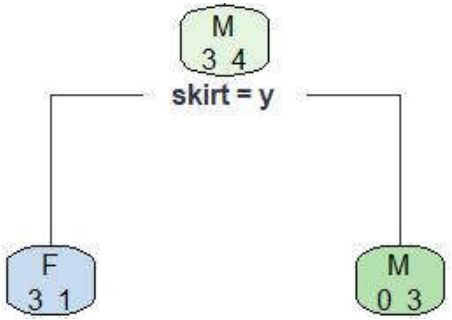
3、右下节点（命名3号节点，全部7个观测值：3个F及4个M）拆分变量的竞争

（1）考虑haircol（头发颜色），用代码：

```
> table(df[df[,4]=="n",][,c(1,2)])
   haircol
sex black yellow
  F     2      1
  M     1      3
```

（2）考虑skirt（是否穿裙子），用代码：

```
> table(df[df[,4]=="n",][,c(1,3)])
   skirt
sex n y
  F 0 3
  M 3 1
```



决策树的实际计算过程——利用Gini不纯度和信息增益

Gini不纯度(R中的函数rpart的默认方法)

$$\text{Gini}(D) = \sum_{k=1}^{|\mathcal{Y}|} \sum_{k' \neq k} p_k p_{k'} \quad = 1 - \sum_{k=1}^{|\mathcal{Y}|} p_k^2$$

1. 对于styple=“y”，“F”和“M”分别是4和1，因此Gini不纯度为$1-(1/5)^2-(4/5)^2=0.32$。
2. 对于styple=“n”，“F”和“M”分别是3和4，因此Gini不纯度为$1-(3/7)^2-(4/7)^2=0.4898$。
3. 把这个子节点的Gini不纯度按照观测值数目加权平均，得到
   $0.32*5/(5+7)+0.4898*7/(5+7)=0.4190$

CART [Breiman et al., 1984]采用“基尼指数”来选择划分属性

信息增益

- “信息熵”是度量样本集合纯度最常用的一种指标，假定当前样本集合D中第k类样本所占的比例为$p_k(K=1,2,...,|\mathcal{Y}|)$，则$D$的信息熵定义为

$$\mathrm{Ent}(D)=-\sum_{k=1}^{|\mathcal{Y}|}p_k\log_2 p_k$$

$\mathrm{Ent}(D)$的值越小，则D的纯度越高

- 计算信息熵时约定：若$p=0$，则$p\log_2 p=0$
- $\mathrm{Ent}(D)$的最小值为0，最大值为$\log_2|\mathcal{Y}|$

信息增益

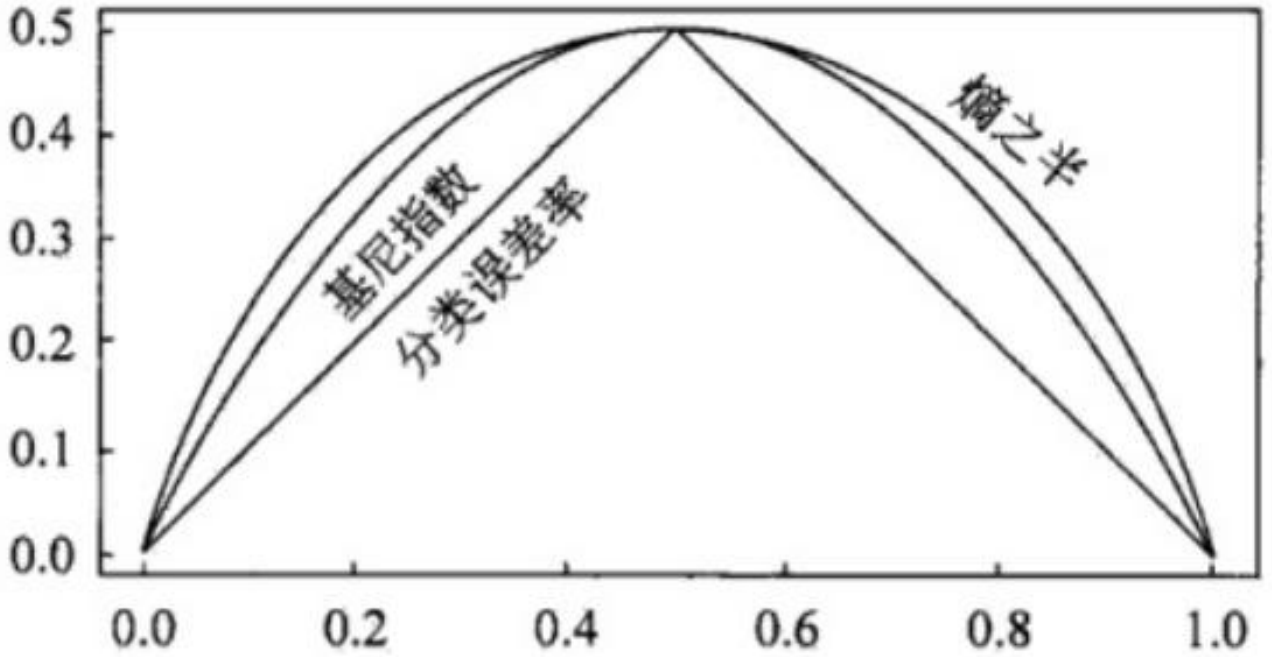
- 离散属性 a 有 V 个可能的取值 $\{a^1, a^2, ..., a^V\}$，用 a 来进行划分，则会产生 a 个分支结点，其中第 V 个分支结点包含了 v 中所有在属性 D 上取值为 a^v 的样本，记为 D^v。则可计算出用属性 d 对样本集 D 进行划分所获得的“信息增益”：

$$\mathrm{Gain}(D,a) = \mathrm{Ent}(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} \mathrm{Ent}(D^v)$$

为分支结点权重，样本数越多的分支结点的影响越大

- 一般而言，信息增益越大，则意味着使用属性 a 来进行划分所获得的“纯度提升”越大

□ ID3决策树学习算法[Quinlan, 1986]以信息增益为准则来选择划分属性



横坐标表示概率p，纵坐标表示损失，从上图可以看出基尼指数和熵之半的曲线很接近，都可以近似分类误差率。

预测值，拟合值，误判率(预判精度)

将训练得到的模型应用于一个数据，那么会得到**预测值**。
对于训练模型的训练集做预测所得到预测值也称为**拟合值**。

```
> predict(a,df,type = "class")
 1  2  3  4  5  6  7  8  9 10 11 12
 F  F  F  F  M  M  F  F  M  F  M  F
Levels: F M
```

误判率为1/12=0.0833。

```
> table(df$sex,predict(a,df,type = "class")) #混淆矩阵

   F M
 F 7 0
 M 1 4
> (miss=sum(df$sex!=predict(a,df,type = "class")))#误判个数
[1] 1
> miss/nrow(df)#误判率nrow(df)是数据df的行数(样本量)
[1] 0.08333333
```

```
> df1=read.csv("new4.csv")
> table(df1$sex,predict(a,df1,type = "class")) #混淆矩阵

   F M
 F 6 5
 M 2 7
> sum(df1$sex!=predict(a,df1,type = "class"))/nrow(df1)
[1] 0.35
```

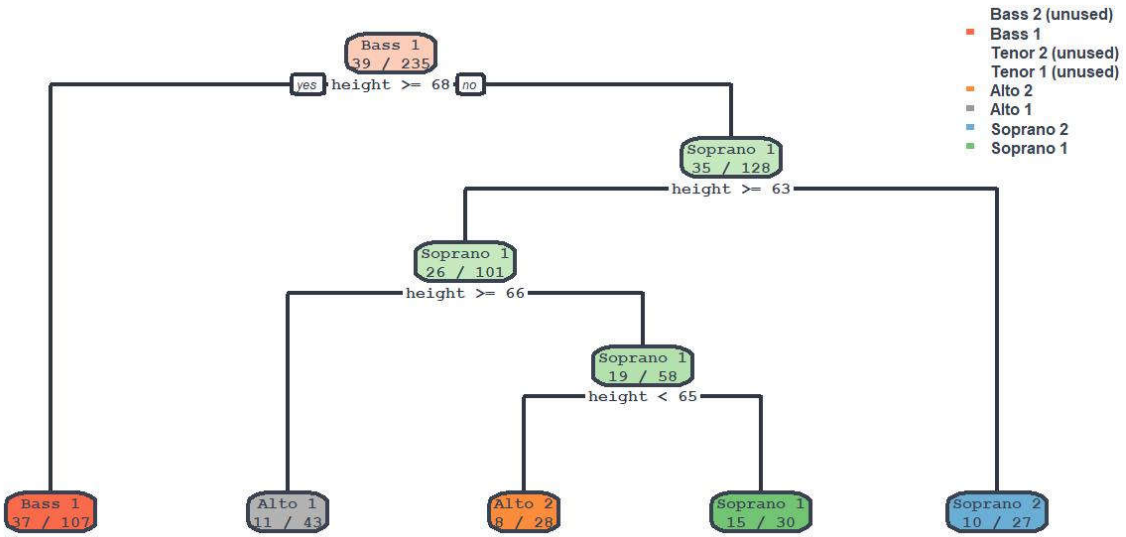
在20个观测值中有7个误判的，误判率为0.35，比对训练集预测的误判率要高4.1倍。

如果用训练集所得到的误判率和用非训练集（或测试集）得到的误判率之间差别很大，则说明该模型有过拟合现象（overfit）。也就是说该模型没有普遍意义，只适合于训练该模型的训练集。

【案例】R中程序包lattice的数据singer

该数据记录了1979年纽约合唱协会的歌手们的身高(变量height)及语音分组(变量voice.part),变量 height的单位为英寸(inch),而变量voice.part的单位(从低音到高音)为:Bass2、bass1、Tenor2、Tenor1、Alto2、Alto1、 Soprano2、Soprano1等8种。这里试图用变量 voice.part作为因变量,变量 height为自变量做分类。

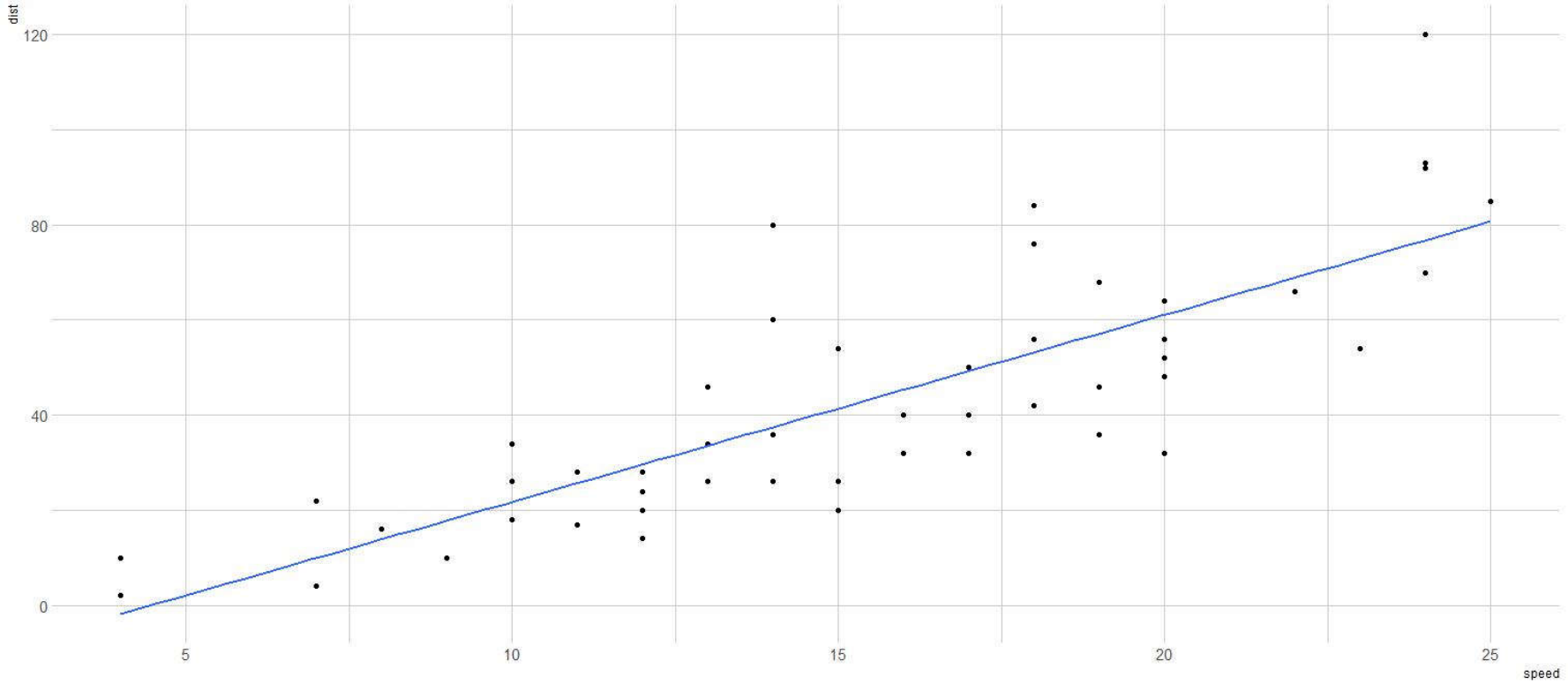
```
> library(rpart.plot)
> library(lattice)
> (a=rpart(voice.part~height,singer))
n= 235

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

 1) root 235 196 Bass 1 (0.11 0.17 0.089 0.089 0.11 0.15 0.13 0.15)
   2) height>=67.5 107  70 Bass 1 (0.21 0.35 0.19 0.12 0.056 0.047 0.019 0.0093) *
   3) height< 67.5 128  93 Soprano 1 (0.023 0.016 0.0078 0.062 0.16 0.23 0.22 0.27)
     6) height>=62.5 101  75 Soprano 1 (0.03 0.02 0.0099 0.079 0.21 0.22 0.18 0.26)
      12) height>=65.5 43  32 Alto 1 (0.07 0.047 0.023 0.12 0.19 0.26 0.14 0.16) *
      13) height< 65.5 58  39 Soprano 1 (0 0 0 0.052 0.22 0.19 0.21 0.33)
        26) height< 64.5 28  20 Alto 2 (0 0 0 0.071 0.29 0.25 0.25 0.14) *
        27) height>=64.5 30  15 Soprano 1 (0 0 0 0.033 0.17 0.13 0.17 0.5) *
     7) height< 62.5 27  17 Soprano 2 (0 0 0 0 0 0.3 0.37 0.33) *
> rpart.plot(a,type=2,extra=2)
> mean(singer[,2]!=predict(a,singer,type = "class"))#训练集误判率
[1] 0.6553191
```



【案例】训练简单最小二乘线性回归模型的例子

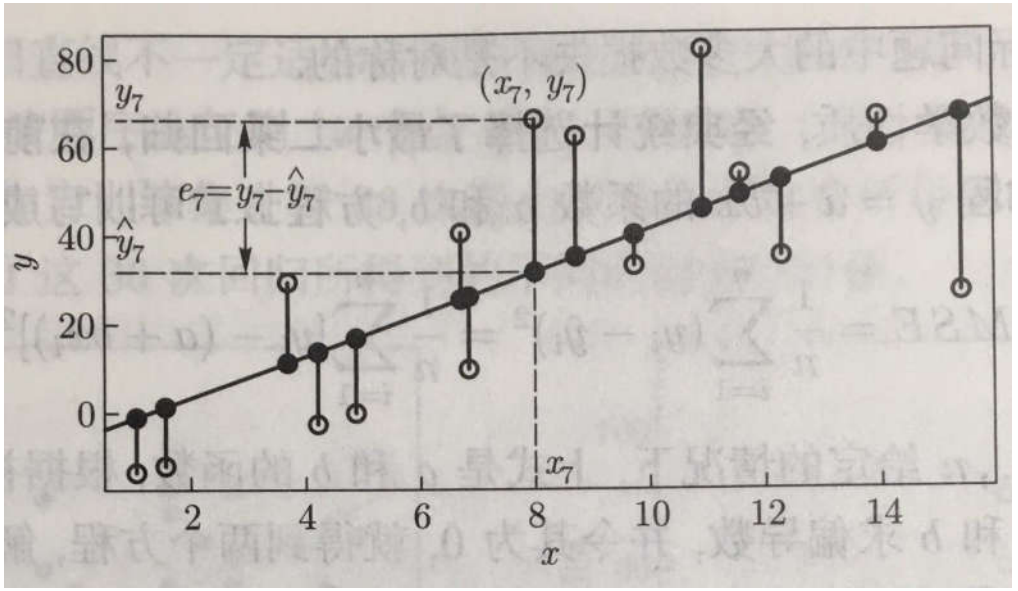
该数据是由Ezekiel（1930）提供的，并且被McNeil（1977）引用。该数据给出了20世纪20年代汽车的速度（speed，单位：每小时英里（mph）和停车的距离（dist，单位：英尺（fit））。我们的目的是建立一个用速度（speed）预测停车距离（dist）的模型。

```r
library(ggplot2)
ggplot(cars,aes(x=speed,y=dist))+
  geom_point()+
  geom_smooth(method="lm",se=FALSE)
```



■ **线性回归（linear regression）目的**

学得一个线性模型以尽可能准确地预测y实值输出标记。

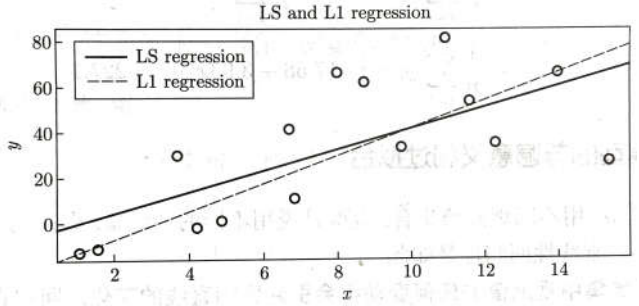


■ 到目前为止，我们的数学假定为：模型是一条直线。

■ 在无穷多条可能的直线中，如何寻找一条最合适的直线？

【1】选取残差的绝对值的平均值，即选取“平均距离”：$\frac{1}{n}\sum_{i=1}^{n}|e_i| = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$，使之最小的回归模型称为最小一乘回归或最小绝对值离差回归或L1回归。

【2】选取残差的平方的平均值，即选取“均方误差（MSE）”：$MSE = \frac{1}{n}\sum_{i=1}^{n} e^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$，使之最小的回归模型称为最小二乘回归。



经典统计选择最小二乘回归，在前计算机时代它有可计算性。

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1}{n}\sum_{i=1}^{n}[y_i - (a + bx_i]^2$$

```
> (a=lm(dist~speed,cars))#lm为线性模型（linear model）的缩写

Call:
lm(formula = dist ~ speed, data = cars)

Coefficients:
(Intercept)        speed
    -17.579        3.932

> mean((cars$dist-a$fitted.values)^2)  #MSE
[1] 227.0704
```

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1}{n}\sum_{i=1}^{n}[y_i - (a + bx_i]^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}[y_i - (-17.58 + 3.93x_i]^2 = 227.1$$

过拟合（over-fitting）

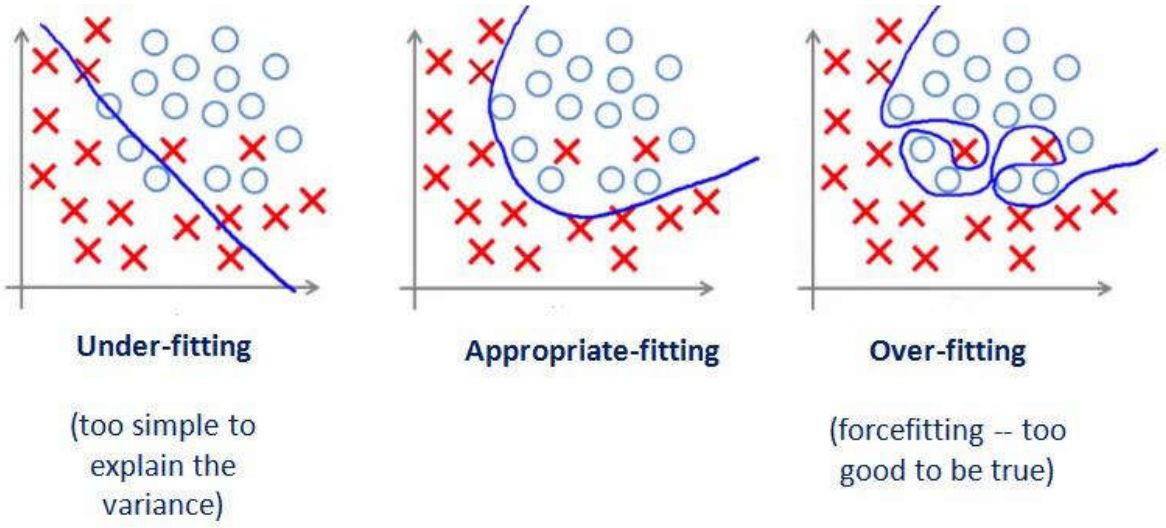
直观表现是模型在训练集上表现好，但在测试集上表现不好，泛化性能差。过拟合是在模型参数拟合过程中由于训练数据包含抽样误差，在训练时复杂的模型将抽样误差也进行了拟合导致的。

引起过拟合的可能原因有：

【1】模型本身过于复杂，以至于拟合了训练样本集中的噪声。此时需要选用更简单的模型，或者对模型进行裁剪。

【2】训练样本太少或者缺乏代表性。此时需要增加样本数，或者增加样本的多样性。

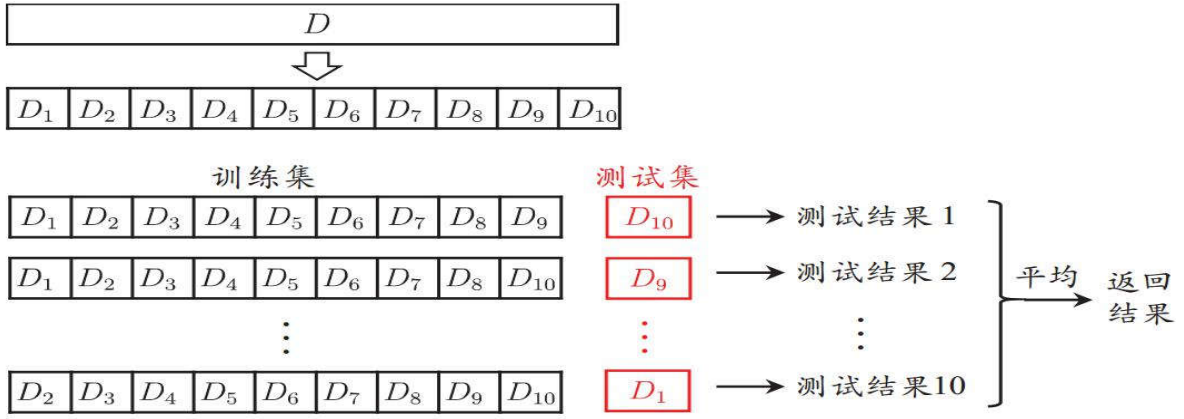
【3】训练样本噪声的干扰，导致模型拟合了这些噪声，这时需要剔除噪声数据或者改用对噪声不敏感的模型。



1.3模型评价

1.3.1交叉验证

将数据集分层采样划分为k个大小相似的互斥子集，每次用k-1个子集的并集作为训练集，余下的子集作为测试集，最终返回k个测试结果的均值，k最常用的取值是10.



10 折交叉验证示意图

1.3.2 分类问题交叉验证的预测精度标准

- 对于分类问题，交叉验证主要看模型对测试集分类的误判率：

比如测试集一共有n个观测值，有m个误判，
则误判率为：$\alpha = m/n$, 同样 $\beta = 1 - \alpha$作为准确率。

1.3.2 回归问题交叉验证的预测精度标准

对于回归问题, 最常用的是下面几种: **均方误差、均方误差平方根、标准化均方误差、平均绝对值误差、R 平方. 但要注意的是: 这些术语和数理统计中的定义完全不一样, 这里的预测值 $\hat{y}_i$ 是训练集学习到的模型拟合测试集的结果, 这里的观测值 y_i 为测试集中的观测值. 而在经典统计中, 训练集和测试集等同 (没有交叉验证).**

(1) **均方误差** (MSE, mean squared error):

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}$$

经典统计的 MSE 不是用交叉验证算出来的.

