北京郵電大學
BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS

Data Analytics with R
语言数据分析

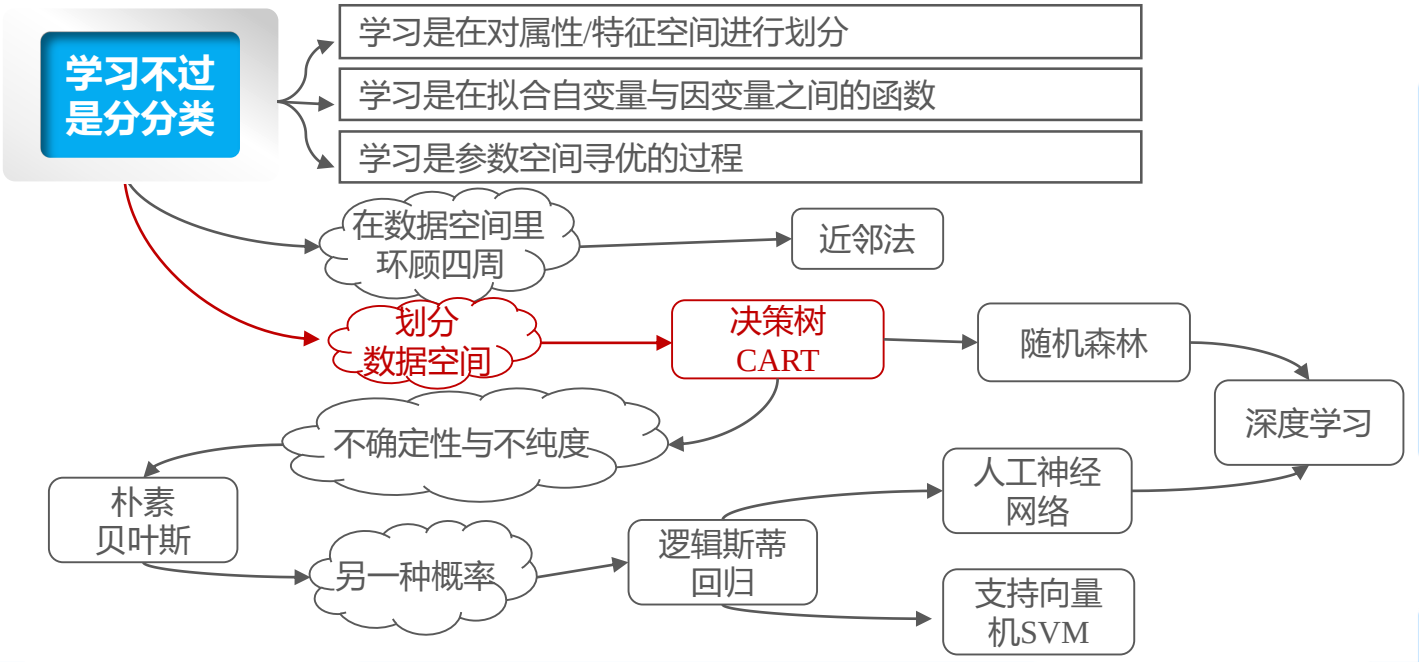
既是世间法、自当有分别

艾新波 / 2018 • 北京

课程体系

- R语言数据分析
  - 上部：论道 
    - 第1章 气象万千、数以等观
    - 第2章 所谓学习、归类而已
    - 第3章 格言联璧话学习
    - 第4章 源于数学、归于工程
  - 中部：执具 
    - 第5章 工欲善其事必先利其器
    - 第6章 基础编程
    - 第7章 数据对象
  -
    - 第8章 人人都爱tidyverse
    - 第9章 最美不过数据框
  - 下部 博术 
    - 第10章 观数以形
    - 第11章 相随相伴、谓之关联
    - 第12章 既是世间法、自当有分别
    - 第13章 方以类聚、物以群分
    - 第14章 庐山烟雨浙江潮

算法模型


学习不过是分分类
学习是在对属性/特征空间进行划分
学习是在