(2) **均方误差平方根** (RMSE, root mean squared error):

$$RMSE = \sqrt{MSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$$

经典统计的 RMSE 不是用交叉验证算出来的.

(3) **标准化均方误差** (NMSE, normalized mean squared error):

$$NMSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

这个统计量可能会大于 1, 这说明用模型预测的结果 ($\{\hat{y}_i\}$) 还不如不用模型, 而用因变量的样本均值 ($\bar{y}$) 作每个点的预测.

(4) **平均绝对值误差** (MAE, mean absolute error):

$$MAE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n}$$

(5) **R 平方** (RSQUARE 或 score):

$$R^2 = 1 - NMSE = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

这个统计量可能会是负值, 而经典统计中的 R^2(可决系数) 不会是负值.

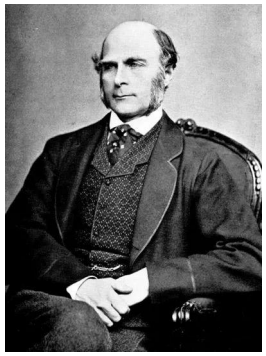
1.3.3 Z折交叉验证时提取各折下标集的R函数

```r
#分类情况
Fold=function(Z=10, w, D, seed=7777){
  n=nrow(w); d=1:n; e=levels(w[,D]);
  N=length(e)#目标变量的水平个数
  set.seed(seed)
  dd=lapply(1:N,function(i){
    d0=d[w[,D]==e[i]]; j=length(d0)
    ZT=rep(1:Z,ceiling(j/Z))[1:j]
    id=cbind(sample(ZT),d0);id})
  #上面每个dd[[i]]是随机1:Z及i类的下标集组成的矩阵
  mm=lapply(1:Z,
            function(i){u=NULL;for(j in 1:N)
            u=c(u,dd[[j]][dd[[j]][,1]==i,2]);u})
  return(mm)
}


library(tidyr)
Z=5
mm=Fold(5,singer,2)
pred=singer[,2]
for(i in 1:Z){
  pred[mm[[i]]] = rpart(voice.part~height,singer[-mm[[i]],])%>%
  predict(singer[mm[[i]],],type="class")
}
mean(pred!=singer[,2])
```

```r
#回归情况
n=nrow(cars);Z=3;set.seed(1010)
I=sample(rep(1:Z, ceiling(n/Z)))[1:n]
mm=lapply(1:Z,function(i){(1:n)[I==i]})
mm

library(tidyr)
pd=rep(9999,n)
for (i in 1:Z)
  pd[mm[[i]]]=lm(dist~speed,cars[-mm[[i]],]) %>% predict(cars[mm[[i]],])
NMSE=sum((cars$dist-pd)^2)/sum((cars$dist-mean(cars$dist))^2)
NMSE
```

二、最小二乘线性回归

2.1基本概念



回归分析之父—弗朗西斯·高尔顿

- 线性回归是最简单的回归方法。
- 线性回归中的因变量通常为连续型数据，并假定自变量和因变量之间存在某种线性关系。当模型只有一个自变量和一个因变量，称之为简单线性回归或一元线性回归；当模型有一个应变量和多个自变量时，称之为多元线性回归或多变量线性回归。
- 多元线性回归一般定义为：

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \epsilon$$

其中y是因变量X_i是自变量ϵ为误差项β_i是回归系数，当$\beta_i = 0, i > 1$时，多元线性回归退化为一元线性回归。

线性回归的基本假设：

1. 自变量和因变量之间必须存在某种线性关系
2. 不能存在任何异常点
3. 没有异方差性
4. 样本观测值相互独立
5. 误差项服从均值为0方差为常数的正态分布
6. 不存在多重共线性

参数估计

为估计回归系数,我们基于最小二乘原则最小化误差项平方和，即

$$Min\sum\epsilon^2 = Min\sum(y - \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k)^2$$

解上述等式，可得回归系：数的估计值为

$$\hat{\beta} = (X'X)^{-1}X'y$$

2.2 案例：社区犯罪

UCI Machine Learning Repository
Center for Machine Learning and Intelligent Systems

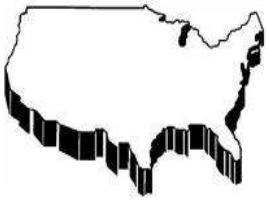
About Citation Policy Donate a Data Set Contact

Search
Repository Web

View ALL Data Sets

Communities and Crime Data Set

Download: Data Folder, Data Set Description



Abstract: Communities within the United States. The data combines socio-economic data from the 1990 US Census, law enforcement data from the 1990 US LEMAS survey, and crime data from the 1995 FBI UCR.

Data Set Characteristics:	Multivariate	**Number of Instances:**	1994	**Area:**	Social
Attribute Characteristics:	Real	**Number of Attributes:**	128	**Date Donated**	2009-07-13
Associated Tasks:	Regression	**Missing Values?**	Yes	**Number of Web Hits:**	268825

Source:

Creator: Michael Redmond (redmond '@' lasalle.edu); Computer Science; La Salle University; Philadelphia, PA, 19141, USA
-- culled from 1990 US Census, 1995 US FBI Uniform Crime Report, 1990 US Law Enforcement Management and Administrative Statistics Survey, available from ICPSR at U of Michigan.
-- Donor: Michael Redmond (redmond '@' lasalle.edu); Computer Science; La Salle University; Philadelphia, PA, 19141, USA
-- Date: July 2009

http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime

Data Set Information:

Many variables are included so that algorithms that select or learn weights for attributes could be tested. However, clearly unrelated attributes were not included; attributes were picked if there was any plausible connection to crime (N=122), plus the attribute to be predicted (Per Capita Violent Crimes). The variables included in the dataset involve the community, such as the percent of the population considered urban, and the median family income, and involving law enforcement, such as per capita number of police officers, and percent of officers assigned to drug units.

The per capita violent crimes variable was calculated using population and the sum of crime variables considered violent crimes in the United States: murder, rape, robbery, and assault. There was apparently some controversy in some states concerning the counting of rapes. These resulted in missing values for rape, which resulted in incorrect values for per capita violent crime. These cities are not included in the dataset. Many of these omitted communities were from the midwestern USA.

Data is described below based on original values. All numeric data was normalized into the decimal range 0.00-1.00 using an Unsupervised, equal-interval binning method. Attributes retain their distribution and skew (hence for example the population attribute has a mean value of 0.06 because most communities are small). E.g. An attribute described as 'mean people per household' is actually the normalized (0-1) version of that value.

The normalization preserves rough ratios of values WITHIN an attribute (e.g. double the value for double the population within the available precision - except for extreme values (all values more than 3 SD above the mean are normalized to 1.00; all values more than 3 SD below the mean are nromalized to 0.00)).

However, the normalization does not preserve relationships between values BETWEEN attributes (e.g. it would not be meaningful to compare the value for whitePerCap with the value for blackPerCap for a community)

A limitation was that the LEMAS survey was of the police departments with at least 100 officers, plus a random sample of smaller departments. For our purposes, communities not found in both census and crime datasets were omitted. Many communities are missing LEMAS data.

.arff header for Weka:

Citation Request:

Please cite the UCI Machine Learning Repository, my sources and my related paper:

U. S. Department of Commerce, Bureau of the Census, Census Of Population And Housing 1990 United States: Summary Tape File 1a & 3a (Computer Files),

U.S. Department Of Commerce, Bureau Of The Census Producer, Washington, DC and Inter-university Consortium for Political and Social Research Ann Arbor, Michigan. (1992)

U.S. Department of Justice, Bureau of Justice Statistics, Law Enforcement Management And Administrative Statistics (Computer File) U.S. Department Of Commerce, Bureau Of The Census Producer, Washington, DC and Inter-university Consortium for Political and Social Research Ann Arbor, Michigan. (1992)

U.S. Department of Justice, Federal Bureau of Investigation, Crime in the United States (Computer File) (1995)

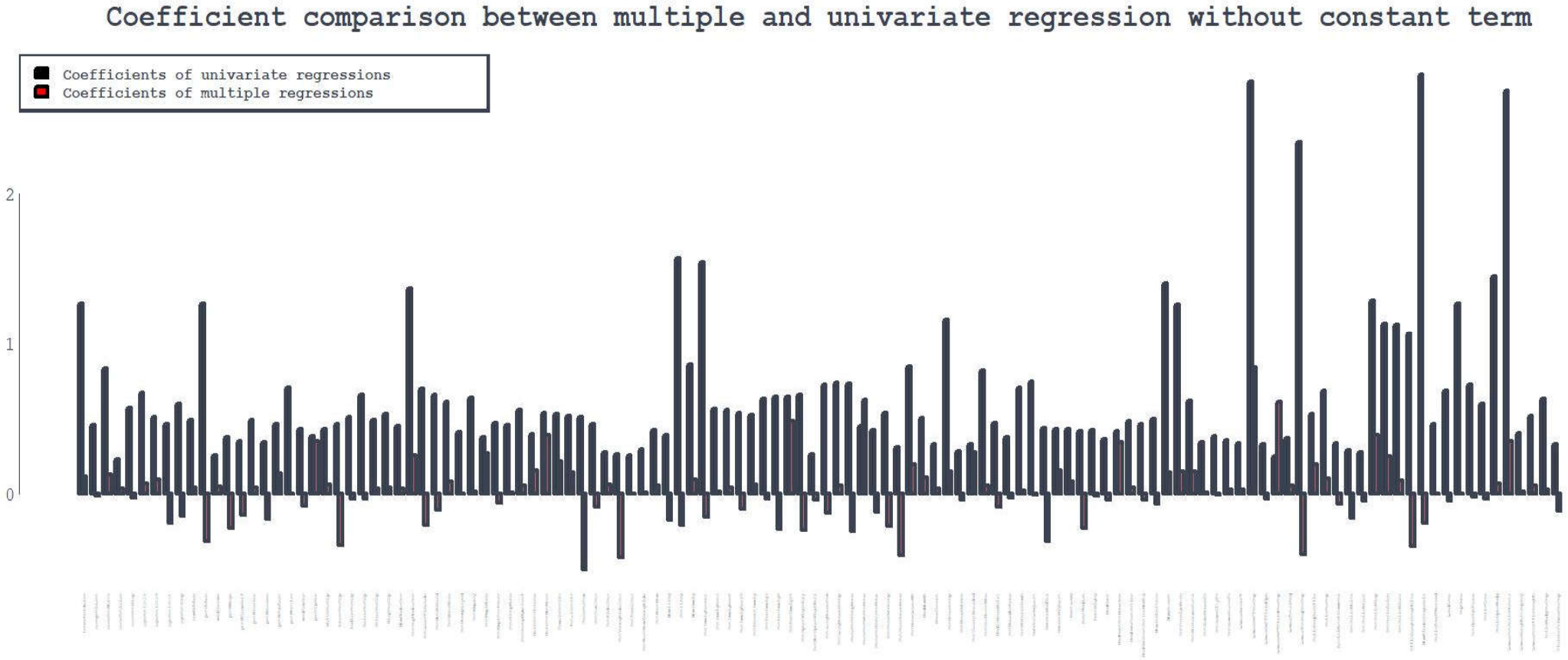
Redmond, M. A. and A. Baveja: A Data-Driven Software Tool for Enabling Cooperative Information Sharing Among Police Departments. European Journal of Operational Research 141 (2002) 660-678.

```r
#最小二乘线性回归
w=read.csv("commun123.csv")
n=nrow(w)
Pr=data.frame(rf=rep(0,n),lm=rep(0,n))
Z=10;n=nrow(w);set.seed(1010)
I=sample(rep(1:Z,ceiling(n/Z)))[1:n]
library(randomForest)
for(i in 1:Z){
  Pr$rf[I==i]=randomForest(ViolentCrimesPerPop~.,w[I!=i,])%>%predict(w[I==i,])
  Pr$lm[I==i]=lm(ViolentCrimesPerPop~., w[I!=i,],maxit=1000,size=8)%>%predict(w[I==i,])
}
RSS=sum((w[,123]-mean(w[,123]))^2)
sum((w[,123]-Pr$rf)^2)/RSS
sum((w[,123]-Pr$lm)^2)/RSS
```

```
> RSS=sum((w[,123]-mean(w[,123]))^2)
> sum((w[,123]-Pr$rf)^2)/RSS
[1] 0.3369671
> sum((w[,123]-Pr$lm)^2)/RSS
[1] 0.3505241
```

2.3 多自变量线性回归大小有意义吗？

```
w=read.csv("commun123.csv")
nm=names(w)
fo=list()
for (i in 1:122) fo[[i]]=formula(paste(nm[123],"~", nm[i], "-1"))
sbeta=vector()
for(i in 1:122) sbeta[i]=lm(fo[[i]],w)$coef
mbeta=lm(ViolentCrimesPerPop~.-1, w)$coef
b=data.frame(sbeta, mbeta)
row.names(b)=nm[-1]

barplot(t(b), beside=T, col=1:2,las=2,cex.names=.3)
title("Coefficient comparison between multiple and univariate regression without constant term")
legend("topleft", c("Coefficients of univariate regressions", "Coefficients of multiple regressions"),fill=c("black","red"))
```


Coefficient comparison between multiple and univariate regression without constant term
Coefficients of univariate regressions
Coefficients of multiple regressions
2
1
0

三、Logistic回归

3.1 基本概念

➢广义线性模型：在限定自变量用线性组合的形式，而因变量的值域并非实轴$(-\infty,+\infty)$的情况下，产生一类模型。

- 广义指因变量可以假定为任何指数族分布变量的情况。
- 连接函数指$g(.)$，将因变量和自变量的线性组合连接起来的函数，即

$$g(\mu)=\beta_0+\beta_1x_1+\beta_2x_2+\cdots+\beta_px_p$$

其中，$\mu=E(Y)$。也就是说，需要找的是因变量Y的期望μ的函数等于自变量的线性组合。

- 连接函数的逆函数$g^{-1}(.)$称为均值函数。

Bernouli分布的广义线性模型

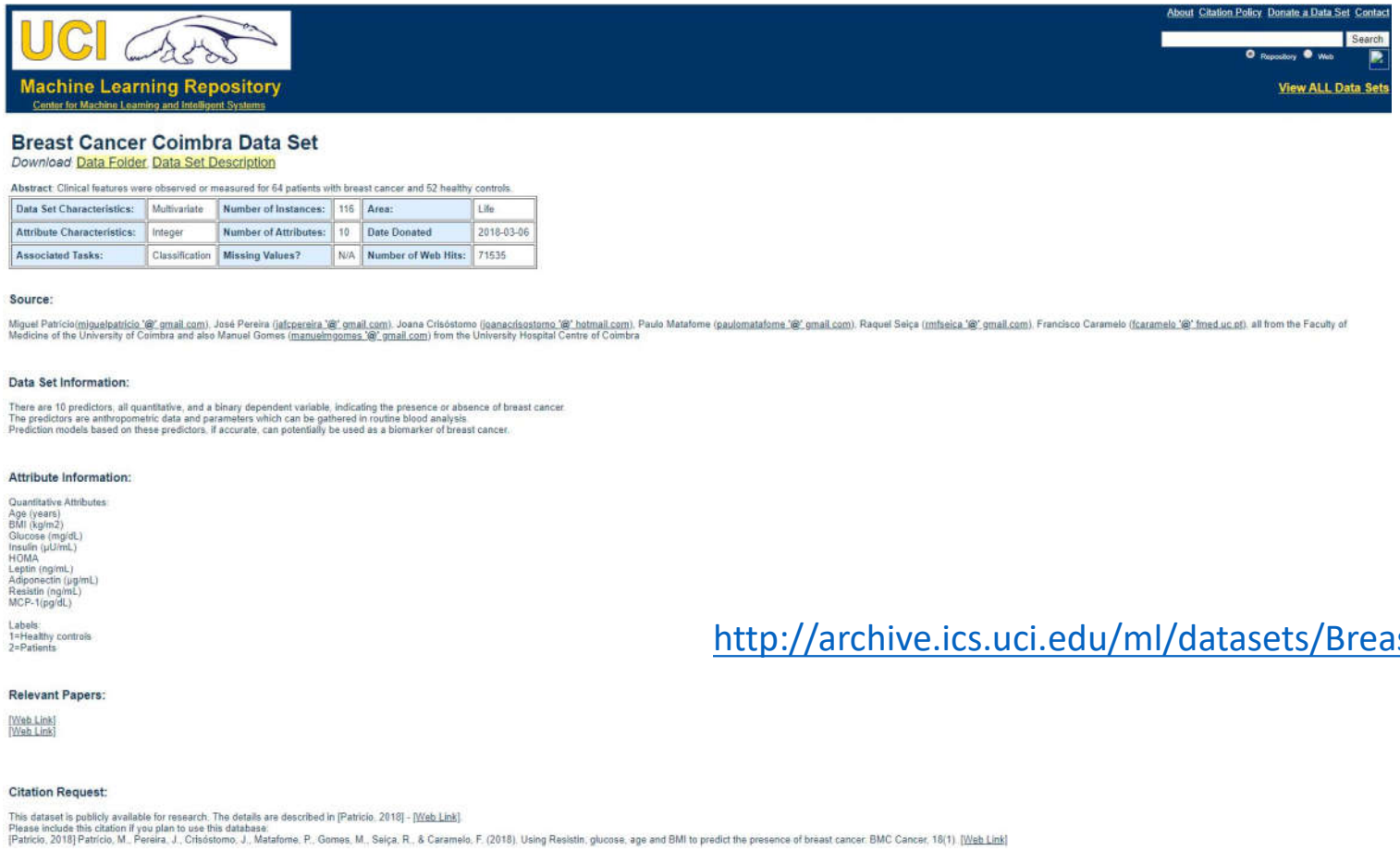
不可测的参数

- Bernouli参数p和自变量 $x_1, x_2, \ldots, x_p$ 的关系常常选用两种形式的连接函数：logit和probit:
- 广义指因变量可以假定为任何指数族分布变量的情况。

- logit连接函数:$\log\left(\frac{p}{1-p}\right) = \beta_0 + \sum_{i=1}^{p}\beta_i x_i$;逆函数： $p = \frac{\exp(\beta_0+\sum_{i=1}^{p}\beta_i x_i)}{1+\exp(\beta_0+\sum_{i=1}^{p}\beta_i x_i)}$
  - ✓ 连接函数$\log\left(\frac{p}{1-p}\right) = x$称为logit函数，而其逆函数$p = e^x/(1+e^x)$称为Logistic函数
- probit连接函数： $\Phi^{-1}(p) = \beta_0 + \sum_{i=1}^{p}\beta_i x_i$;逆函数： $p = \Phi(\beta_0 + \sum_{i=1}^{p}\beta_i x_i)$
  - ✓ probit连接函数$\Phi^{-1}(.)$是标准正态分布累积分布函数$\Phi(.)$的逆。
- 总体来看，两种连接形式的趋势都有近似“S”形曲线的样式。

3.2 Logistic回归及ROC曲线

案例



UCI Machine Learning Repository
Center for Machine Learning and Intelligent Systems

About Citation Policy Donate a Data Set Contact
Search
Repository Web
View ALL Data Sets

Breast Cancer Coimbra Data Set

Download: Data Folder, Data Set Description

Abstract: Clinical features were observed or measured for 64 patients with breast cancer and 52 healthy controls.

Data Set Characteristics:	Multivariate	**Number of Instances:**	116	**Area:**	Life
Attribute Characteristics:	Integer	**Number of Attributes:**	10	**Date Donated**	2018-03-06
Associated Tasks:	Classification	**Missing Values?**	N/A	**Number of Web Hits:**	71535

Source:

Miguel Patrício (miguelpatricio '@' gmail.com), José Pereira (jafcpereira '@' gmail.com), Joana Crisóstomo (joanacrisostomo '@' hotmail.com), Paulo Matafome (paulomatafome '@' gmail.com), Raquel Seiça (rmfseica '@' gmail.com), Francisco Caramelo (fcaramelo '@' fmed.uc.pt), all from the Faculty of Medicine of the University of Coimbra and also Manuel Gomes (manuelmgomes '@' gmail.com) from the University Hospital Centre of Coimbra

Data Set Information:

There are 10 predictors, all quantitative, and a binary dependent variable, indicating the presence or absence of breast cancer.
The predictors are anthropometric data and parameters which can be gathered in routine blood analysis.
Prediction models based on these predictors, if accurate, can potentially be used as a biomarker of breast cancer.

Attribute Information:

Quantitative Attributes:
Age (years)
BMI (kg/m2)
Glucose (mg/dL)
Insulin (µU/mL)
HOMA
Leptin (ng/mL)
Adiponectin (µg/mL)
Resistin (ng/mL)
MCP-1(pg/dL)

Labels:
1=Healthy controls
2=Patients

Relevant Papers:

[Web Link]
[Web Link]

Citation Request:

This dataset is publicly available for research. The details are described in [Patrício, 2018] - [Web Link].
Please include this citation if you plan to use this database:
[Patrício, 2018] Patrício, M., Pereira, J., Crisóstomo, J., Matafome, P., Gomes, M., Seiça, R., & Caramelo, F. (2018). Using Resistin, glucose, age and BMI to predict the presence of breast cancer. BMC Cancer, 18(1). [Web Link]

http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Coimbra

```r
w=read.csv("dataR2.csv");w[,10]=factor(w[,10])
Z=10;D=10;n=nrow(w)
mm=Fold(Z,w,D)

pred=rep(0,n)  #在数据中增加一列准备放预测的值
for (i in 1:Z) {
  pred[mm[[i]]]=glm(Classification~.,w[-mm[[i]],],family=binomial) %>%
  predict(w[mm[[i]],],type="response")
}

table(w$Classification,pred>0.5)#如果用0.5分割
```

```
> table(w$Classification,pred>0.5)

    FALSE TRUE
  1    39   13
  2    14   50
```

二分类问题的ROC曲线

统计真实标记和预测结果的组合可以得到“混淆矩阵”

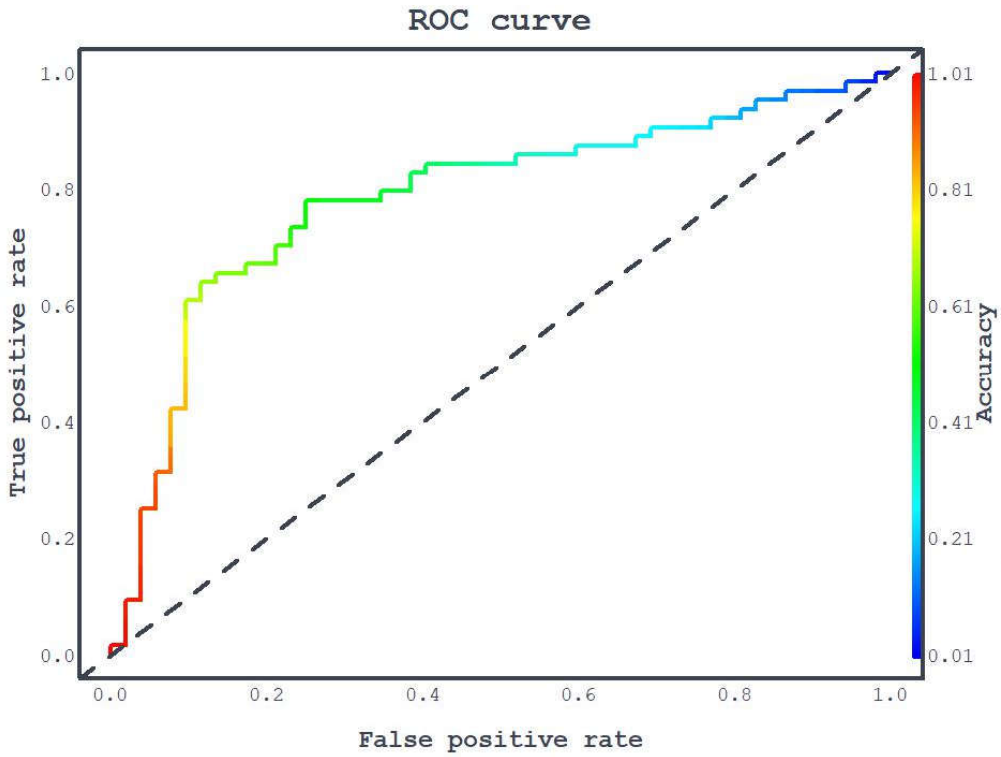
分类结果混淆矩阵

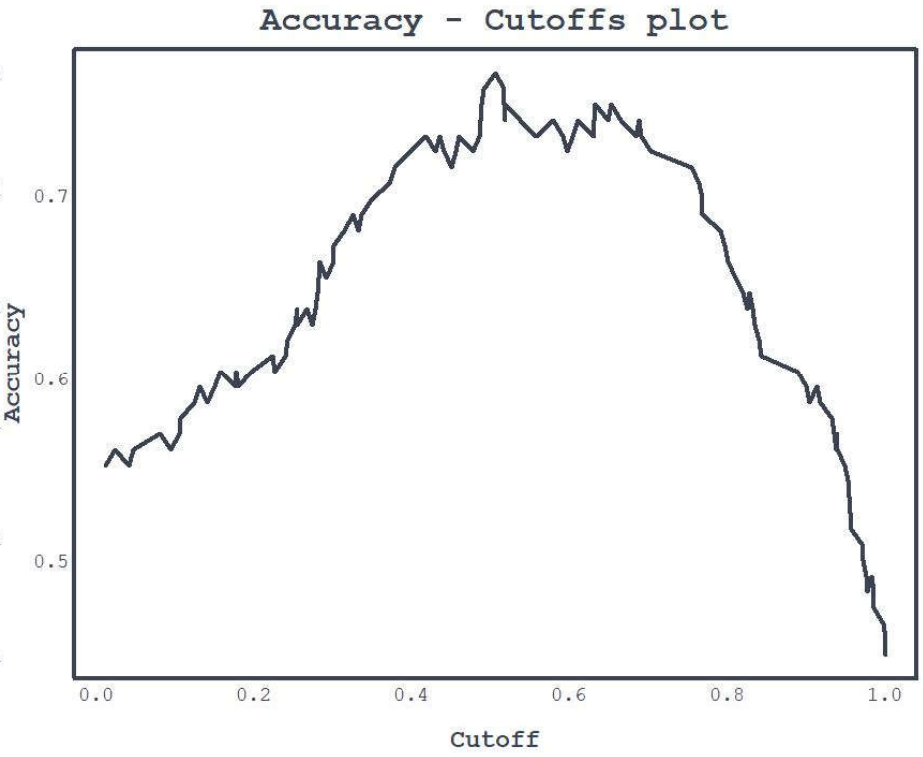
真实情况	预测结果	
	正例	反例
正例	TP (真正例)	FN (假反例)
反例	FP (假正例)	TN (真反例)

- 假阳性率反映了实际是反例被预测为正例的样本占总反例样本的比例，FPR=FP/(TN+FP)；
- 真阳性率反映了被正确预测的正例的样本占总正例样本的比例，TPR=TP/(TP+FN);
- 在理想的分类下，TPR为1，FPR为0，所有的正例和负例都被正确地判断出来。

```r
library(ROCR)
par(mfrow=c(1,2),mar=c(4,4,3,2))
ROCRpred<-prediction(pred,w$Classification)
ROCRperf<-performance(ROCRpred,'tpr','fpr')
plot(ROCRperf,colorize=TRUE, text.adj=c(-0.2,1.7))
abline(0,1,lty=2)
title("ROC curve")
plot(performance(ROCRpred,"acc"))
title("Accuracy - Cutoffs plot")

auc <-performance(ROCRpred,measure = "auc")
auc@y.values[[1]]
```

```
> table(w$Classification,pred>0.56)

    FALSE TRUE
  1    41   11
  2    19   45
```





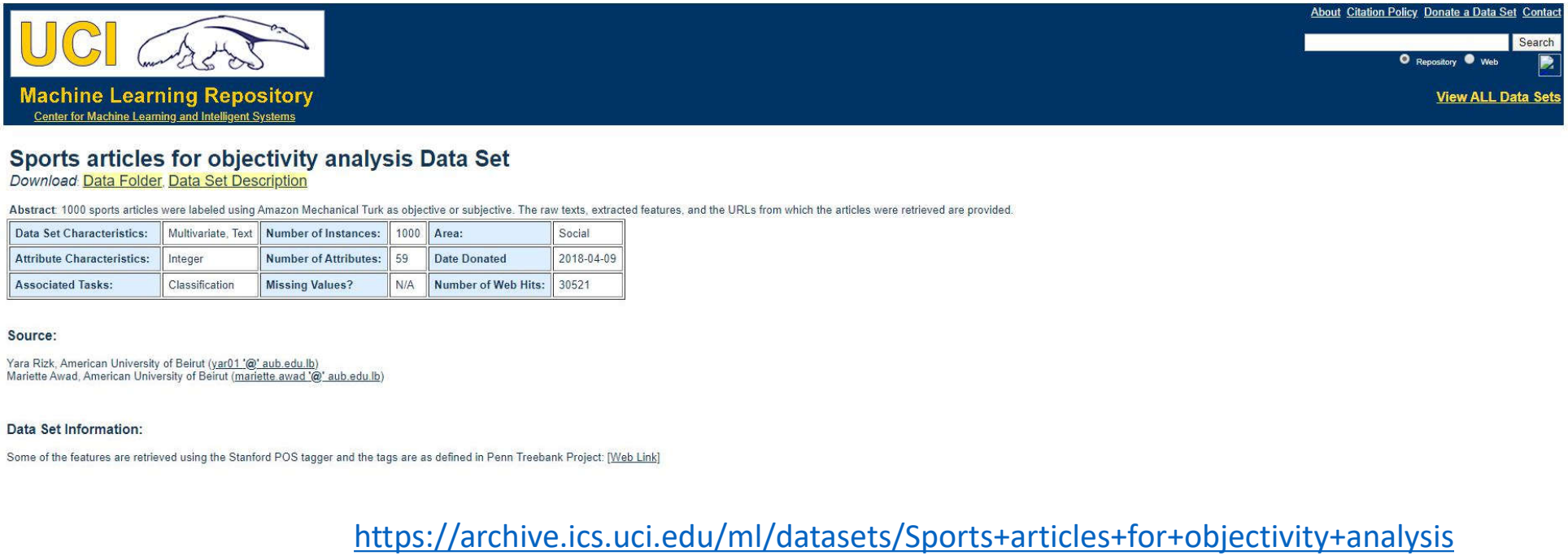
四、决策树及其组合方法

4. 决策树

- 决策树及基于决策树的组合方法形成一大类机器学习方法,它们是较基础及较典型的一类算法。区别于传统的基于假定数学模型的方法,基于决策树的方法是基于数据和算法的,它更关注如何构建从经验中学习的计算机算法。著名统计学家 Leo Breiman曾将这两类数据分析方法称为两种文化,以此来反映二者的方法有多不同(Breiman2001)。
- 决策树既可以做分类也可以做回归,分别称为分类树和回归树,以强调其目的。决策是组合方法(ensemble method)中最常用的一类基学习器.组合方法是通过多个学习器(分类或回归模型)来提高预测精度的。目前的组合方法主要包括 Bagging、 AdaBoost和随机森林。
- 为了理解这些组合方法,认知决策树的基本原理是非常重要的。

4.1.1决策树分类

案例



https://archive.ics.uci.edu/ml/datasets/Sports+articles+for+objectivity+analysis

Attribute Information:

TextID text file name
URL link to article
Label objective vs. subjective
totalWordsCount total number of words in the article
semanticobjscore Frequency of words with an objective SENTIWORDNET score
semanticsubjscore Frequency of words with a subjective SENTIWORDNET score
CC Frequency of coordinating conjunctions
CD Frequency of numerals and cardinals
DT Frequency of determiners
EX Frequency of existential there
FW Frequency of foreign words
INs Frequency of subordinating preposition or conjunction
JJ Frequency of ordinal adjectives or numerals
JJR Frequency of comparative adjectives
JJS Frequency of superlative adjectives
LS Frequency of list item markers
MD Frequency of modal auxiliaries
NN Frequency of singular common nouns
NNP Frequency of singular proper nouns
NNPS Frequency of plural proper nouns
NNS Frequency of plural common nouns
PDT Frequency of pre-determiners
POS Frequency of genitive markers
PRP Frequency of personal pronouns
PRP$ Frequency of possessive pronouns
RB Frequency of adverbs
RBR Frequency of comparative adverbs
RBS Frequency of superlative adverbs
RP Frequency of particles
SYM Frequency of symbols
TOs Frequency of 'to' as preposition or infinitive marker
UH Frequency of interjections
VB Frequency of base form verbs
VBD Frequency of past tense verbs
VBG Frequency of present participle or gerund verbs
VBN Frequency of past participle verbs
VBP Frequency of present tense verbs with plural 3rd person subjects
VBZ Frequency of present tense verbs with singular 3rd person subjects
WDT Frequency of WH-determiners
WP Frequency of WH-pronouns
WP$ Frequency of possessive WH-pronouns
WRB Frequency of WH-adverbs
baseform Frequency of infinitive verbs (base form verbs preceded by â€œtoâ€□)
Quotes Frequency of quotation pairs in the entire article
questionmarks Frequency of questions marks in the entire article
exclamationmarks Frequency of exclamation marks in the entire article
fullstops Frequency of full stops
commas Frequency of commas
semicolon Frequency of semicolons
colon Frequency of colons
ellipsis Frequency of ellipsis
pronouns1st Frequency of first person pronouns (personal and possessive)
pronouns2nd Frequency of second person pronouns (personal and possessive)
pronouns3rd Frequency of third person pronouns (personal and possessive)
compsupadjadv Frequency of comparative and superlative adjectives and adverbs
past Frequency of past tense verbs with 1st and 2nd person pronouns
imperative Frequency of imperative verbs
present3rd Frequency of present tense verbs with 3rd person pronouns
present1st2nd Frequency of present tense verbs with 1st and 2nd person pronouns
sentence1st First sentence class
sentencelast Last sentence class
txtcomplexity Text complexity score

Relevant Papers:

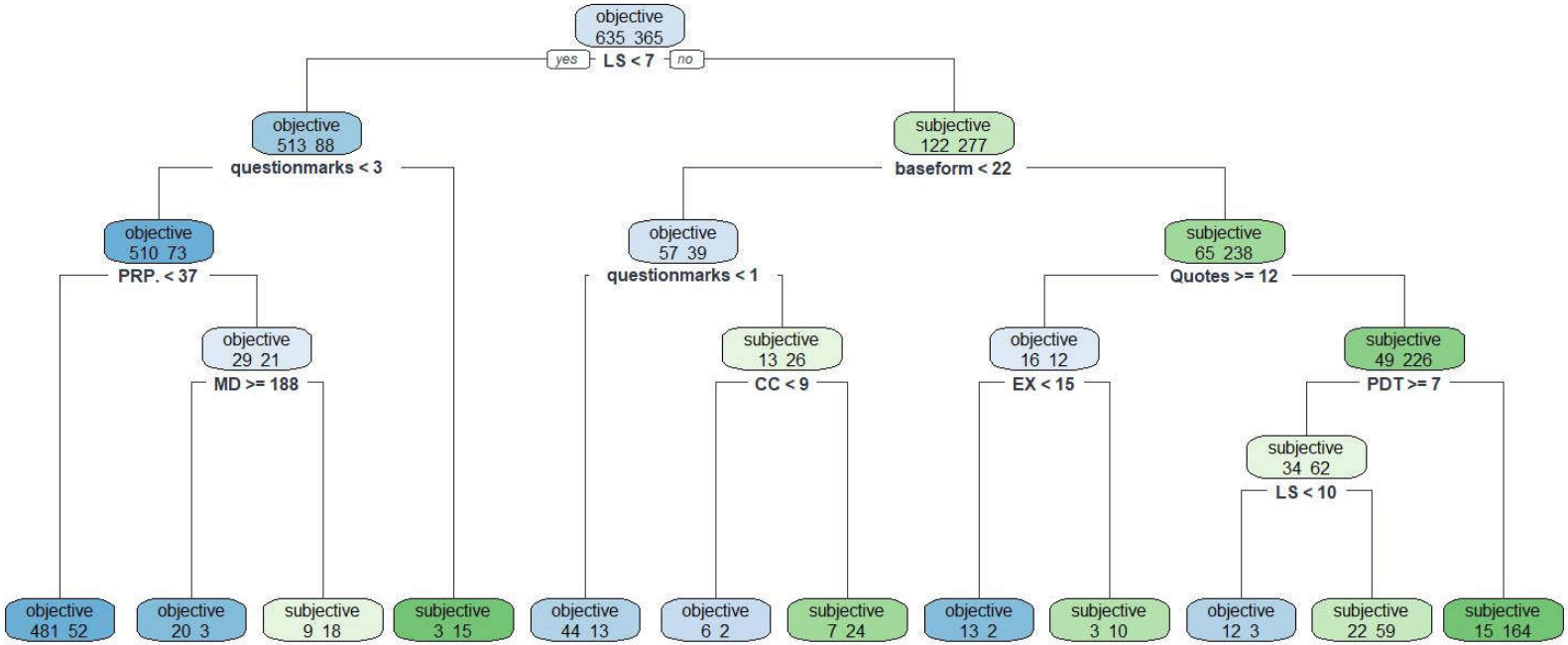
Nadine Hajj, Yara Rizk, and Mariette Awad, 'A Subjectivity Classification Framework for Sports Articles using Cortical Algorithms for Feature Selection,' Springer Neural Computing and Applications, 2018.
Yara Rizk, and Mariette Awad, 'Syntactic Genetic Algorithm for a Subjectivity Analysis of Sports Articles,' International Conference on Cybernetic Intelligent Systems, Limerick, Ireland, 2012.

Citation Request:

Nadine Hajj, Yara Rizk, and Mariette Awad, 'A Subjectivity Classification Framework for Sports Articles using Cortical Algorithms for Feature Selection,' Springer Neural Computing and Applications, 2018.

决策树分类结果

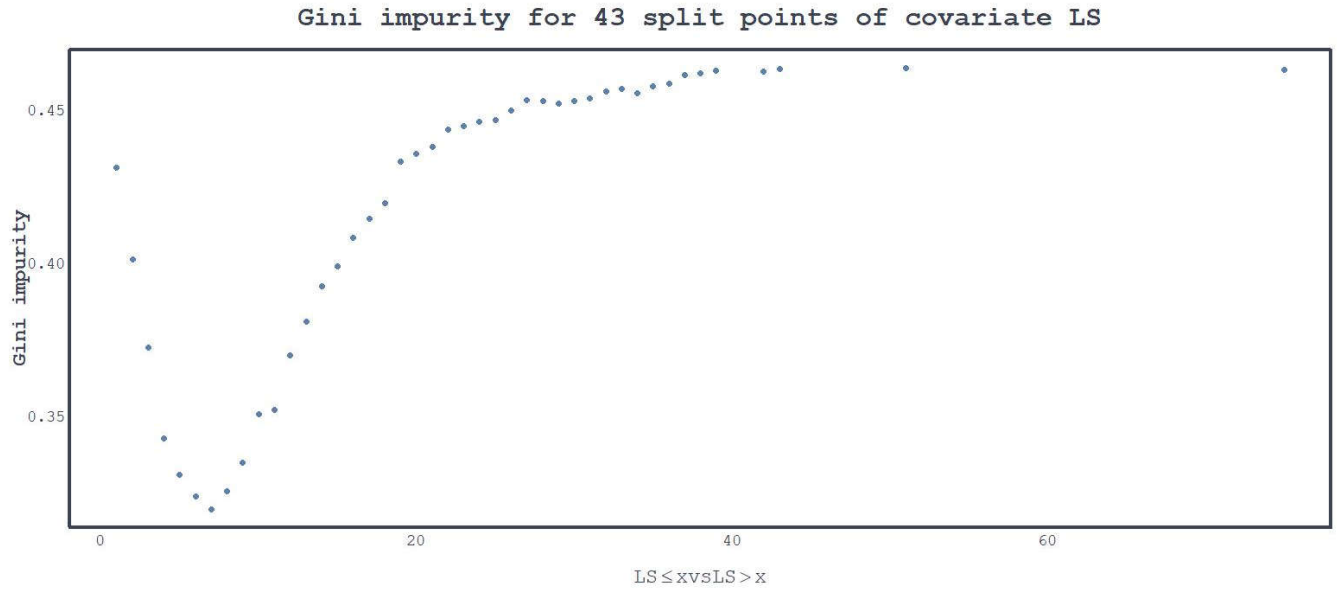
```
library(rpart.plot)
w=read.csv("Sports.csv")
b=rpart(Label~.,w)
rpart.plot(b,extra=1)
```



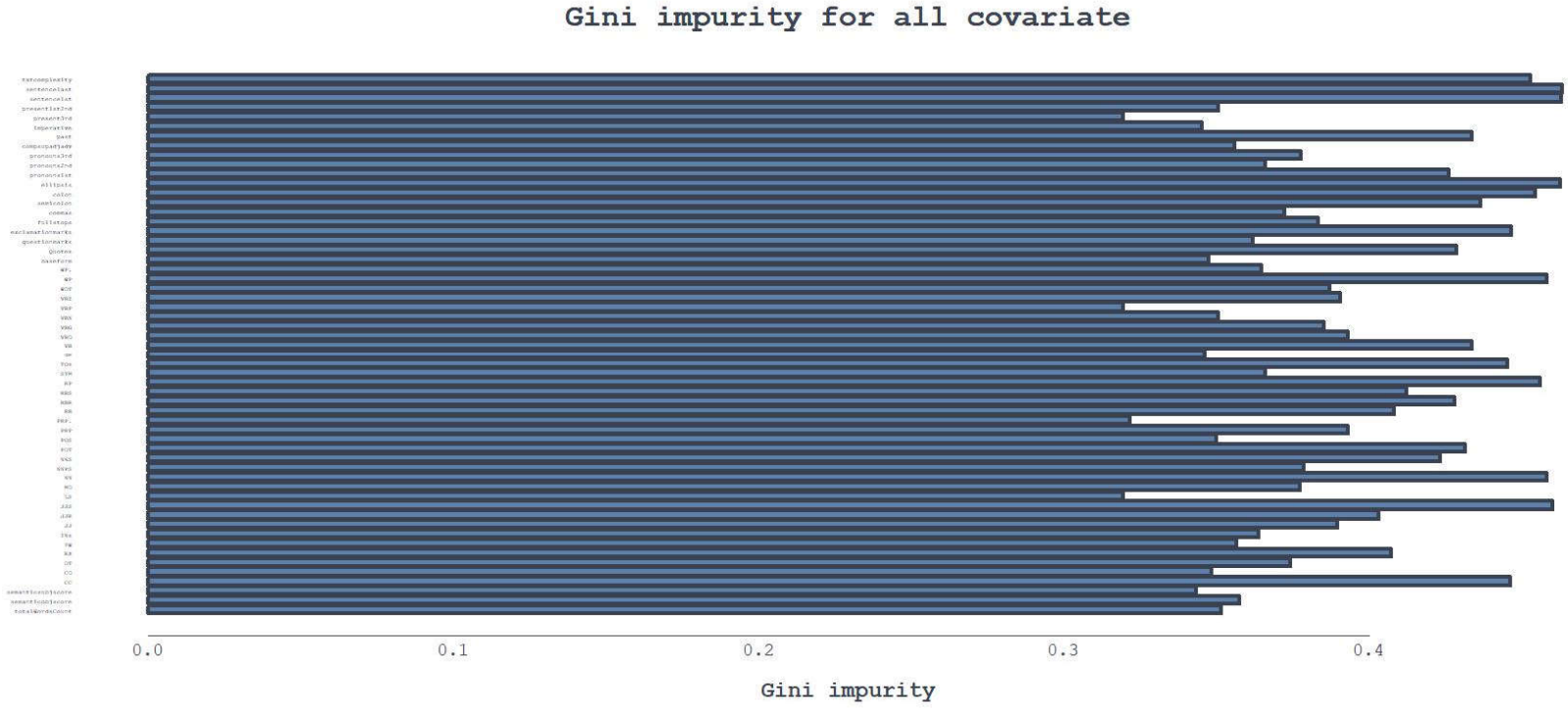
```
> b
n= 1000

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

 1) root 1000 365 objective (0.63500000 0.36500000)
   2) LS< 6.5 601  88 objective (0.85357737 0.14642263)
     4) questionmarks< 2.5 583  73 objective (0.87478559 0.12521441)
       8) PRP.< 36.5 533  52 objective (0.90243902 0.09756098) *
       9) PRP.>=36.5 50  21 objective (0.58000000 0.42000000)
        18) MD>=187.5 23   3 objective (0.86956522 0.13043478) *
        19) MD< 187.5 27   9 subjective (0.33333333 0.66666667) *
     5) questionmarks>=2.5 18   3 subjective (0.16666667 0.83333333) *
   3) LS>=6.5 399 122 subjective (0.30576441 0.69423559)
     6) baseform< 21.5 96  39 objective (0.59375000 0.40625000)
      12) questionmarks< 0.5 57  13 objective (0.77192982 0.22807018) *
      13) questionmarks>=0.5 39  13 subjective (0.33333333 0.66666667)
        26) CC< 8.5 8   2 objective (0.75000000 0.25000000) *
        27) CC>=8.5 31   7 subjective (0.22580645 0.77419355) *
     7) baseform>=21.5 303  65 subjective (0.21452145 0.78547855)
      14) Quotes>=11.5 28  12 objective (0.57142857 0.42857143)
        28) EX< 14.5 15   2 objective (0.86666667 0.13333333) *
        29) EX>=14.5 13   3 subjective (0.23076923 0.76923077) *
      15) Quotes< 11.5 275  49 subjective (0.17818182 0.82181818)
        30) PDT>=6.5 96  34 subjective (0.35416667 0.64583333)
          60) LS< 9.5 15   3 objective (0.80000000 0.20000000) *
          61) LS>=9.5 81  22 subjective (0.27160494 0.72839506) *
        31) PDT< 6.5 179  15 subjective (0.08379888 0.91620112) *
```

数量自变量竞争拆分变量

```
gip=function(y,x,p){#计算自变量观测值在分割点p的Gini不纯度
  pr=prop.table(table(x>=p))
  p1=prop.table(table(y[x<p]))
  p2=prop.table(table(y[x>=p]))
  return(c(1-sum(p1^2),1-sum(p2^2))%*%pr)
}

d=sort(unique(w$LS))#变量w$LS的不重复观测值

g=NULL
for (i in d[-1]){  #计算43个分割对应的Gini不纯度
  g=c(g,gip(w$Label,w$LS,i))
}

plot(d[-1],g, col=4,pch=16,
xlab=expression(paste(LS<=x," vs ",LS>x)),
ylab="Gini impurity",
main="Gini impurity for 43 split points of covariate LS")
```



```r
v=w[,-c(17,40)]#去掉全为0的两个变量的数据
G=NULL
for(i in 2:ncol(v)){
  d=sort(unique(v[,i]))
  g=NULL
  for(j in d[-1]){
    g=c(g,gip(v$Label,v[,i],j))
  }
    G=c(G,min(g))
}
#绘图:
barplot(G,names.arg=names(v)[-1],horiz=TRUE,las=1,cex.names=.4,
        xlab="Gini impurity", col=4)
title("Gini impurity for all covariate")
```

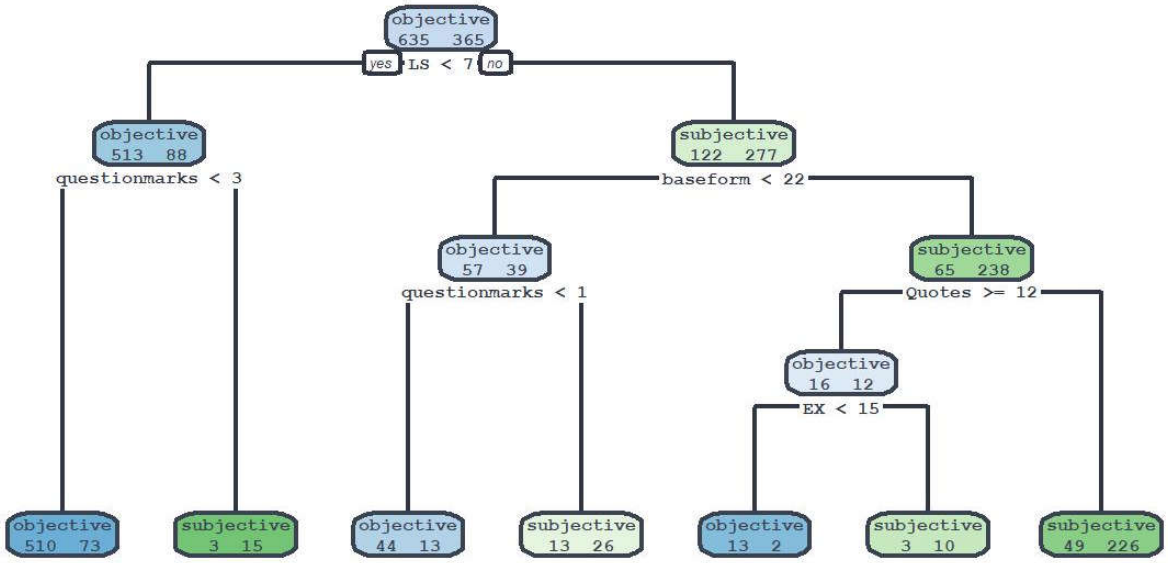


决策树的剪枝

复杂度参数（complexity parameter,cp）,用于控制决策树的大小并选择最佳树大小。如果从当前节点向决策树添加另一个变量来拆分的成本高于cp值，则树的构建不会继续。

```
> b=rpart(Label~.,w)
> printcp(b)

Classification tree:
rpart(formula = Label ~ ., data = w)

Variables actually used in tree construction:
[1] baseform      CC            EX            LS            MD            PDT           PRP.          questionmarks Quotes

Root node error: 365/1000 = 0.365

n= 1000

        CP nsplit rel error  xerror     xstd
1 0.424658      0   1.00000 1.00000 0.041710
2 0.049315      1   0.57534 0.63562 0.036571
3 0.035616      2   0.52603 0.64658 0.036788
4 0.032877      3   0.49041 0.59452 0.035712
5 0.015068      4   0.45753 0.55068 0.034720
6 0.012329      6   0.42740 0.52877 0.034192
7 0.010959     10   0.37808 0.54521 0.034590
8 0.010000     11   0.36712 0.55616 0.034849
> b$cptable[which.min(b$cptable[,"xerror"]),"CP"]
[1] 0.01232877
```

```
ptree<-prune(b,cp=b$cptable[which.min(b$cptable[,"xerror"]),"CP"])
rpart.plot(ptree,extra = 1)
```



决策树分类的混淆矩阵

```
> table(w$Label,predict(ptree,w,type = "class"))

             objective subjective
  objective        567         68
  subjective        88        277
> mean(w$Label!=predict(ptree,w,type = "class"))
[1] 0.156
```

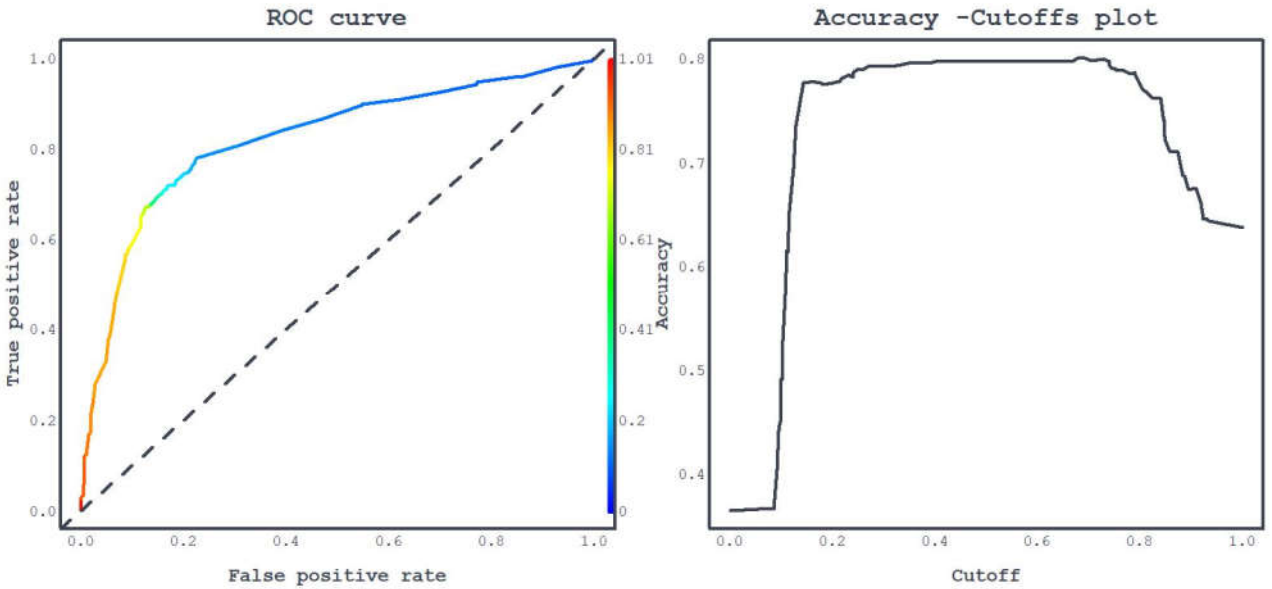
```
#决策树分类的交叉验证
library(magrittr)
Z=10; D=1;n=nrow(w)
mm=Fold(Z,w,D)
pr=data.frame(cp10=w$Label,cp109=w$Label)#准备放预测的值
for(i in 1:Z){
  pr$cp10[mm[[i]]]=rpart(Label~.,w[-mm[[i]],],cp=0.01)%>%
    predict(w[mm[[i]],],type="class")
  pr$cp109[mm[[i]]]=rpart(Label~.,w[-mm[[i]],],cp=0.0109589)%>%
    predict(w[mm[[i]],],type="class")
}
mean(pr$cp10!=w$Label)#0.192
mean(pr$cp109!=w$Label)#.189
```

分类的ROC曲线

```
#分类的ROC曲线
w=read.csv("Sports.csv")
library(rpart)
library(magrittr)
Z=10; D=1;n=nrow(w)
mm=Fold(Z,w,D)
p=rep(0,n)

for(i in 1:Z){
  p[mm[[i]]] =rpart(Label~.,w[-mm[[i]],])%>%
    predict(w[mm[[i]],],type="prob")%>%
    .[,2]
}

library(ROCR)
par(mfrow=c(1,2), mar=c(4,4,3,2))
ROCRpred <-prediction(p, w$Label)
ROCRperf<-performance(ROCRpred,'tpr','fpr')
plot(ROCRperf, colorize=TRUE,text.adj=c(-0.2,1.7))
abline(0,1,lty =2)
title("ROC curve")
plot(performance(ROCRpred, "acc"))
title("Accuracy -Cutoffs plot")
```

- ROC曲线上一个好的分类器，应该紧贴左上角，曲线上离左上角最近的点对应的临界点即为使得该分类器最好的临界点。
- 其中，约登指数来描述ROC曲线，定义约登指数=TPR-FPR，约登指数越高的点越接近左上角，分类器越好。

```
> auc <-performance(ROCRpred,measure = "auc")
> auc@y.values[[1]]
[1] 0.8213677
```



4.1.2决策树回归

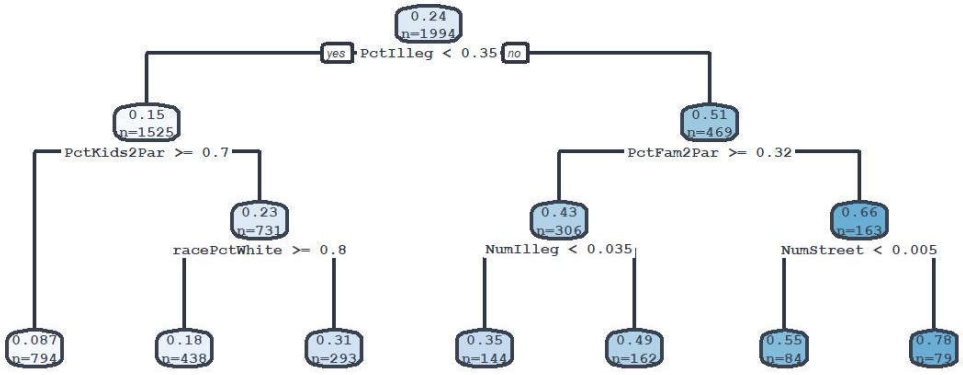
```
#决策树回归
library(rpart.plot)
w=read.csv("commun123.csv")
a=rpart(ViolentCrimesPerPop~.,w)
rpart.plot(a,extra=1)
> a
n= 1994

node), split, n, deviance, yval
      * denotes terminal node

 1) root 1994 108.184000 0.23797890
   2) PctIlleg< 0.345 1525  31.867720 0.15493770
     4) PctKids2Par>=0.695 794   4.910266 0.08700252 *
     5) PctKids2Par< 0.695 731  19.312720 0.22872780
      10) racePctWhite>=0.795 438   8.287112 0.17726030 *
      11) racePctWhite< 0.795 293   8.130995 0.30566550 *
   3) PctIlleg>=0.345 469  31.605720 0.50799570
     6) PctFam2Par>=0.315 306  14.914930 0.42630720
      12) NumIlleg< 0.035 144   5.636156 0.35055560 *
      13) NumIlleg>=0.035 162   7.717951 0.49364200 *
     7) PctFam2Par< 0.315 163  10.815500 0.66134970
      14) NumStreet< 0.005 84   4.686281 0.54880950 *
      15) NumStreet>=0.005 79   3.934119 0.78101270 *
```

在回归中每个节点选择拆分变量的步骤是:

(1) 对于某个节点, 在每个自变量中选择一个分割点, 使得和别的分割点相比, 其子节点的残差平方和 (residual sum of squares, RSS) 最小[①]. 那么, RSS 如何计算? 以根节点的拆分为例, 考虑用变量 PctIlleg 来拆分, 并且取分割点 0.35. 于是, 根节点数据根据是否满足条件 `PctIlleg<0.35` 拆分出左右两个子节点所代表的 2 个数据子集, 记左边子节点的因变量为 $y_i^{(l)}, i = 1, \cdots, 1525$, 而记右边子节点的因变量为 $y_i^{(r)}, i = 1, \cdots, 469$, 左右两个子节点的预测值分别是这两个节点因变量观测值的均值, 这两个均值分别是 $\bar{y}^{(l)} = 0.15$ 和 $\bar{y}^{(r)} = 0.51$, 那么该分割所得到的 RSS 为:

$$RSS = \sum_{i=1}^{1525}(y_i^{(l)} - 0.15)^2 + \sum_{i=1}^{469}(y_i^{(r)} - 0.51)^2.$$

(2) 在每个自变量都选出该节点 (使 RSS 最小) 的最优分割后, 比较所有自变量按照各自最优分割所计算的 RSS, 选择使得 RSS 最小的自变量为首选拆分变量. 注意: 均方误差 MSE=RSS/n.

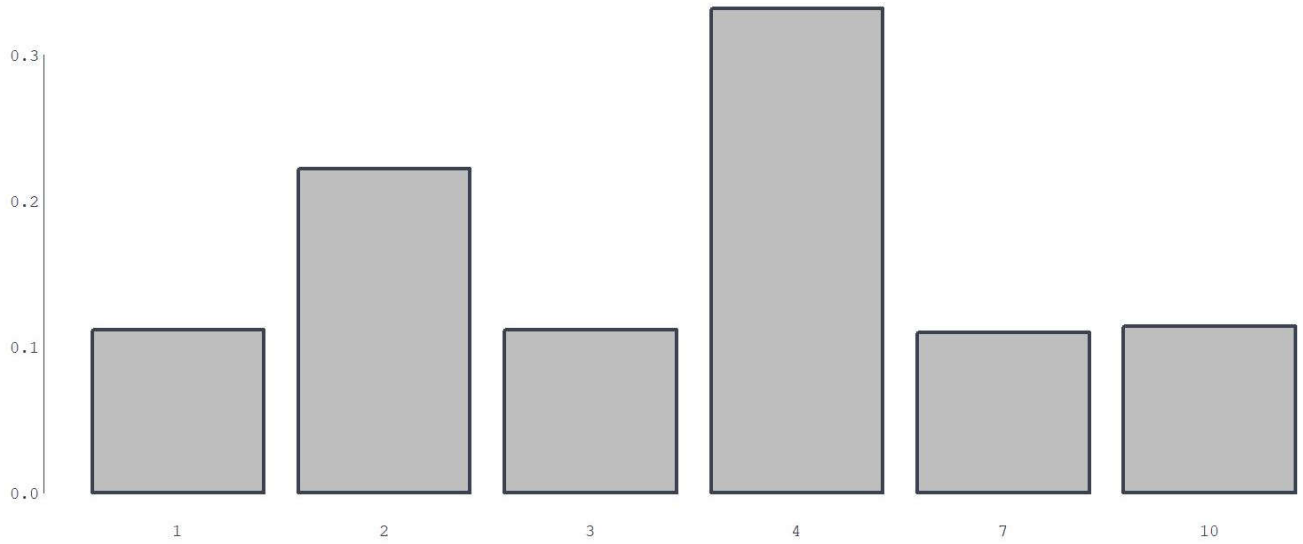


4.2有放回再抽样简介

在抽样过程中，每次被抽中的观测值都会被放回数据集中，而下一次抽样还还有可能被抽到。

4.2.1 经验分布

```
library(tidyr)
x=c(1,2,2,3,7,10,4,4,4)
sample(x,100000,rep=TRUE)%>%table()%>%prop.table()%>%barplot()
```



4.2.2 OOB（out of bag，也称口袋外）数据

由于是有放回抽样，每次抽样时，即使是等可能抽样，都会存在有些观测值会重复被抽到，而这些观测值不会被抽到。

```
> x=0:9;set.seed(100)
> for (i in 1:10) {
+   s=sample(x,10,replace = TRUE);oob=setdiff(x,s)
+   cat(paste0("sample ", i,":"), s,"oob:",oob,"\n") #cat是一种输出格式
+ }
sample 1: 9 6 5 2 8 9 6 5 5 3 oob: 0 1 4 7
sample 2: 6 5 1 6 6 6 7 1 2 2 oob: 0 3 4 8 9
sample 3: 7 1 8 1 2 3 3 3 4 6 oob: 0 5 9
sample 4: 8 3 1 5 6 0 5 8 8 8 oob: 2 4 7 9
sample 5: 5 7 6 0 8 5 3 7 2 3 oob: 1 4 9
sample 6: 2 2 3 4 6 3 2 8 7 5 oob: 0 1 9
sample 7: 1 4 2 4 4 3 4 8 9 8 oob: 0 5 6 7
sample 8: 9 4 9 9 6 7 8 9 6 2 oob: 0 1 3 5
sample 9: 9 9 8 5 4 1 7 7 1 2 oob: 0 3 6
sample 10: 6 9 4 6 7 0 7 5 9 2 oob: 1 3 8
```

4.2.3 非等权放回再抽样

为了“照顾”某些观测值（所代表的信息），在再抽样中，增加这些观测值被抽中的概率。

```
> x=c(2,2,2,9,9,7,7,7,6,6,6,7,9,9,3)
> set.seed(1111)
> sample(x,15,rep=T) #等概率抽样
 [1] 7 7 2 6 9 2 7 7 6 3 3 7 3 7 2
> pr=c(rep(1,14),10) #设定x中3被抽中的概率为前14个数目每个被抽中概率的10倍
> sample(x,rep=T,prob = pr) #增加3被抽中的概率
 [1] 3 2 7 9 6 3 2 9 2 3 3 3 3 9 3
```

4.3Bagging（bootstrap aggreating）(Breiman,1996)

对于一个样本量为n的原始数据,有放回等可能地重复抽取(比如是m个)和原始数据同样大(样本量n)的样本,然后对每个样本建立一个决策树,如此建立m个决策树。得到这些决策树之后,具体的预测过程如下:

- 决策树分类:对于一个新的观测值,每个决策树会给出该观测值属于哪一类的预测(一共有m个结果),那么 Bagging的最终预测值就是这m个决策树“等权投票”的结果,即m个结果中占多数的结果.以判断性别为例,100棵树会给出100个结果,有的是男性,有的是女性,如果预测男性的树的棵数过半,则 Bagging最后预测为男性,否则为女性。

- 决策树回归:对于一个新的观测值,每个决策树会给出该观测值因变量一个预测值(一共有m个结果),那么 Bagging的最终预测值就是这m个预测值的均值。

4.3.1 Bagging分类

```
> library(ipred)
> w=read.csv("derm.csv")
> for (i in (1:ncol(w))[-34]) w[,i]=factor(w[,i])#把除年龄之外的哑元变量因子化
> a=bagging(V35~.,w)#主程序
> table(w$V35,predict(a,w))#混淆矩阵

    1   2   3   4   5   6
1 112   0   0   0   0   0
2   0  61   0   0   0   0
3   0   0  72   0   0   0
4   0   0   0  49   0   0
5   0   0   0   0  52   0
6   0   0   0   0   0  20
```

```
> library(rpart)
> library(tidyr)
> Z=10;D=35;n=nrow(w)
> mm=Fold(Z,w,D)
> Pr=data.frame(bag=w$V35,tree=w$V35)
> for(i in 1:Z) {
+   Pr$bag[mm[[i]]]=bagging(V35~.,w[-mm[[i]],])%>%
+     predict(w[mm[[i]],])
+   Pr$tree[mm[[i]]]=rpart(V35~.,w[-mm[[i]],])%>%
+     predict(w[mm[[i]],],type="class")
+ }
```

```
> table(w$V35,Pr$bag)

    1   2   3   4   5   6
1 110   1   0   1   0   0
2   1  56   1   2   0   1
3   0   0  71   0   1   0
4   0   4   0  45   0   0
5   2   0   0   0  50   0
6   1   0   0   0   0  19
> mean(Pr$bag!=w$V35)
[1] 0.04098361
> table(w$V35,Pr$tree)

    1   2   3   4   5   6
1 106   3   0   3   0   0
2   1  56   1   1   0   2
3   0   0  69   2   1   0
4   0   5   0  44   0   0
5   2   0   0   0  50   0
6   1   2   0   0   0  17
> mean(Pr$tree!=w$V35)
[1] 0.06557377
```

4.3.2 Bagging回归

关于交叉验证的NMSE的计算公式为：$NMSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}$

```
w=read.csv("commun123.csv");n=nrow(w)
Pr=data.frame(bag=rep(0,n),tree=rep(0,n))
Z=10;set.seed(1010)
I=sample(rep(1:Z,ceiling(n/Z)))[1:n]
for (i in 1:Z) {
  Pr$bag[I==i]=
    bagging(ViolentCrimesPerPop~.,w[I!=i,])%>%
    predict(w[I==i,])
  Pr$tree[I==i]=
    rpart(ViolentCrimesPerPop~.,w[I!=i,])%>%
    predict(w[I==i,])
}
> sum((Pr$bag-w[,123])^2)/sum((w[,123]-mean(w[,123]))^2)
[1] 0.3784855
> sum((Pr$tree-w[,123])^2)/sum((w[,123]-mean(w[,123]))^2)
[1] 0.4427989
```

4.4随机森林

和Bagging的不同点在于：

（1）它的每棵树的每个节点的拆分变量都不是由全部自变量竞争，而是由随机选取的少数变量竞争。

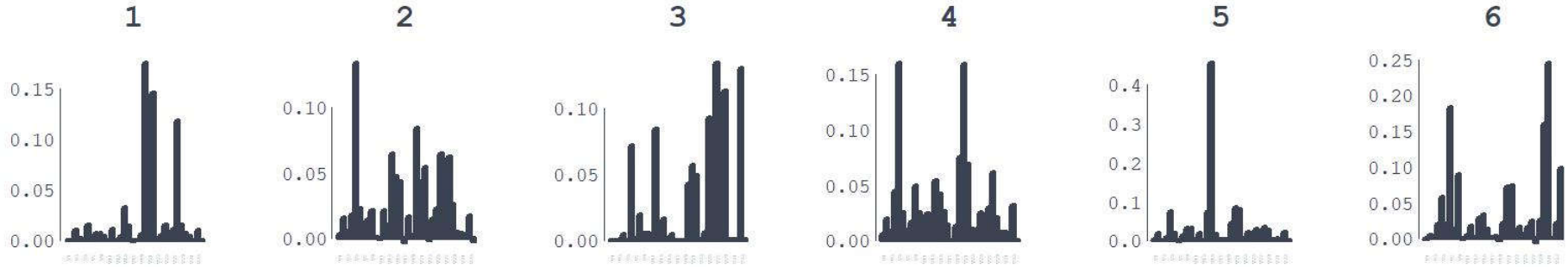
（2）每棵决策树都长不到再长为止。

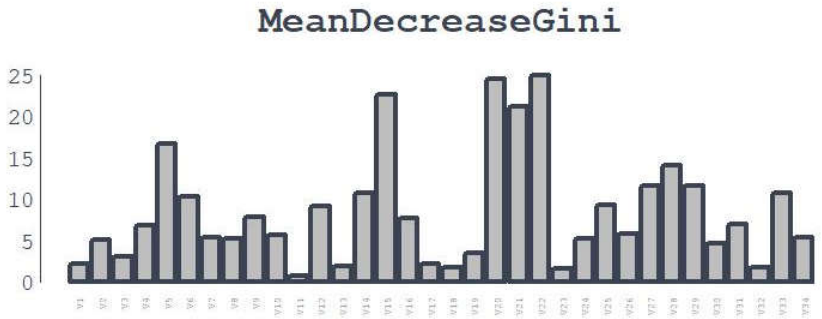
4.4.1 随机森林分类

```
library(randomForest)
w=read.csv("derm.csv")
for (i in (1:ncol(w))[-34])w[,i]=factor(w[,i])
a=randomForest(V35~.,w,localImp=TRUE,proximity=TRUE)
> a

Call:
 randomForest(formula = V35 ~ ., data = w, localImp = TRUE, proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 5

        OOB estimate of  error rate: 2.73%
Confusion matrix:
    1  2  3  4  5  6 class.error
1 112  0  0  0  0  0  0.00000000
2   1 55  0  5  0  0  0.09836066
3   0  0 72  0  0  0  0.00000000
4   0  4  0 45  0  0  0.08163265
5   0  0  0  0 52  0  0.00000000
6   0  0  0  0  0 20  0.00000000
```

随机森林分类的变量重要性

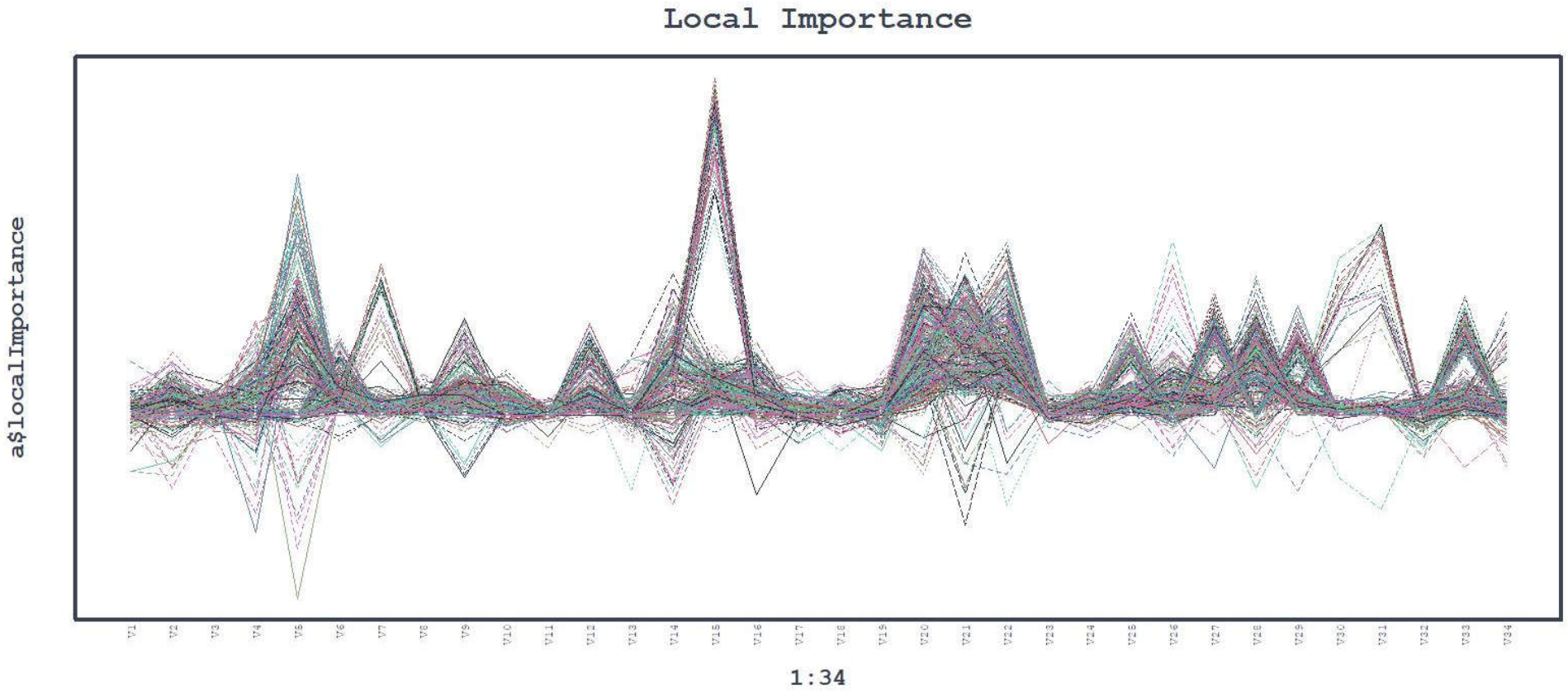
```
layout(matrix(c(1:6,rep(7,3),rep(8,3)),2,6,by=T))
for(i in 1:8) {
  if(i>6) ca=.5 else ca=.3
  barplot(a$importance[,i],
          main = colnames(a$importance)[i],
          las=2,cex.names = ca)
}
```





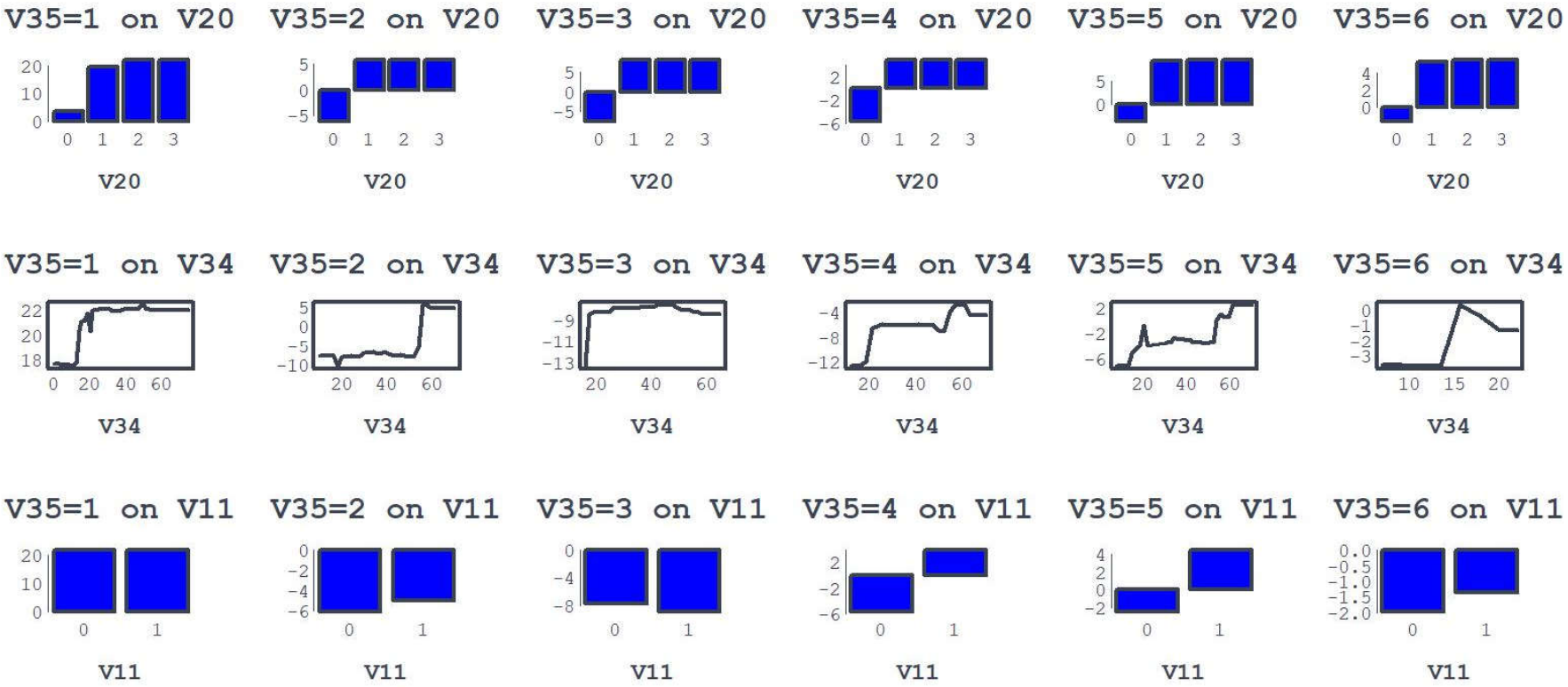
随机森林分类的变量局部重要性

```
matplot(1:34,a$localImportance,
        type = "l",las=2,main="Local Importance",
        cex.axis=.7,at=1:34,labels=names(w)[-35])
```



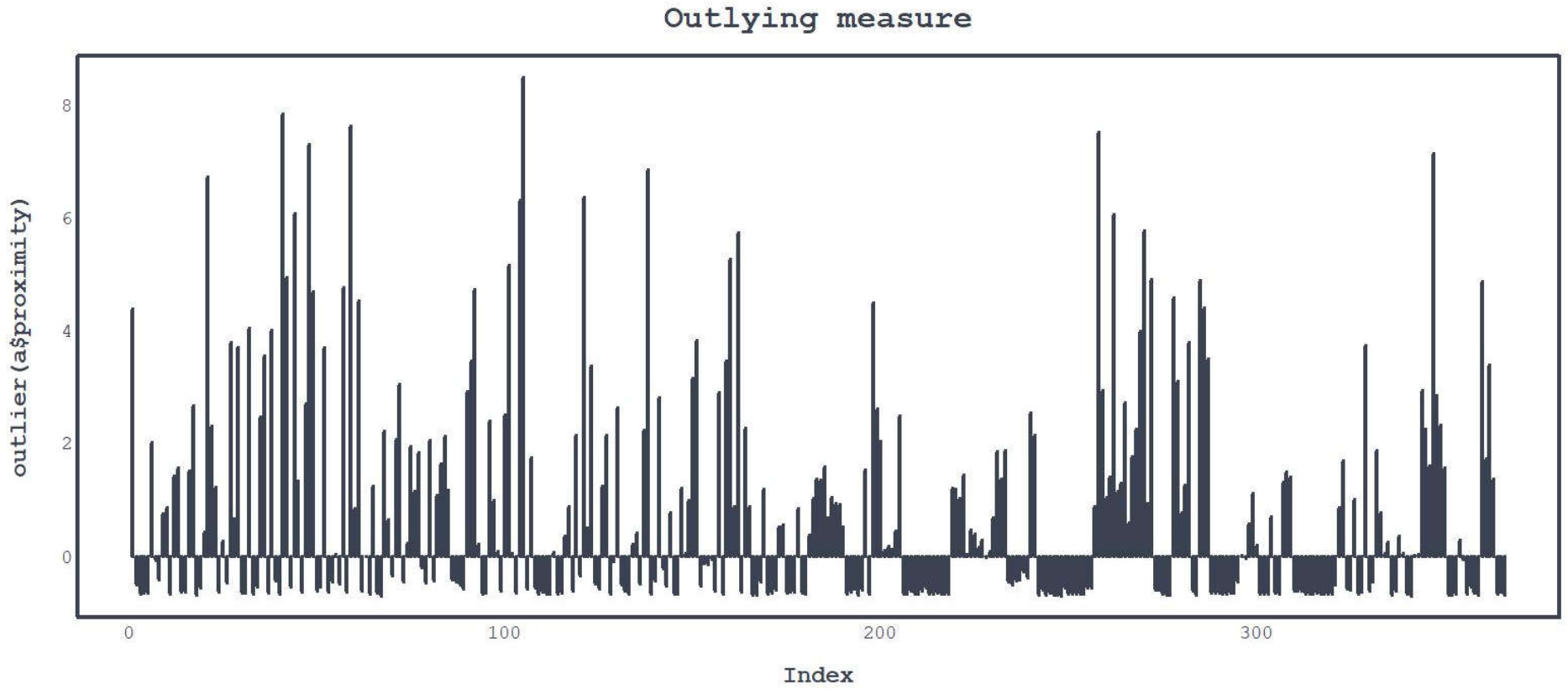
随机森林分类的变量部分依赖性

```
par(mfcol=c(3,6))
dl=levels(w$V35)
for (j in dl) {
  partialPlot(a,pred.data=w[w[,35]==j,],x.var=V20,
              main = paste0("V35=",j," on V20"))
  partialPlot(a,pred.data=w[w[,35]==j,],x.var=V34,
              main = paste0("V35=",j," on V34"))
  partialPlot(a,pred.data=w[w[,35]==j,],x.var=V11,
              main = paste0("V35=",j," on V11"))
}
```



随机森林分类的离群点

```
plot(outlier(a$proximity),type="h",main = "Outlying measure")
```



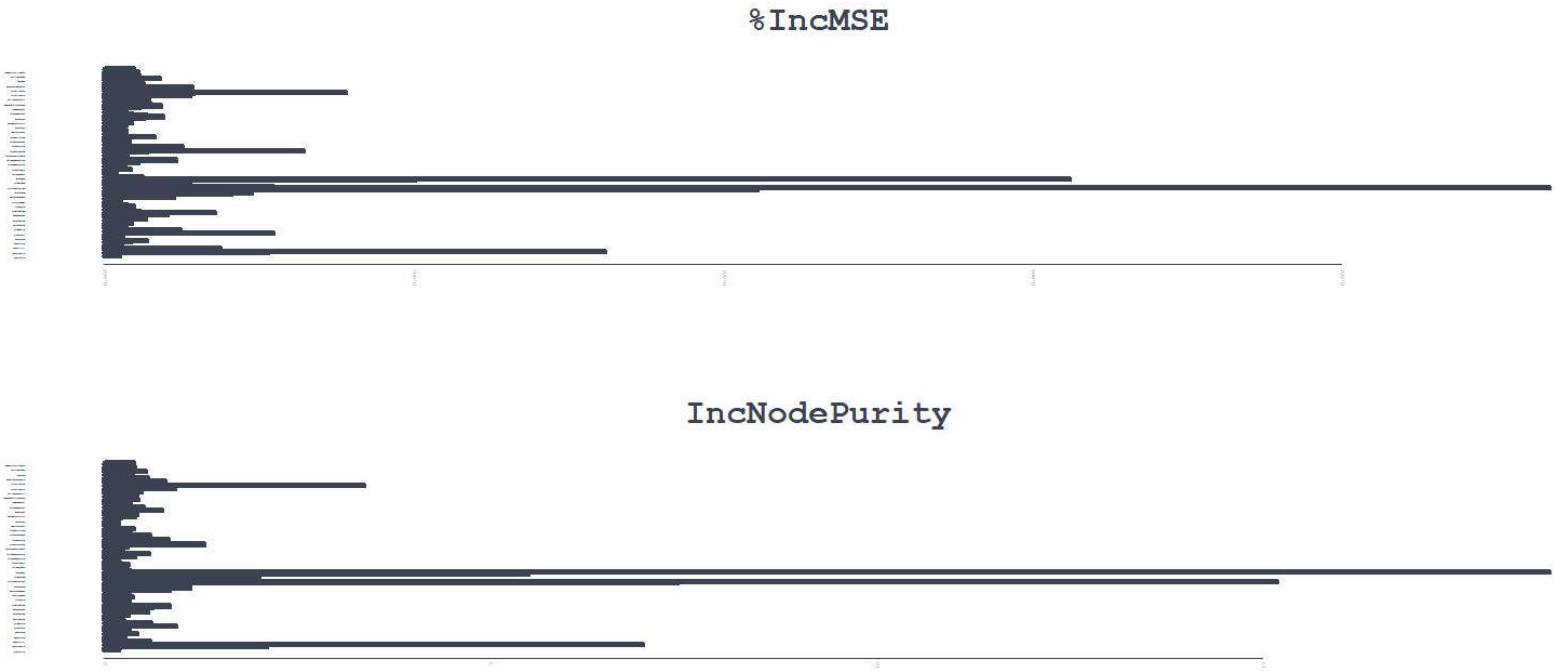
4.4.2 随机森林回归

```
w=read.csv("commun123.csv")
library(randomForest)
a=randomForest(ViolentCrimesPerPop~.,w,localImp=TRUE,proximity=TRUE)

mean((w[,123]-a$predicted)^2)/mean((w[,123]-mean(w[,123]))^2)
```

```
> mean((w[,123]-a$predicted)^2)/mean((w[,123]-mean(w[,123]))^2)
[1] 0.3334754
```

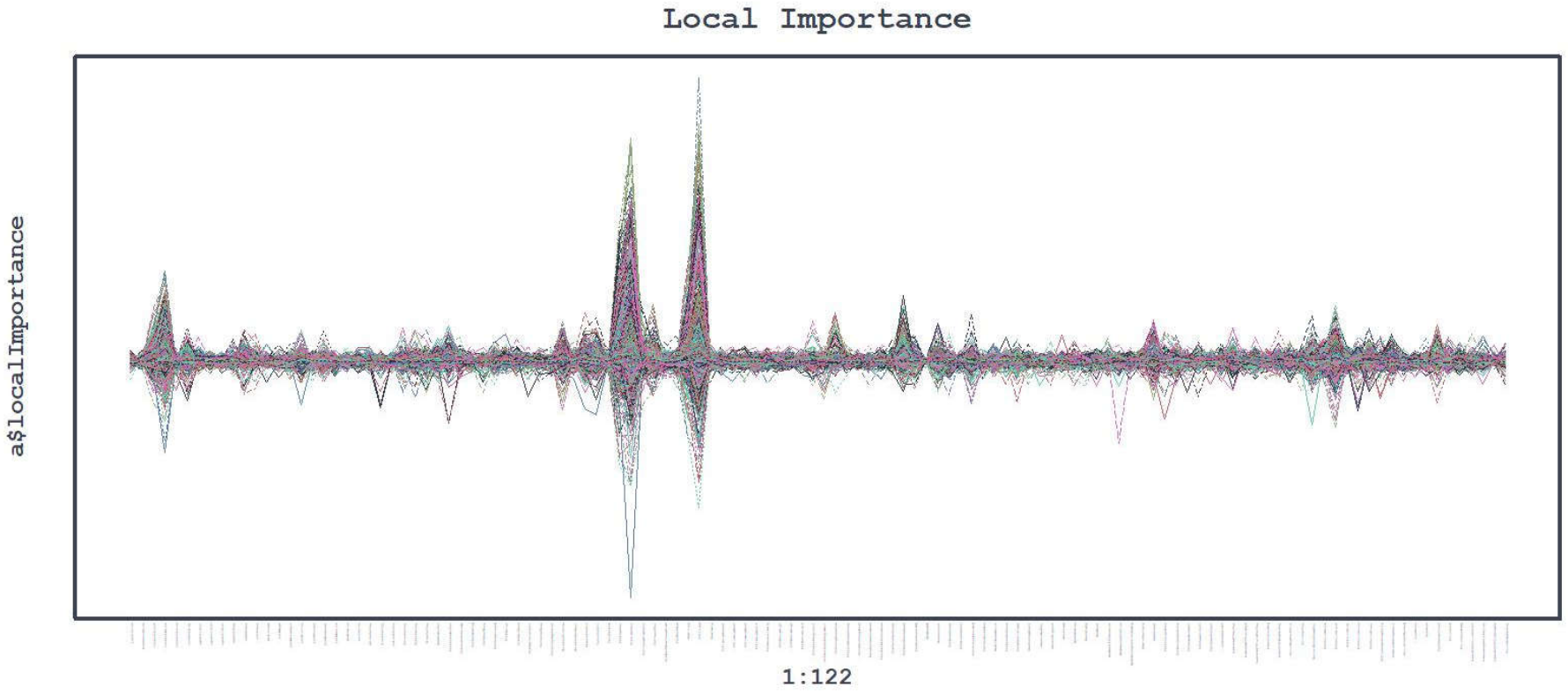
随机森林回归的变量重要性

```
par(mfrow=c(2,1))
for(i in 1:2) {
  barplot(a$importance[,i],
  main = colnames(a$importance)[i],
  las=2,cex.names =.1,horiz = TRUE,cex.axis = .3)
}
```



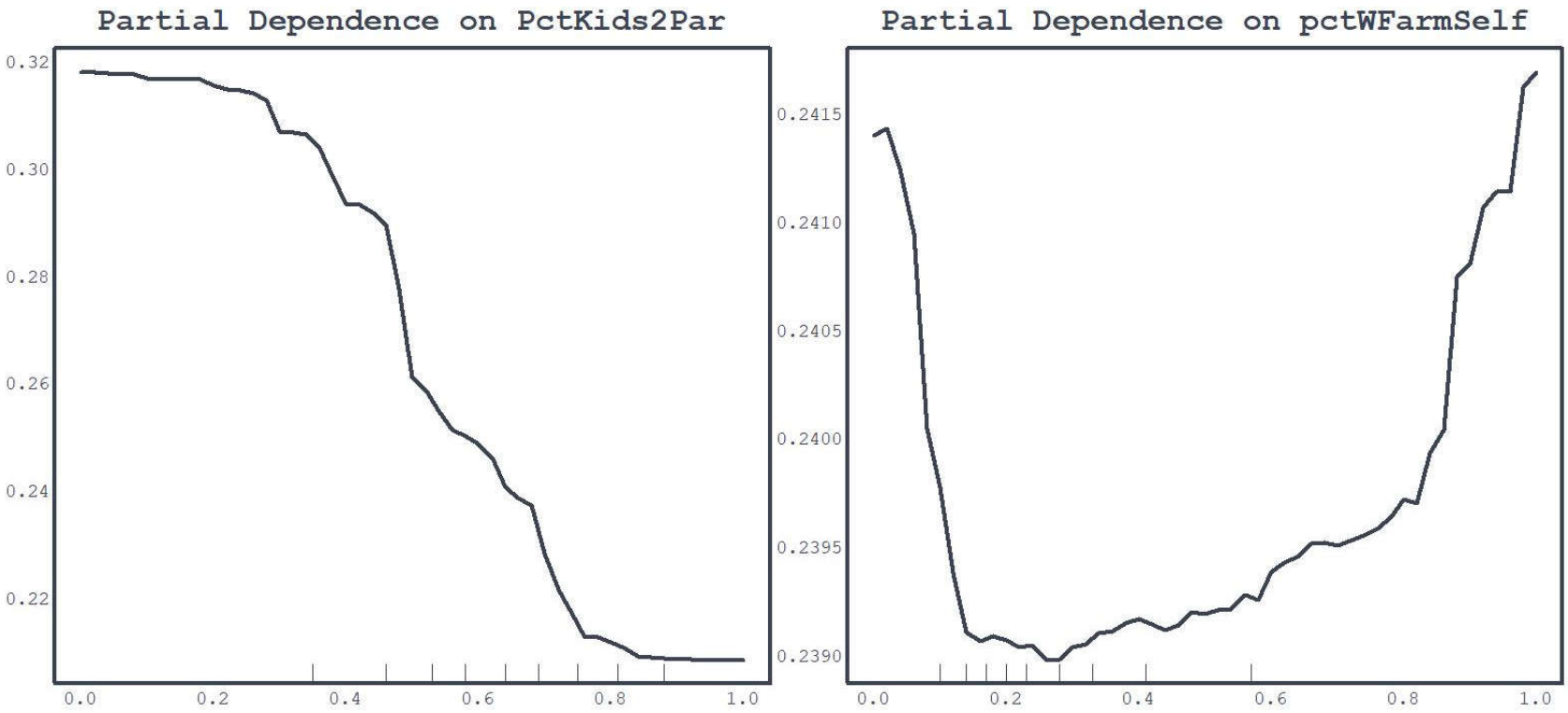
随机森林回归的变量局部重要性

```
matplot(1:122,a$localImportance,
        type = "l",las=2,main="Local Importance",
        cex.axis=.2,at=1:122,labels=names(w)[-123])
```



随机森林回归的变量部分依赖性

```
par(mfcol=c(1,2),mar=c(2,3,3,1))
partialPlot(a,pred.data=w,x.var=PctKids2Par)
partialPlot(a,pred.data=w,x.var=pctWFarmSelf)
```



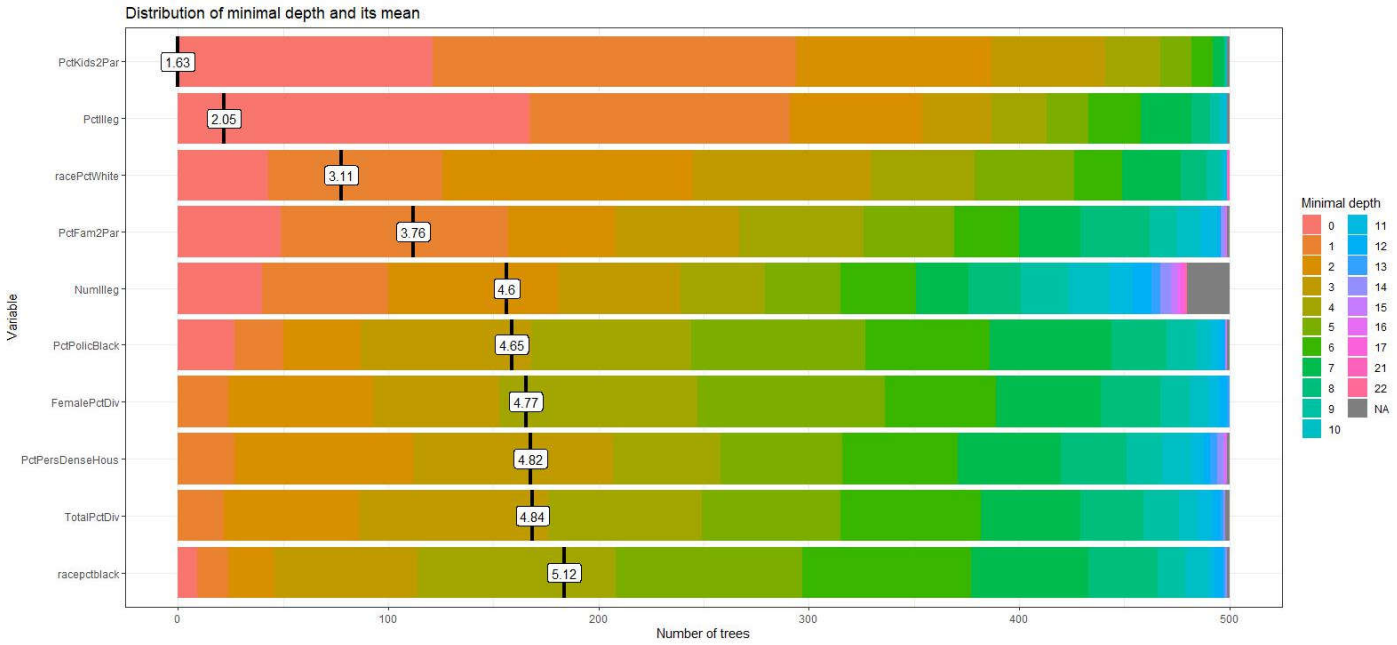
4.4.2一个解释随机森林的程序包

最小深度分布

```
library(randomForestExplainer)
library(randomForest)
set.seed(1010)
a=randomForest(ViolentCrimesPerPop~.,w,localImp=TRUE,proximity=TRUE)
min_depth_frame<-min_depth_distribution(a)
min_depth_frame %>% head(n=10)
```

	tree	variable	minimal_depth
1	1	agePct12t21	11
2	1	agePct12t29	6
3	1	agePct16t24	8
4	1	agePct65up	11
5	1	AsianPerCap	4
6	1	blackPerCap	7
7	1	FemalePctDiv	9
8	1	HispPerCap	5
9	1	householdsize	9
10	1	HousVacant	12

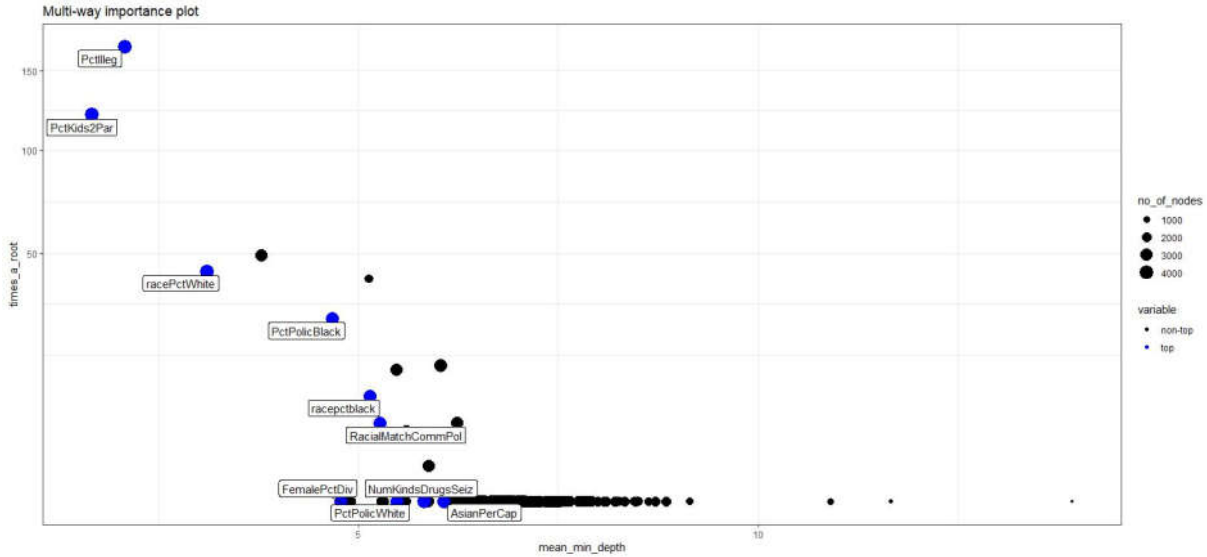
plot_min_depth_distribution(min_depth_frame, mean_sample = "relevant_trees")



各种变量重要性信息

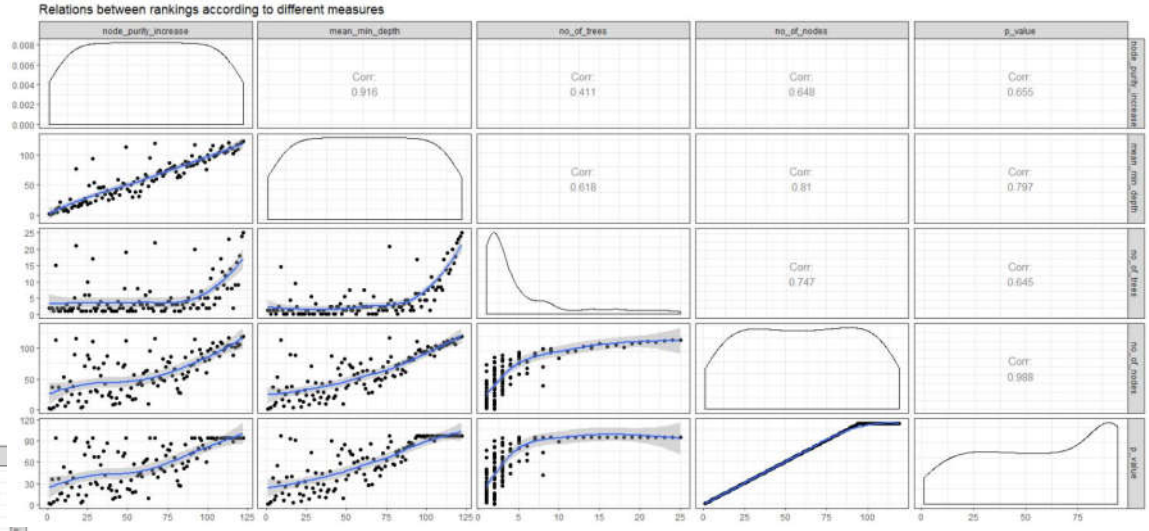
```
> importance_frame<-measure_importance(a)
> importance_frame %>% head(n=10)
        variable mean_min_depth no_of_nodes mse_increase node_purity_increase no_of_trees times_a_root      p_value
1     agePct12t21       7.147568        2593 9.048781e-05            0.2918658         497            0 9.688094e-01
2     agePct12t29       7.389424        2578 1.025571e-04            0.2328929         496            0 9.845303e-01
3     agePct16t24       7.135856        2737 5.586547e-05            0.2286502         499            0 1.752818e-01
4      agePct65up       7.249712        2620 8.496869e-05            0.2537044         498            0 9.091100e-01
5     AsianPerCap       6.066000        3652 2.929447e-04            0.6282714         500            0 2.501851e-70
6     blackPerCap       7.053712        3071 7.562162e-05            0.3116465         498            0 2.240454e-13
7    FemalePctDiv       4.774000        3391 8.167924e-04            1.0971276         500            0 2.422431e-39
8     HispPerCap       6.501856        3205 1.921646e-04            0.4200304         499            0 1.430567e-22
9  householdsize       7.915424        2437 3.074846e-05            0.2042945         496            0 9.999996e-01
10    HousVacant       5.593424        2388 5.799182e-04            0.8482842         496            0 1.000000e+00
```

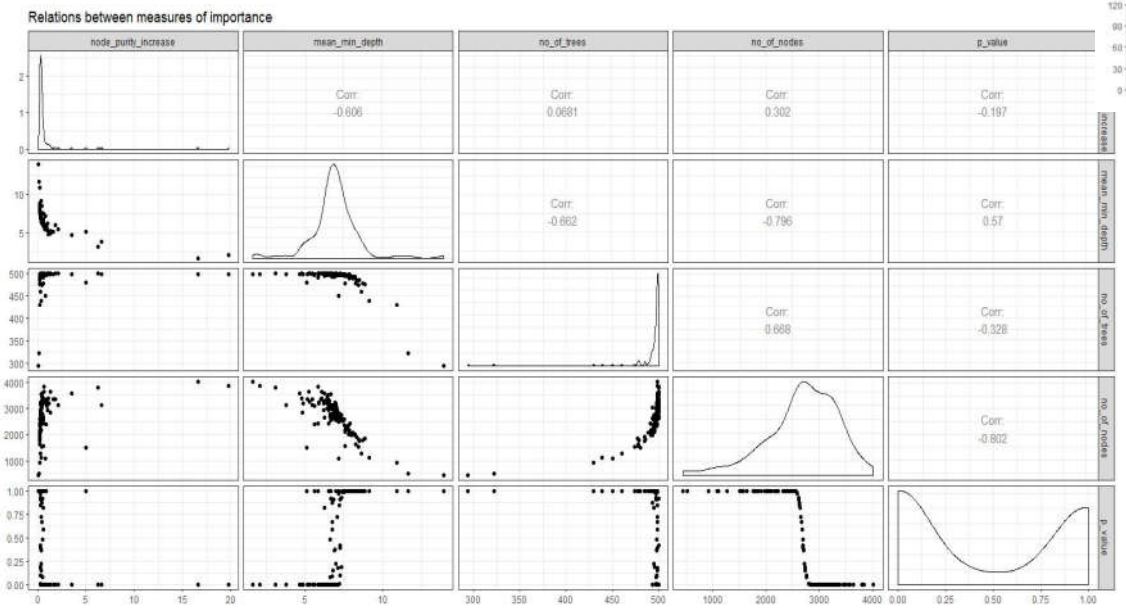
plot_multi_way_importance(importance_frame, size_measure = "no_of_nodes")



各种重要性的成对散点图及重要性排序的成对散点图

```
plot_importance_ggpairs(importance_frame)
plot_importance_rankings(importance_frame)
```

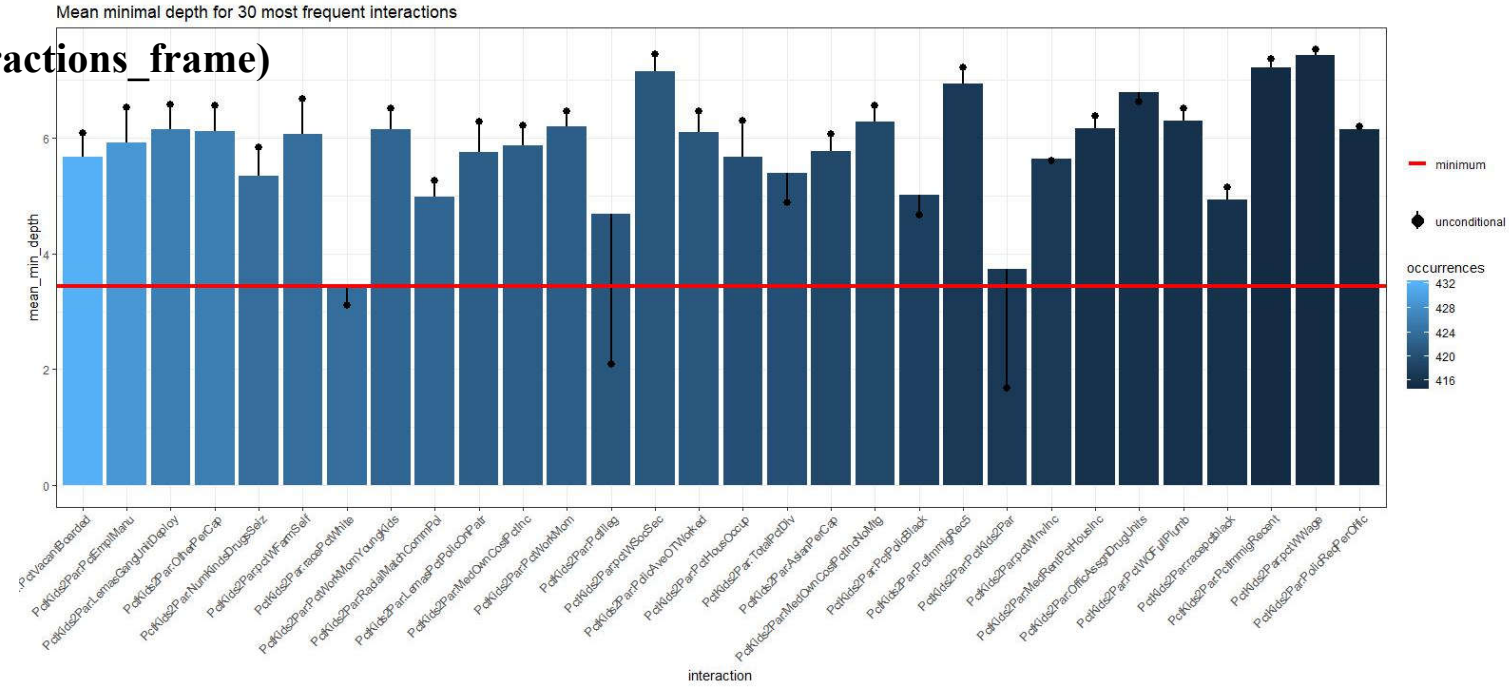




自变量交互作用

```
> vars<-important_variables(importance_frame,k=5,
+                           measures = c("mean_min_depth","no_of_trees"))
> interactions_frame<-min_depth_interactions(a,vars)
> interactions_frame%>%head()
     variable root_variable mean_min_depth occurrences                interaction uncond_mean_min_depth
1 agePct12t21  FemalePctDiv       9.203148         152  FemalePctDiv:agePct12t21               7.147568
2 agePct12t21    PctFam2Par       9.846684         246    PctFam2Par:agePct12t21               7.147568
3 agePct12t21      PctIlleg       8.532500         351      PctIlleg:agePct12t21               7.147568
4 agePct12t21   PctKids2Par       7.821059         395   PctKids2Par:agePct12t21               7.147568
5 agePct12t21 PctPolicBlack       9.364321         160 PctPolicBlack:agePct12t21               7.147568
6 agePct12t21  racePctWhite       8.961944         312  racePctWhite:agePct12t21               7.147568
```

plot_min_depth_interactions(interactions_frame)



4.5 AdaBoost（Adaptive Boosting）分类

在第一棵树之后，抽样就不是等概率的，而是对于前一棵树错判的观测值增加其被抽中的概率来为下一棵抽样。其目的是增加那些在前一棵树被错判的观测值的代表性，而增加其判对的机会。

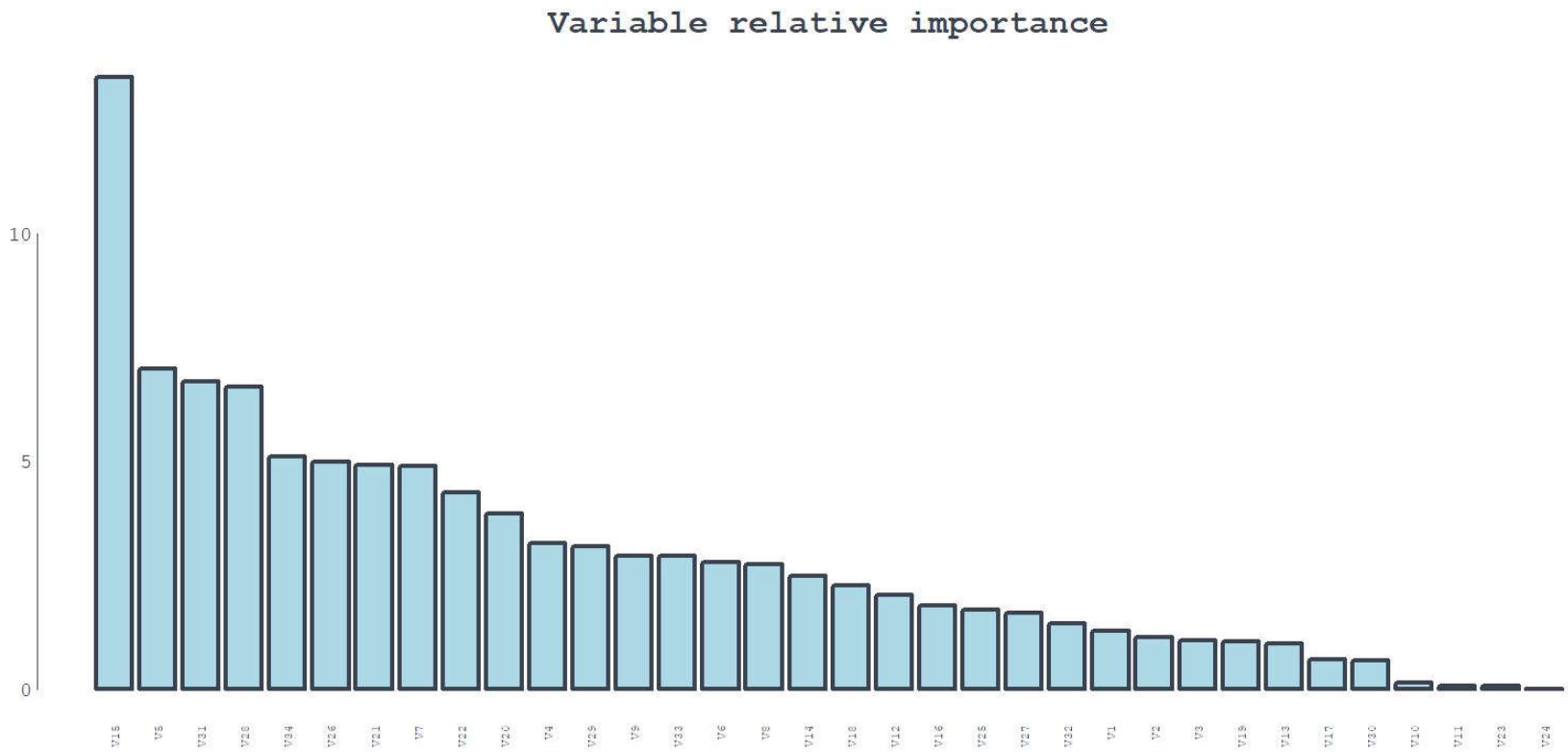
```
library(adabag)
w=read.csv("derm.csv")
for (i in (1:ncol(w))[-34]) w[,i]=factor(w[,i])
  a=boosting(V35~.,w)
predict(a,w)$confusion
```

```
> predict(a,w)$confusion
                Observed Class
Predicted Class   1   2   3   4   5   6
              1 112   0   0   0   0   0
              2   0  61   0   0   0   0
              3   0   0  72   0   0   0
              4   0   0   0  49   0   0
              5   0   0   0   0  52   0
              6   0   0   0   0   0  20
```

```
a.cv<-boosting.cv(V35~.,v=10,data=w,mfinal = 100,
                  control = rpart.control(cp=0.01))
a.cv$confusion;a.cv$error
```

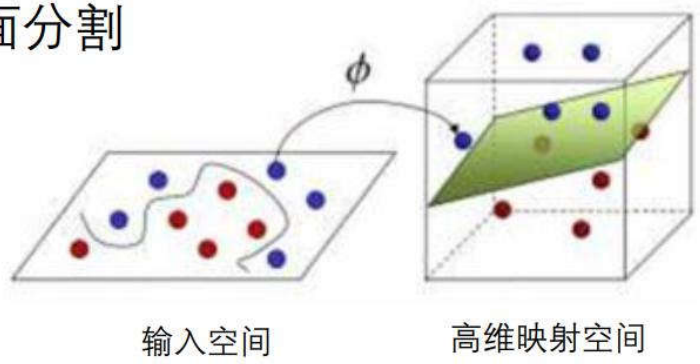
```
> a.cv$confusion;a.cv$error
                Observed Class
Predicted Class   1   2   3   4   5   6
              1 112   1   0   0   0   0
              2   0  53   0   3   0   0
              3   0   1  71   0   0   0
              4   0   4   0  46   0   0
              5   0   0   1   0  52   0
              6   0   2   0   0   0  20
[1] 0.03278689
```

importanceplot(a,cex.names=.7,las=2)

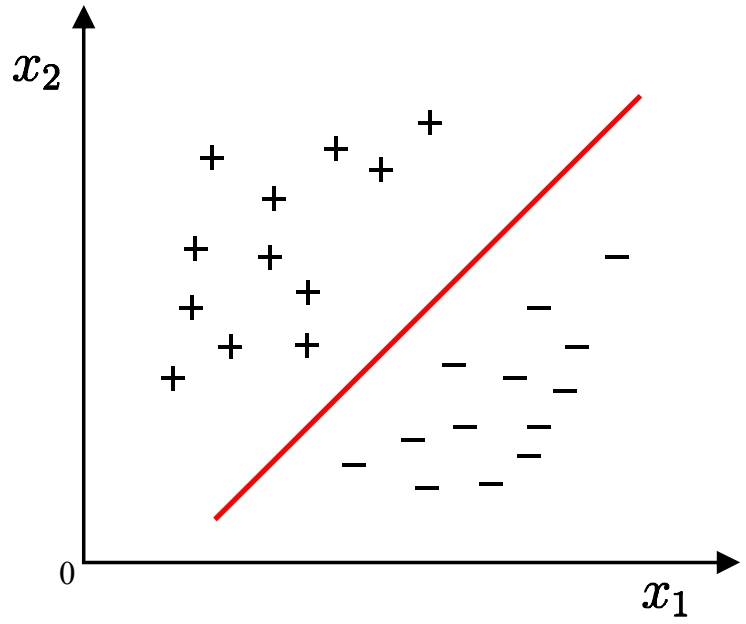


五、支持向量机
（support vector machine,SVM）

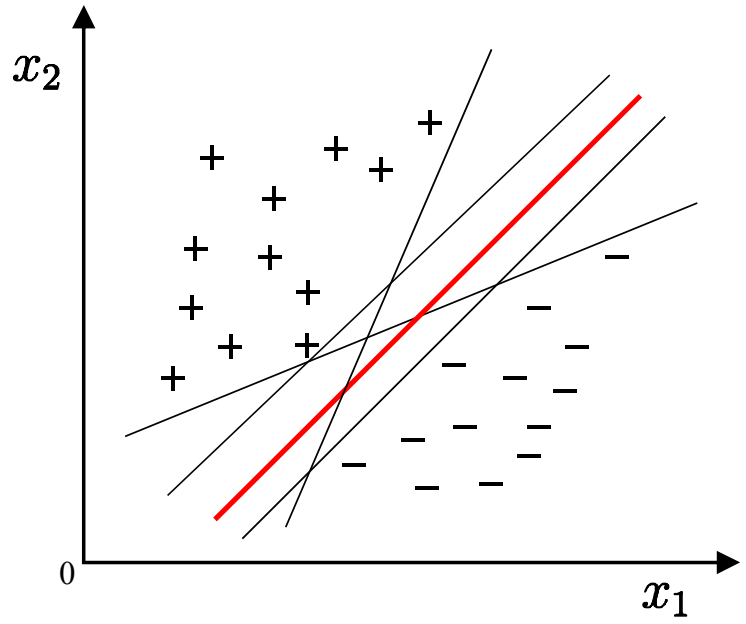
映射到高维空间，寻找超平面分割



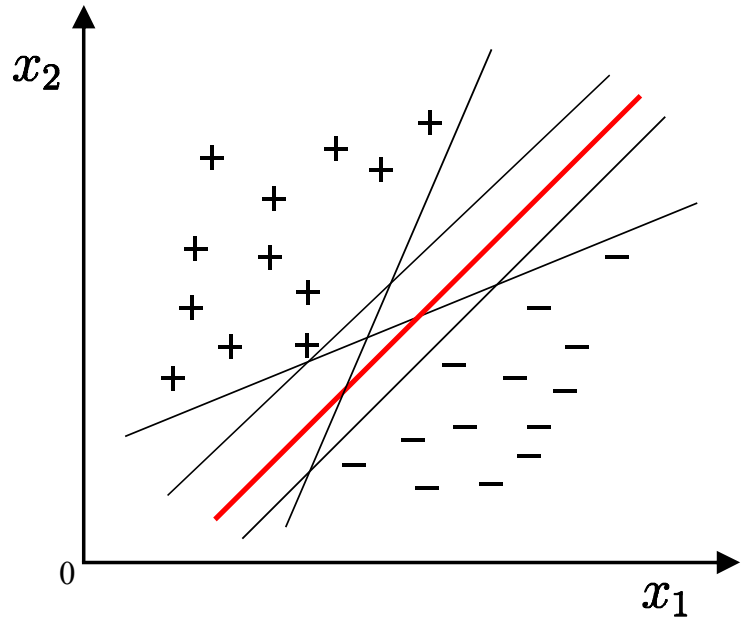
线性模型：在样本空间中寻找一个超平面，将不同类别的样本分开.



-Q:将训练样本分开的超平面可能有很多, 哪一个好呢?



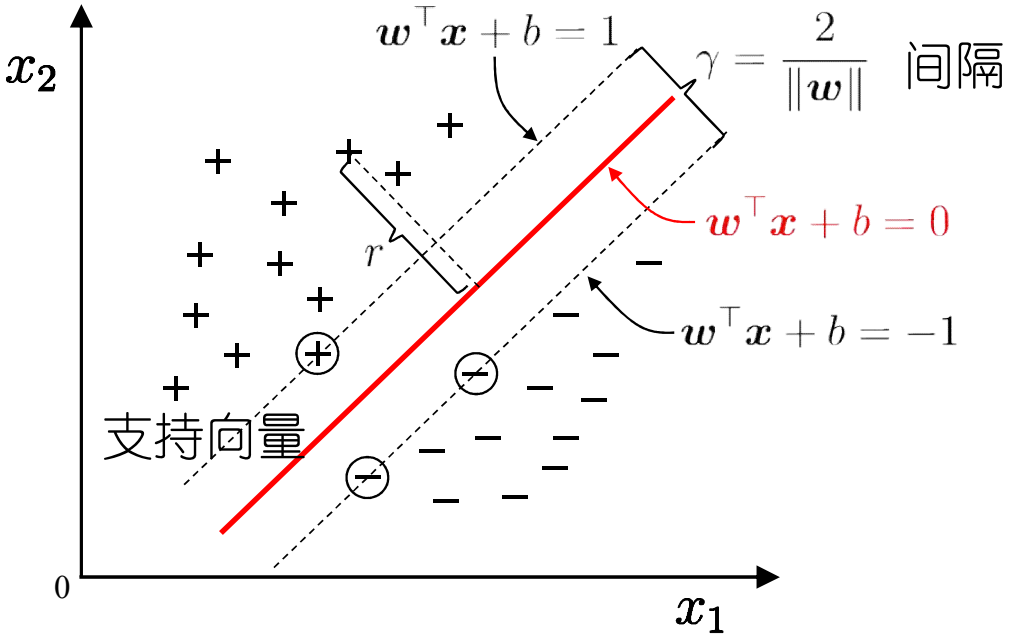
-Q:将训练样本分开的超平面可能有很多,哪一个好呢?



-A:应选择"正中间",容忍性好,鲁棒性高,泛化能力最强.

间隔与支持向量

超平面方程: $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$



支持向量机分类

```
library(kernlab)
w=read.csv("derm.csv")
for (i in (1:ncol(w))[-34]) w[,i]=factor(w[,i])
set.seed(1010)
a=ksvm(V35~.,w,cross=10)
a@error #基于训练集的误判率
a@cross #10折交叉验证的误判率
```

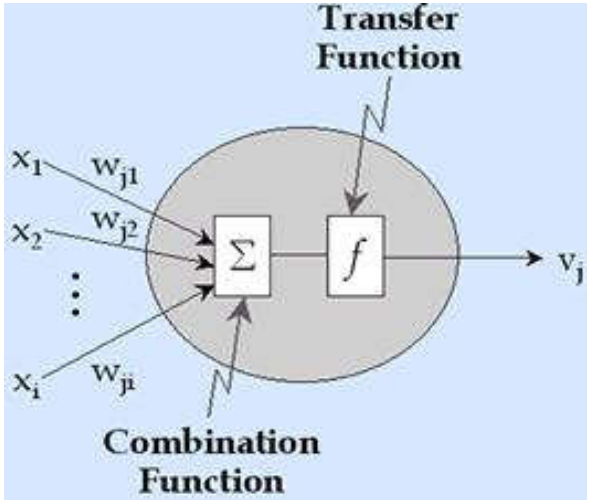
```
> a@error
[1] 0.008196721
> a@cross
[1] 0.03806306
```

支持向量机回归

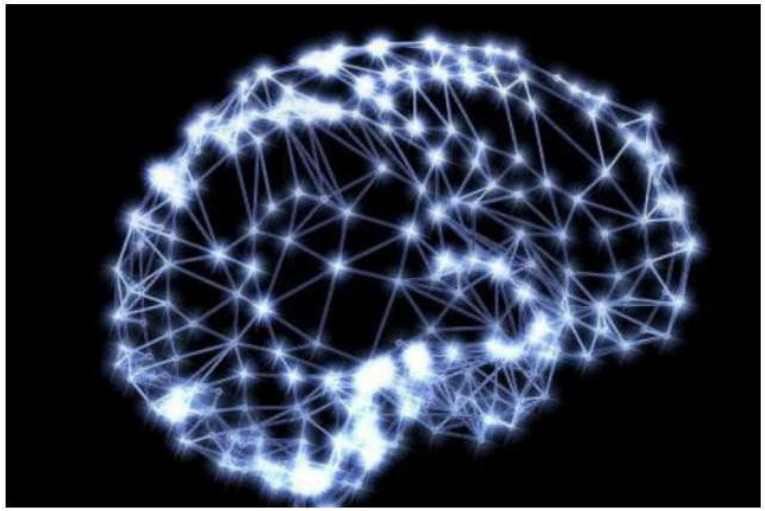
```
#支持向量机回归
library(e1071)
w=read.csv("commun123.csv")
set.seed(1010)
b=svm(ViolentCrimesPerPop~.,w,cros=10)

sum((w[,123]-b$fitted)^2)/sum((w[,123]-mean(w[,123]))^2)
b$tot.MSE/mean((w[,123]-mean(w[,123]))^2)
```

```
> sum((w[,123]-b$fitted)^2)/sum((w[,123]-mean(w[,123]))^2)
[1] 0.2011809
> b$tot.MSE/mean((w[,123]-mean(w[,123]))^2)
          [,1]
[1,] 0.3596483
```

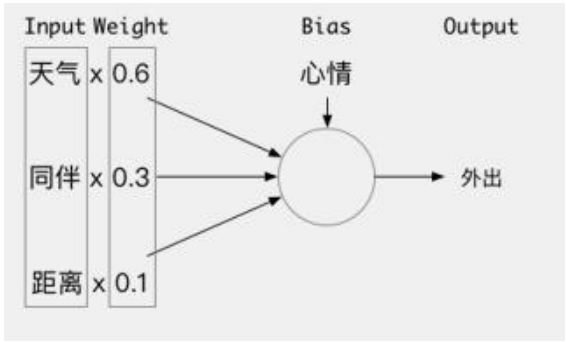
六、人工神经网络

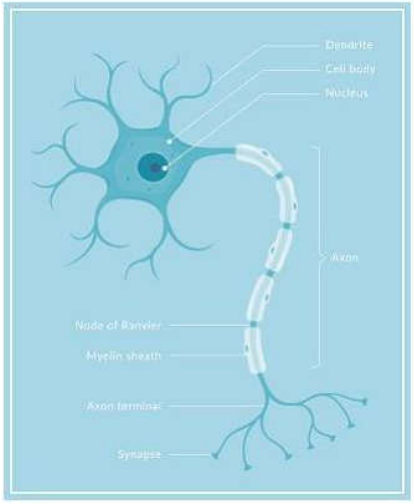

Transfer
Function
x1
wj1
x2
wj2
xi
wji
Σ
f
vj
Combination
Function

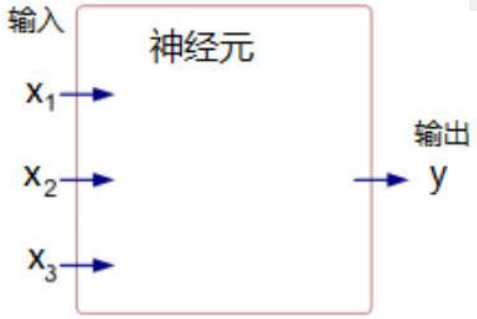
- 神经网络，也称为人工神经网络（Artificial Neural Network, ANN）
  - 20世纪80 年代开始研究，后陷入低潮
  - 随着计算能力增强和大数据出现，深度学习（也就是深度神经网络）技术呈爆发式发展
  - 目前是机器学习以及人工智能领域最重要的方法之一



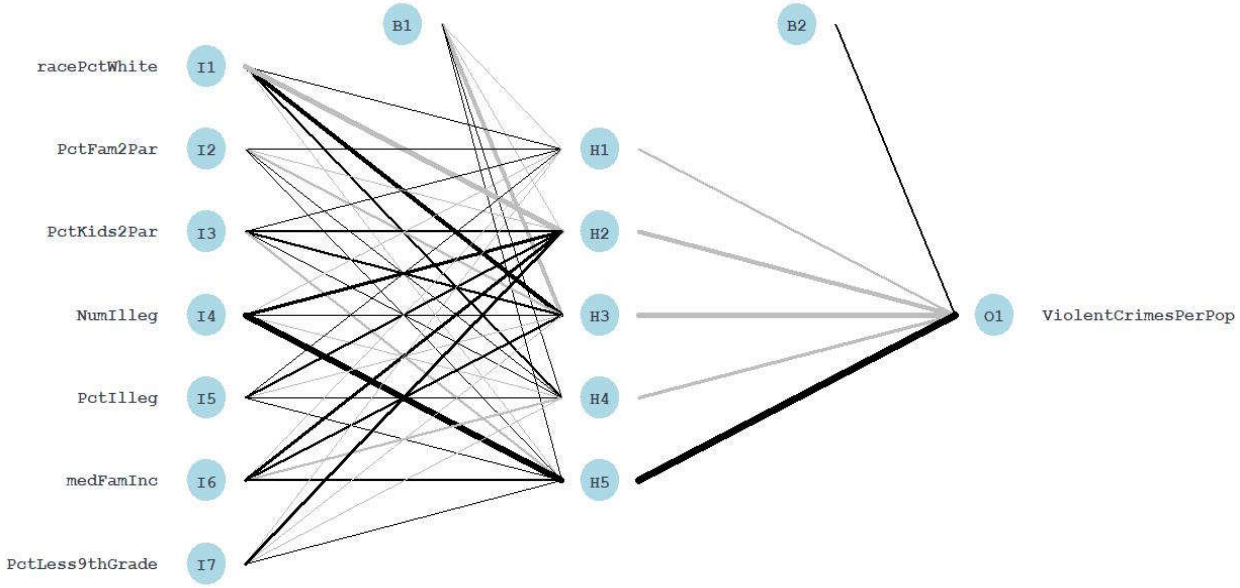
- 神经网络模拟人脑的神经网络来处理问题
  - 人脑思维基础是神经元，神经元相互连接
  - 当某个神经元接受输入，达到某种状态，它就会“兴奋”，向相连神经元发送化学物质
  - “人造神经元”模型，称为**感知器**（percepton）







```
library(nnet)
w=read.csv("commun123.csv")
set.seed(1010)
sel=c(4,44,45,50,51,20,30)#选取7个自变量
w1=w[,c(sel,123)]#因变量是第123个
a=nnet(ViolentCrimesPerPop~.,data=w1,method="nnet",
       maxit=1000,size=5,decay=0.01,trace=F)
library(NeuralNetTools)
plotnet(a,pad_x=.7)
```



神经网络有三层，最左边的是输入层，该层包含7个自变量相对应的7个节点(I1,I2,…,I7);中间一层为隐藏层，5个节点：H1,H2,…,H5;最右边是输出层，只有一个节点（O1），与回归的因变量对应。

本例中, 对于每个观测值, 假定输入层 7 个节点的自变量值为 $x_1, x_2, \cdots, x_7$, 下面描述我们所用的神经网络是怎么进行训练的:

(1) 在输入层首先随意对每个隐藏层确定一组权重 (注意: 一共 5 个隐藏层节点):

$$w_{1k}, w_{2k}, \cdots, w_{7k}, \ k = 1, 2, \cdots, 5.$$

每一个权重的下标 ik 表示第 i 个变量及第 k 个隐藏层节点, 此外再加上一个常数项 w_{0k}.

(2) 根据得到的权重可以组成 5 个线性组合 (加权平均):

$$w_{0k} + w_{1k}x_1 + \cdots + w_{7k}x_7, \ k = 1, 2, \cdots, 5.$$

(3) 然后把这 5 个线性组合的值经过一个函数变换 (记为函数 ϕ) 加到 5 个隐藏层:

$$z_k = \phi(w_{0k} + w_{1k}x_1 + \cdots + w_{7k}x_7), \ k = 1, 2, \cdots, 5.$$

因此, $z_1, z_2, \cdots, z_5$ 为 5 个隐藏层节点的值. 这个函数 ϕ 称为**激活函数** (activation function), 一般是从实轴 $(-\infty, \infty)$ 到 $[0, 1]$(或 $[-1, 1]$) 的映射. 比如 $\phi(x) = 1/(1 + \mathrm{e}^{-x}), \phi(x) = \tan h(x)$ 等, 当然也有其他映射范围的激活函数, 比如 $\phi(x) = \max(0, x)$.

(4) 对于隐藏层到输出层也类似于输入层到隐藏层设定一组权重 $w_1^{(h)}, w_2^{(h)}, \cdots, w_5^{(h)}$ (因为我们例子的输出层只有一个节点), 外加一个常数项 $w_0^{(h)}$. 组成一个线性组合, 再通过一个激活函数 (记为 $\phi^{(h)}$) 的映射, 得到:

$$y = \phi^{(h)}(w_0^{(h)} + w_1^{(h)}z_1 + \cdots + w_5^{(h)}z_5).$$

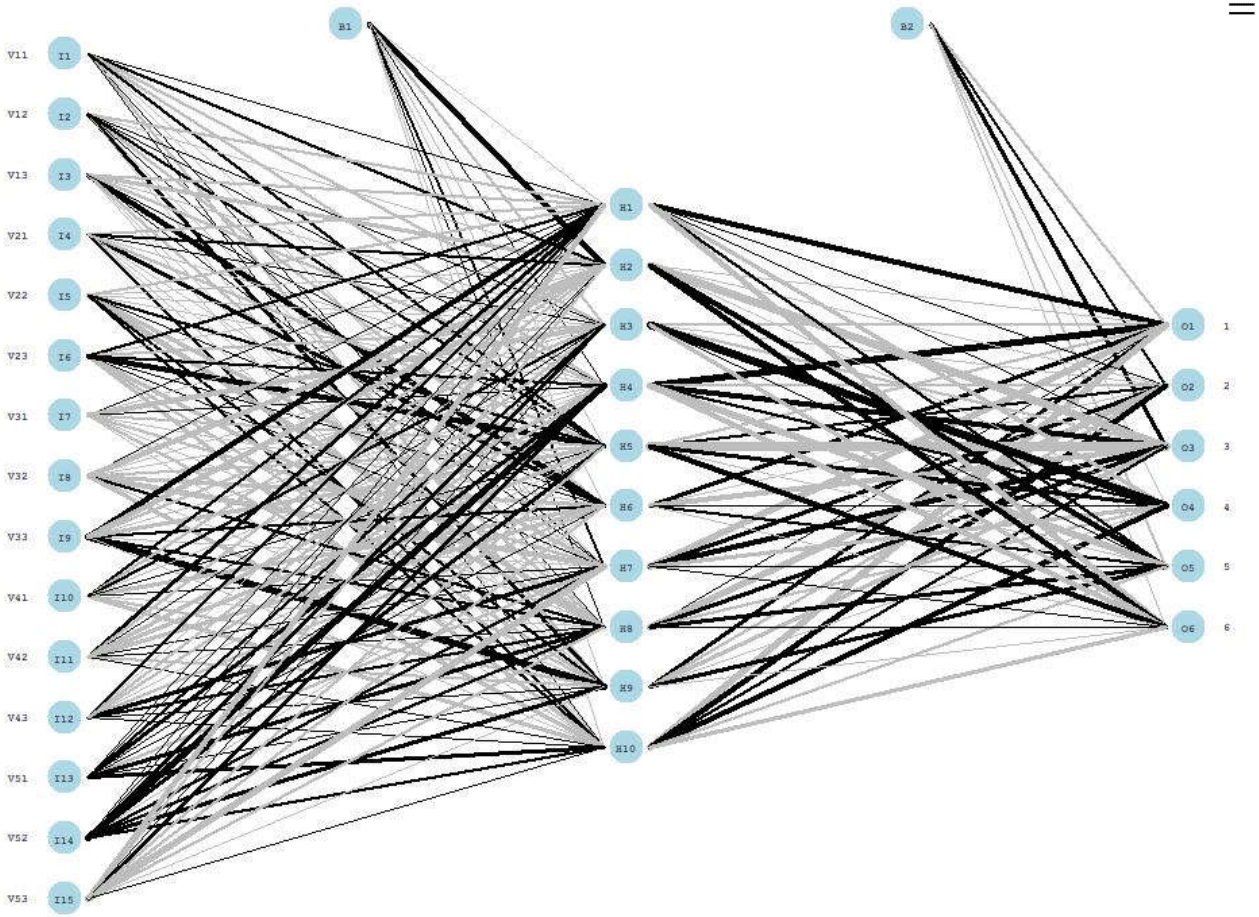
(5) 这样得到一个 y 的值, 再和真实的 y 值做比较, 就得到误差以及误差的特点 (比如它们的带符号差值), 并以此反馈修正前面两级的权重, 回到前面第 2 步. 不断重复, 直到达到预定的精确度或预定的步数限制为止. 这种反馈误差的方法称为**误差的反向传播** (back propagation of error), 即著名的 BP 算法.

图 10.1 中的每条连线代表着一个权重, 一个新观测值的预测值是自变量通过两层激活函数变换的加权平均. 上面第 3、4 步两层公式结合起来为:

$$y = \phi^{(h)}\left(\sum_{k=1}^{5} w_k^{(h)} z_k + w_0^{(h)}\right) = \phi^{(h)}\left\{\sum_{k=1}^{5} w_k^{(h)}\left[\phi\left(\sum_{i=1}^{7} w_{ik}x_i + w_{0k}\right)\right] + w_0^{(h)}\right\} \quad (10.1)$$

其中的权重就是通过反复迭代修正之后得到的, 这个过程比较耗时, 一旦权重学习出来了, 得到新的观测值的预测就非常快了.

```
u=read.csv("derm.csv")[,c(1:5,35)]
for (i in 1:ncol(u)) u[,i]=factor(u[,i])
set.seed(1010)
b=nnet(V35~.,data=u,method="nnet",
       maxit=1000,size=10,decay=0.01,trace=F)
plotnet(b,pad_x = 1,circle_cex = 3,cex_val = .5)
```



$$y_i = \phi^{(h)}\left(\sum_{k=1}^{10} \omega_{kj}^{(h)} z_{kj} + \omega_{0j}^{(h)}\right)$$

$$= \phi^{(h)}\left\{\sum_{k=1}^{10} \omega_{kj}^{(h)}\left[\phi\left(\sum_{i=1}^{15} \omega_{ik} x_i + \omega_{0k}\right)\right] + \omega_{0j}^{(h)}\right\}$$

程序中，选项maxit为最大迭代次数，size为隐藏层节点个数，decay为修正权重时的尺度，trace代表是否输出迭代过程。

对神经网络精度很重要的两个选项为size和decay。

神经网络分类

```
library(e1071)
w=read.csv("derm.csv")
for (i in (1:ncol(w))[-34]) w[,i]=factor(w[,i])
tune.model=tune.nnet(V35~.,data=w,size=1:9)
summary(tune.model)
> summary(tune.model)

Parameter tuning of 'nnet':

- sampling method: 10-fold cross validation

- best parameters:
 size
    9

- best performance: 0.1362763

- Detailed performance results:
  size     error dispersion
1    1 0.5095796 0.09525170
2    2 0.3671321 0.09042601
3    3 0.3192492 0.09392662
4    4 0.2267117 0.06925070
5    5 0.2266967 0.07711457
6    6 0.2011411 0.08364854
7    7 0.1480030 0.04097006
8    8 0.1446396 0.04697410
9    9 0.1362763 0.05103555
```

```
Z=10;D=35;mm=Fold(Z,w,D)
Pr=data.frame(rf=w$V35,net=w$V35)
for (i in 1:Z) {
  Pr$rf[mm[[i]]]=
    randomForest(V35~.,w[-mm[[i]],]) %>% predict(w[mm[[i]],])
  Pr$net[mm[[i]]]=
    a=nnet(V35~.,w[-mm[[i]],],method="nnet",
           maxit=100,size=9,trace=F) %>%
    predict(w[mm[[i]],],type="class")

}
> table(w$V35,Pr$rf);mean(w$V35!=Pr$rf)

    1  2  3  4  5  6
1 112  0  0  0  0  0
2   1 57  0  3  0  0
3   0  0 72  0  0  0
4   0  5  0 44  0  0
5   0  0  0  0 52  0
6   0  0  0  0  0 20
[1] 0.02459016
> table(w$V35,Pr$net);mean(w$V35!=Pr$net)

    1  2  3  4  5  6
1 107  2  0  0  2  1
2   4 39  2  7  8  1
3   1  1 67  2  0  1
4   0  6  1 40  2  0
5   2  1  5  3 39  2
6   0  1  1  0  3 15
[1] 0.1612022
```

神经网络回归

案例

Package 'mlbench'

February 20, 2015

Version 2.1-1

Title Machine Learning Benchmark Problems

Date 2010-12-10

Author Friedrich Leisch and Evgenia Dimitriadou.

Maintainer Friedrich Leisch <Friedrich.Leisch@R-project.org>

Description A collection of artificial and real-world machine learning benchmark problems, including, e.g., several data sets from the UCI repository.

Depends R (>= 2.10)

License GPL-2

Suggests lattice

ZipData No

Repository CRAN

Date/Publication 2012-07-10 11:51:32

NeedsCompilation yes

BostonHousing *Boston Housing Data*

Description

Housing data for 506 census tracts of Boston from the 1970 census. The dataframe BostonHousing contains the original data by Harrison and Rubinfeld (1979), the dataframe BostonHousing2 the corrected version with additional spatial information (see references below).

Usage

```
data(BostonHousing)
data(BostonHousing2)
```

Format

The original data are 506 observations on 14 variables, medv being the target variable:

crim	per capita crime rate by town
zn	proportion of residential land zoned for lots over 25,000 sq.ft
indus	proportion of non-retail business acres per town
chas	Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
nox	nitric oxides concentration (parts per 10 million)
rm	average number of rooms per dwelling
age	proportion of owner-occupied units built prior to 1940
dis	weighted distances to five Boston employment centres
rad	index of accessibility to radial highways
tax	full-value property-tax rate per USD 10,000
ptratio	pupil-teacher ratio by town
b	$1000(B - 0.63)^2$ where B is the proportion of blacks by town
lstat	percentage of lower status of the population
medv	median value of owner-occupied homes in USD 1000's

The corrected data set has the following additional columns:

cmedv	corrected median value of owner-occupied homes in USD 1000's
town	name of town
tract	census tract
lon	longitude of census tract
lat	latitude of census tract

Source

The original data have been taken from the UCI Repository Of Machine Learning Databases at

- http://www.ics.uci.edu/~mlearn/MLRepository.html,

the corrected data have been taken from Statlib at

- http://lib.stat.cmu.edu/datasets/

See Statlib and references there for details on the corrections. Both were converted to R format by Friedrich Leisch.

References

Harrison, D. and Rubinfeld, D.L. (1978). Hedonic prices and the demand for clean air. *Journal of Environmental Economics and Management*, **5**, 81–102.

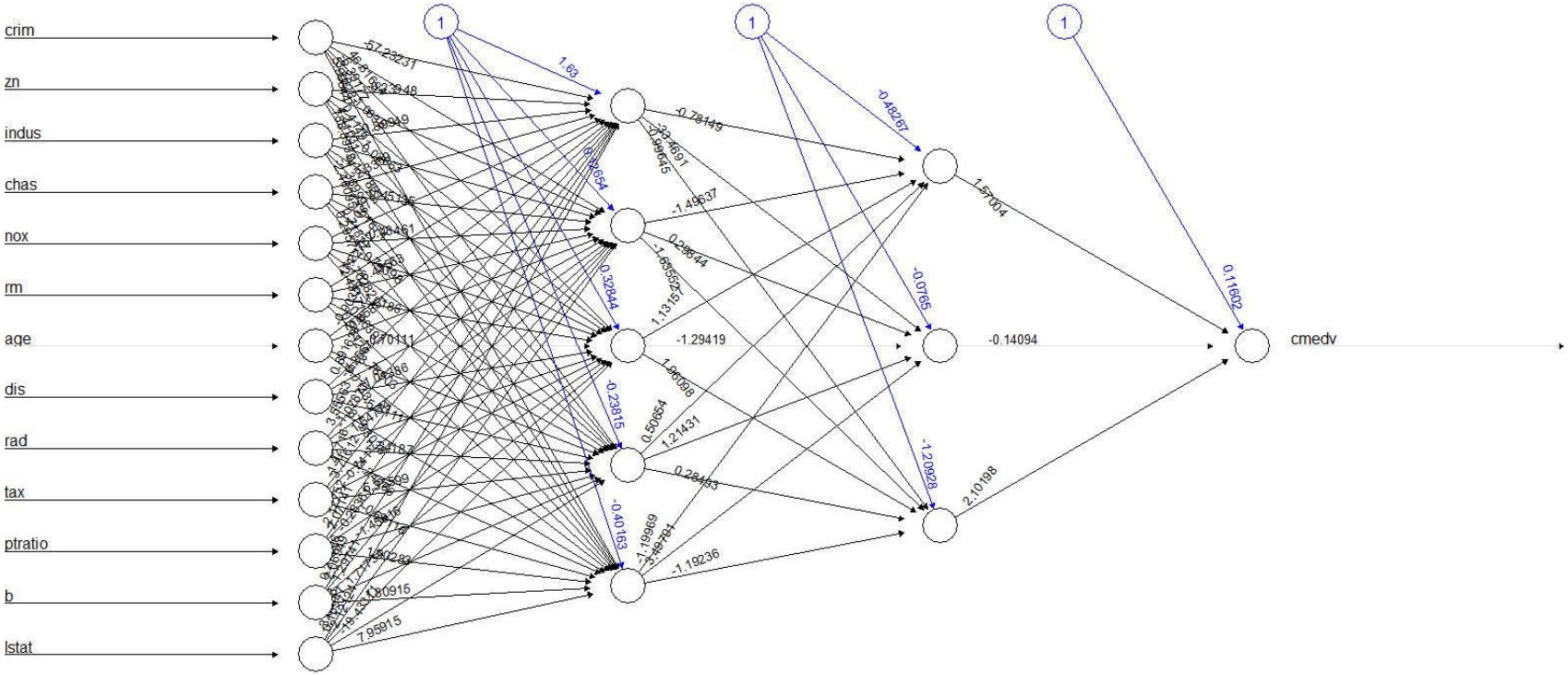
Gilley, O.W., and R. Kelley Pace (1996). On the Harrison and Rubinfeld Data. *Journal of Environmental Economics and Management*, **31**, 403–405. [Provided corrections and examined censoring.]

Newman, D.J. & Hettich, S. & Blake, C.L. & Merz, C.J. (1998). UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

Pace, R. Kelley, and O.W. Gilley (1997). Using the Spatial Configuration of the Data to Improve Estimation. *Journal of the Real Estate Finance and Economics*, **14**, 333–340. [Added georeferencing and spatial estimation.]

神经网络回归

```
w=read.csv("BostonHousing2.csv")
s.w=lapply(w[,-(1:5)], function(x){(x-min(x))/(max(x)-min(x))})%>%as.data.frame()

library(neuralnet)
nm<-names(s.w)
f<-as.formula(paste("cmedv ~",paste(nm[!nm %in% "cmedv"], collapse = " + ")))
res<-neuralnet(f,data=s.w,hidden=c(5,3),linear.output=T)
plot(res)
```



神经网络回归的10折交叉验证

```
library(randomForest)
n=nrow(s.w)
Pr=data.frame(rf=rep(0,n),net=rep(0,n))
Z=10;set.seed(1010)
I=sample(rep(1:Z,ceiling(n/Z)))[1:n]
for (i in 1:Z) {
  Pr$rf[I==i]=randomForest(f,w[I!=i,-(1:5)])%>%
    predict(w[I==i,-(1:5)])
  nn=neuralnet(f,data=s.w[I!=i,],hidden = c(5,3),linear.output = T)
  Pr$net[I==i]=compute(nn,s.w[I==i,-1])$net.result
}
Pr$net=Pr$net*(max(w$cmedv)-min(w$cmedv))+min(w$cmedv)#变换原先的尺度
RSS=sum((w$cmedv-mean(w$cmedv))^2)
sum((w$cmedv-Pr$rf)^2)/RSS
sum((w$cmedv-Pr$net)^2)/RSS
```

```
> sum((w$cmedv-Pr$rf)^2)/RSS
[1] 0.1167826
> sum((w$cmedv-Pr$net)^2)/RSS
[1] 0.1359136
```

七、朴素贝叶斯

7.1 朴素贝叶斯原理

1、每个数据样本用一个n维特征向量$X=\{x_1, x_2, \ldots, x_n\}$表示，分别描述对$n$个属性$A_1, A_2, \ldots, A_n$样本的$n$个度量。

2、假定有m个类$C_1, C_2, \ldots, C_m$。给定一个未知的数据样本X，分类法将预测X属于具有最高后验概率（条件X下）的类。即是说，朴素贝叶斯分类将未知的样本分配给类C_i，

当且仅当

$$P(C_i|X) > P(C_j|X), 1 \le j \le m, j \ne i$$

根据贝叶斯定理

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$$

因此，由于$P(X)$对于所有类为常数，只需要$P(X|C_i)P(C_i)$最大即可。

3、假定属性值相互条件独立，即在属性间不存在依赖关系，这样，

$$P(X|C_i)=\prod_{k=1}^{n}P(x_k|C_i)$$

概率$P(x_k|C_i)$可以由训练样本估值，其中

（1）如果A_k是离散型属性，则$P(x_k|C_i)=s_{ik}/s_i$，其中s_{ik}是在属性A_k上具有x_k的类C_i的训练样本数，而s_i是C_i中的训练样本数。

（2）如果A_k是连续型属性，则通常假定该属性服从高斯分布。因而，

$$P(x_k \mid C_i)=g(x_k,\mu_{C_i},\sigma_{C_i})=\frac{1}{\sqrt{2\pi\sigma_{C_i}}}e\frac{(x_k-\mu_{C_i})^2}{2\sigma_{C_i}^2}$$

其中，给定类C_i的训练样本属性A_k的值，$g(x_k,\mu_{C_i},\sigma_{C_i})$是属性$A_k$的高斯密度函数。

4、为对未知样本X分类，对每个类C_i，计算$P(X|C_i)P(C_i)$。样本被指派到类C_i，当且仅当

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j)$$

即是说，X被指派到$P(X|C_i)P(C_i)$最大的类。

7.2 朴素贝叶斯方法分类

```
w=read.csv("derm.csv")
for (i in (1:ncol(w))[-34]) w[,i]=factor(w[,i])
library(e1071);library(randomForest)

Z=10;D=35
mm=Fold(Z,w,D)
Pr=data.frame(rf=w$V35,NB=w$V35)
for (i in 1:Z) {
  Pr$rf[mm[[i]]]=
    randomForest(V35~.,w[-mm[[i]],])%>%
    predict(w[mm[[i]],])
  Pr$NB[mm[[i]]]=
    a=naiveBayes(V35~.,w[-mm[[i]],])%>%
    predict(w[mm[[i]],])
}
table(w$V35,Pr$rf);mean(w$V35!=Pr$rf)
table(w$V35,Pr$NB);mean(w$V35!=Pr$NB)
```

```
> table(w$V35,Pr$rf);mean(w$V35!=Pr$rf)

     1   2   3   4   5   6
  1 112   0   0   0   0   0
  2   1  58   0   2   0   0
  3   0   0  72   0   0   0
  4   0   5   0  44   0   0
  5   0   0   0   0  52   0
  6   0   0   0   0   0  20
[1] 0.02185792
> table(w$V35,Pr$NB);mean(w$V35!=Pr$NB)

     1   2   3   4   5   6
  1 112   0   0   0   0   0
  2   0  56   0   5   0   0
  3   0   0  72   0   0   0
  4   0   2   0  47   0   0
  5   0   0   0   0  52   0
  6   0   0   0   0   0  20
[1] 0.01912568
```

八、K最邻近方法

8.1 基本原理

- K最近邻方法分类
  - 空间中训练集点的类型都是已知的,如果一个新的观测值来到它们之中,则看距离其最近的k个点中哪种类型多,就把它判为哪一类.当然该方法通常还根据距离远近加权,离新观测值越近的点,投票的权重就越大。
- K最近邻方法的回归
  - 空间中训练集的点都有数值,如果一个新的观测值来到它们之中,则用距离它最近的k个点的值的加权平均作为它的值。
- 事先确定的选择：一个是距离度量的选择,另一个是k的选择,再有就是权重的选择。当然这些都有默认值(缺省值),比如,下面要使用的程序包kknn中,几个默认选项为:k=7、距离 distance=2(欧氏距离)、权重为 kernel=“optimal”。在程序的各种选项中,最重要的当然是k的选择。

8.2 K最邻近分类

案例

SCADI Data Set

Download: Data Folder, Data Set Description

Abstract: First self-care activities dataset based on ICF-CY.

Data Set Characteristics:	Multivariate	**Number of Instances:**	70	**Area:**	Life
Attribute Characteristics:	N/A	**Number of Attributes:**	206	**Date Donated**	2018-04-14
Associated Tasks:	Classification, Clustering	**Missing Values?**	N/A	**Number of Web Hits:**	26512

Source:

--Creators: S.M.M. Fatemi Bushehri, Moslem Dehghanizadeh, Shokoofeh Kalantar, Mohsen Sardari Zarchi

* S.M.M. Fatemi Bushehri: Department of Software Engineering, Yazd Branch, Islamic Azad University, Yazd, Iran
* Moslem Dehghanizadeh: Department of Occupational Therapy, School of Rehabilitation, Iran University of Medical Sciences, Tehran, Iran
* Shokoofeh Kalantar: Student Senior Counseling & Guidance, Islamic Azad University, Department of Human Science, Yazd, Iran
* Mohsen Sardari Zarchi: Department of Computer Engineering, Meybod University, Meybod, Iran

-- Donator: S.M.M. Fatemi Bushehri

Data Set Information:

This dataset contains 206 attributes of 70 children with physical and motor disability based on ICF-CY.
In particular, the SCADI dataset is the only one that has been used by ML researchers for self-care problems classification based on ICF-CY to this date.
The 'Class' field refers to the presence of the self-care problems of the children with physical and motor disabilities.The classes are determined by occupational therapists.
The names and social security numbers of the children were recently removed from the dataset.

Two files have been 'processed', SCADI.arff for using in WEKA and SCADI.CSV for using in MATLAB and similar tools.

Attribute Information:

1: gender: gender (1 = male; 0 = female)
2: age: age in years
3-205: self-care activities based on ICF-CY (1 = The case has this feature; 0 = otherwise)
206: Classes (class1 = Caring for body parts problem; class2 = Toileting problem; class3 = Dressing problem; class4 = Washing oneself and Caring for body parts and Dressing problem; class5 = Washing oneself, Caring for body parts, Toileting, and Dressing problem; class6 = Eating, Drinking, Washing oneself, Caring for body parts, toileting,Dressing, Looking after oneâ€™s health and Looking after oneâ€™s safety problem; class7 = No Problem;)

Relevant Papers:

Zarchi, M. S., SMM Fatemi Bushehri, and M. Dehghanizadeh. 'SCADI: A standard dataset for self-care problems classification of children with physical and motor disability.' International Journal of Medical Informatics (2018).

Bushehri, SMM Fatemi, and Mohsen Sardari Zarchi. "An expert model for self-care problems classification using probabilistic neural network and feature selection approach." Applied Soft Computing 82 (2019): 105545.

```r
w=read.csv("SCADI.csv")
for (i in (1:ncol(w))[-2]) w[,i]=factor(w[,i])
ss=sapply(w, function(x)length(unique(x)))
id=which(ss==1)
length(id) #63
v=w[,-id]

library(kknn)
a=kknn(Classes~.,train=v,test=v)
> table(v$Classes,a$fitted.values)

        class1 class2 class3 class4 class5 class6 class7
 class1      1      0      0      0      0      0      1
 class2      0      5      0      0      0      0      2
 class3      0      0      1      0      0      0      0
 class4      0      1      0     11      0      0      0
 class5      0      0      0      1      0      2      0
 class6      0      0      0      0      0     29      0
 class7      0      0      0      0      0      0     16
> mean(v$Classes!=a$fitted.values)
[1] 0.1
```

```r
set.seed(1010)
cv.kn=cv.kknn(Classes~.,v[,-id],kcv=10)
mean(cv.kn[[1]][,1]!=cv.kn[[1]][,2])

library(randomForest)
set.seed(1010)
(rf=randomForest(Classes~.,v))#OOB交叉验证误判率
sum(v$Classes!=predict(rf,v))#训练集误判率

Call:
 randomForest(formula = Classes ~ ., data = v)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 11

        OOB estimate of  error rate: 15.71%
Confusion matrix:
       class1 class2 class3 class4 class5 class6 class7 class.error
class1      0      0      0      1      0      0      1  1.00000000
class2      0      6      0      0      0      0      1  0.14285714
class3      0      0      0      0      0      0      1  1.00000000
class4      1      1      0     10      0      0      0  0.16666667
class5      0      0      0      1      0      2      0  1.00000000
class6      0      0      0      1      0     28      0  0.03448276
class7      0      0      0      1      0      0     15  0.06250000
> sum(v$Classes!=predict(rf,v))
[1] 0
```

8.3 K最邻近回归

```
w=read.csv("commun123.csv")
Z=10;n=nrow(w);set.seed(1010)
I=sample(rep(1:Z,ceiling(n/Z)))[1:n]
pred=rep(0,n)
for (i in 1:Z) {
  pred[I==i]=
  kknn(ViolentCrimesPerPop~.,w[I!=i,],w[I==i,])$fit
}
sum((w[,123]-pred)^2)/sum((w[,123]-mean(w[,123]))^2)
```

```
> sum((w[,123]-pred)^2)/sum((w[,123]-mean(w[,123]))^2)
[1] 0.4045627
```

九、有监督学习模型比较案例

9.1 多分类问题例子

案例

```
library(mlbench)
library(adabag) ;library(ipred); library(kknn)
library(e1071); library(randomForest)
data(DNA)#3186 181 names(w) [181]
w=DNA
Z=10;D=181;n=nrow(w)
mm=Fold(Z, w, D, 1010)
kn=w[,181]->prf->NB->bag->svmc->ada#在数据中增加一列准备放预测的值
set.seed(1010)
for(i in 1:Z) {
  kn[mm[[i]]]=kknn(Class~.,w[-mm[[i]],],w[mm[[i]],])$fit
  prf[mm[[i]]]=randomForest(Class~.,w[-mm[[i]],])%>%
    predict(w[mm[[i]],])
  svmc[mm[[i]]]=svm(Class~.,w[-mm[[i]],])%>%
    predict(w[mm[[i]],])
  bag[mm[[i]]]=ipred:: bagging(Class~.,w[-mm[[i]],])%>%
    predict(w[mm[[i]],])
  a=boosting(Class~.,w[-mm[[i]],])
  ada[mm[[1]]]=predict(a,w[mm[[i]],])$class
  NB[mm[[i]]]=naiveBayes(Class~.,w[-mm[[i]],])%>%
    predict(w[mm[[i]],])
}
Pred=data.frame(bag,ada, prf, svmc, NB, kn)
error=sapply(Pred, function(x)sum(w$Class!=x)/n)
```

```
> error
        bag         ada         prf        svmc          NB          kn
0.057438795 0.006277464 0.041117389 0.043942247 0.059008161 0.228185813
```

– G. G. Towell, 1991; "Symbolic Knowledge and Neural Networks: Insertion, Refinement, and Extraction", PhD Thesis, University of Wisconsin - Madison.

– G. G. Towell and J. W. Shavlik, 1992; "Interpretation of Artificial Neural Networks: Mapping Knowledge-based Neural Networks into Rules", In Advances in Neural Information Processing Systems, volume 4, Morgan Kaufmann.

DNA *Primate splice-junction gene sequences (DNA)*

Description

It consists of 3,186 data points (splice junctions). The data points are described by 180 indicator binary variables and the problem is to recognize the 3 classes (ei, ie, neither), i.e., the boundaries between exons (the parts of the DNA sequence retained after splicing) and introns (the parts of the DNA sequence that are spliced out).
The StaLog dna dataset is a processed version of the Irvine database described below. The main difference is that the symbolic variables representing the nucleotides (only A,G,T,C) were replaced by 3 binary indicator variables. Thus the original 60 symbolic attributes were changed into 180 binary attributes. The names of the examples were removed. The examples with ambiguities were removed (there was very few of them, 4). The StatLog version of this dataset was produced by Ross King at Strathclyde University. For original details see the Irvine database documentation.

The nucleotides A,C,G,T were given indicator values as follows:

A -> 1 0 0
C -> 0 1 0
G -> 0 0 1
T -> 0 0 0

Hint. Much better performance is generally observed if attributes closest to the junction are used. In the StatLog version, this means using attributes A61 to A120 only.

Usage

```
data(DNA)
```

Format

A data frame with 3,186 observations on 180 variables, all nominal and a target class.

Source

- Source:
  - all examples taken from Genbank 64.1 (ftp site: genbank.bio.net)
  - categories "ei" and "ie" include every "split-gene" for primates in Genbank 64.1
  - non-splice examples taken from sequences known not to include a splicing site

- Donor: G. Towell, M. Noordewier, and J. Shavlik, towell,shavlik@cs.wisc.edu, noordewi@cs.rutgers.edu

These data have been taken from:

- ftp.stams.strath.ac.uk/pub/Statlog

and were converted to R format by Evgenia Dimitriadou.

References

machine learning:
– M. O. Noordewier and G. G. Towell and J. W. Shavlik, 1991; "Training Knowledge-Based Neural Networks to Recognize Genes in DNA Sequences". Advances in Neural Information Processing Systems, volume 3, Morgan Kaufmann.

– G. G. Towell and J. W. Shavlik and M. W. Craven, 1991; "Constructive Induction in Knowledge-Based Neural Networks", In Proceedings of the Eighth International Machine Learning Workshop, Morgan Kaufmann.

9.2 二分类问题

```
w=read.csv("Sports.csv")[,-c(17, 40, 49, 59)]
library(adabag);library(ipred);library(kknn)
library(e1071);library (randomForest);
Z=10;D=1;n=nrow(w)
mm=Fold(Z,w,D,1010)
kn=w[,1]->prf->NB->bag->svmc->ada
set.seed(1010)
for(i in 1:Z){
    kn[mm[[i]]]=kknn(Label~.,w[-mm[[i]],],w[mm[[i]],])$fit
    prf[mm[[i]]]=randomForest(Label~.,w[-mm[[i]],]) %>%
      predict(w[mm[[i]],])
    svmc[mm[[i]]]=svm(Label~.,w[-mm[[i]],])%>%
      predict(w[mm[[i]],])
    bag[mm[[i]]]=ipred::bagging(Label~.,w[-mm[[i]],])%>%
      predict(w[mm[[i]],])
    a=boosting(Label~.,w[-mm[[i]],])
    ada[mm[[i]]]=predict(a,w[mm[[i]],])$class
    NB[mm[[i]]]=naiveBayes(Label~.,w[-mm[[i]],])%>%
      predict(w[mm[[i]],])
}
Pred=data.frame(bag, ada, prf, svmc, NB, kn)
error=sapply(Pred, function(x) sum(w[,1]!=x)/n)
```

```
> error
   bag    ada    prf   svmc     NB     kn
0.172 0.174 0.175 0.162 0.203 0.183
```

案例

UCI Machine Learning Repository
Center for Machine Learning and Intelligent Systems

Mushroom Data Set

Download Data Folder, Data Set Description

Abstract: From Audobon Society Field Guide; mushrooms described in terms of physical characteristics; classification: poisonous or edible

Data Set Characteristics:	Multivariate	**Number of Instances:**	8124	**Area:**	Life
Attribute Characteristics:	Categorical	**Number of Attributes:**	22	**Date Donated**	1987-04-27
Associated Tasks:	Classification	**Missing Values?**	Yes	**Number of Web Hits:**	504344

http://archive.ics.uci.edu/ml/datasets/Mushroom

Source:

Origin:

Mushroom records drawn from The Audubon Society Field Guide to North American Mushrooms (1981). G. H. Lincoff (Pres.), New York: Alfred A. Knopf

Donor:

Jeff Schlimmer (Jeffrey.Schlimmer "@" a.gp.cs.cmu.edu)

Data Set Information:

This data set includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family (pp. 500-525). Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one. The Guide clearly states that there is no simple rule for determining the edibility of a mushroom; no rule like ``leaflets three, let it be'' for Poisonous Oak and Ivy.

Attribute Information:

1. cap-shape: bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s
2. cap-surface: fibrous=f,grooves=g,scaly=y,smooth=s
3. cap-color: brown=n,buff=b,cinnamon=c,gray=g,green=r, pink=p,purple=u,red=e,white=w,yellow=y
4. bruises?: bruises=t,no=f
5. odor: almond=a,anise=l,creosote=c,fishy=y,foul=f, musty=m,none=n,pungent=p,spicy=s
6. gill-attachment: attached=a,descending=d,free=f,notched=n
7. gill-spacing: close=c,crowded=w,distant=d
8. gill-size: broad=b,narrow=n
9. gill-color: black=k,brown=n,buff=b,chocolate=h,gray=g, green=r,orange=o,pink=p,purple=u,red=e, white=w,yellow=y
10. stalk-shape: enlarging=e,tapering=t
11. stalk-root: bulbous=b,club=c,cup=u,equal=e, rhizomorphs=z,rooted=r,missing=?
12. stalk-surface-above-ring: fibrous=f,scaly=y,silky=k,smooth=s
13. stalk-surface-below-ring: fibrous=f,scaly=y,silky=k,smooth=s
14. stalk-color-above-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y
15. stalk-color-below-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y
16. veil-type: partial=p,universal=u
17. veil-color: brown=n,orange=o,white=w,yellow=y
18. ring-number: none=n,one=o,two=t
19. ring-type: cobwebby=c,evanescent=e,flaring=f,large=l, none=n,pendant=p,sheathing=s,zone=z
20. spore-print-color: black=k,brown=n,buff=b,chocolate=h,green=r, orange=o,purple=u,white=w,yellow=y
21. population: abundant=a,clustered=c,numerous=n, scattered=s,several=v,solitary=y
22. habitat: grasses=g,leaves=l,meadows=m,paths=p, urban=u,waste=w,woods=d

Relevant Papers:

Schlimmer,J.S. (1987). Concept Acquisition Through Representational Adjustment (Technical Report 87-19). Doctoral disseration, Department of Information and Computer Science, University of California, Irvine.
[Web Link]

Iba,W., Wogulis,J., & Langley,P. (1988). Trading off Simplicity and Coverage in Incremental Concept Learning. In Proceedings of the 5th International Conference on Machine Learning, 73-79. Ann Arbor, Michigan: Morgan Kaufmann.
[Web Link]

Duch W, Adamczak R, Grabczewski K (1996) Extraction of logical rules from training data using backpropagation networks, in: Proc. of the The 1st Online Workshop on Soft Computing, 19-30.Aug.1996, pp. 25-30, [Web Link]
[Web Link]

Duch W, Adamczak R, Grabczewski K, Ishikawa M, Ueda H, Extraction of crisp logical rules using constrained backpropagation networks - comparison of two new approaches, in: Proc. of the European Symposium on Artificial Neural Networks (ESANN'97), Bruge, Belgium 16-18.4.1997.
[Web Link]

```r
w=read.csv("mushroom.csv")
Z=10; D=1;n=nrow(w)
mm=Fold(Z,w,D,1010)
w[,1]->prf->NB->bag->ada
set.seed(1010)
for(i in 1:Z){
  prf[mm[[i]]]=randomForest(type~.,w[-mm[[i]],])%>%
     predict(w[mm[[i]],])
  bag[mm[[i]]]=ipred::bagging(type~.,w[-mm[[i]],])%>%
     predict(w[mm[[i]],])
 a=boosting(type~.,w[-mm[[i]],])
 ada[mm[[i]]]=predict(a,w[mm[[i]],])$class
 NB[mm[[i]]]=naiveBayes(type~.,w[-mm[[i]],])%>%
    predict(w[mm[[i]],])
}
Pred=data.frame(bag, ada, prf, NB)
error=sapply(Pred, function(x)sum(w$Class!=x)/n)
```

```
> error
bag ada prf  NB
  0   0   0   0
```

9.3 回归问题

```
library(mlbench);data(BostonHousing2)
w=BostonHousing2[,-c(1:5)]
w$chas=factor(w$chas)

w=read.csv("BostonHousing2.csv")[,-c(1:5)]
s.w=lapply(w,function(x){(x-min(x))/(max(x)-min(x))})%>%as.data.frame()
nm<-names(s.w)
f<-as.formula(paste("cmedv ~", paste(nm[!nm %in% "cmedv"], collapse=" + ")))
w$chas=factor(w$chas)
library(dplyr) ;library(ipred);library(kknn)
library(e1071);library(randomForest);library(neuralnet)


Z=10;n=nrow(w);set.seed(1010)
I=sample(rep(1:Z,ceiling(n/Z)))[1:n]
ff=formula("cmedv~.")
rep(0,n)->kn->prf->bag->svmc->Lm->net

set.seed(1010)
for(i in 1:Z){
  kn[I==i]=kknn(ff,w[I!=i,],w[I==i,])$fit
  prf[I==i]=randomForest(ff,w[I!=i,])%>%
    predict(w[I==i,])
  svmc[I==i]=svm(ff,w[I!=i,])%>%
    predict(w[I==i,])
  bag[I==i]=ipred::bagging(ff, w[I!=i,])%>%
    predict(w[I==i,])
  Lm[I==i]=lm(ff,w[I!=i,])%>%
    predict(w[I==i,])
  nn=neuralnet(f,data=s.w[I!=i,], hidden=c(5,3),linear.output=T)
  net[I==i]=compute(nn,s.w[I==i,-1])$net.result
}
net=net*(max(w$cmedv)-min(w$cmedv))+min(w$cmedv)

Pred=data.frame(kn, prf, bag, svmc, Lm, net)
NMSE=sapply(Pred, function(x) sum((w[,1]-x)^2)/sum((w[,1]-mean(w[,1]))^2))

barplot(NMSE, names.arg=c("KNN", "Rabdom Forest", "Bagging", "SVM", "Linear Model","Neural Network"),
        horiz=TRUE, col=4, las=2, main="10-Fold CV NMSE of Boston Housing Regression")
```

```
> NMSE
       kn       prf       bag      svmc        Lm       net
0.1914457 0.1152912 0.1802888 0.1487280 0.2850574 0.1477548
```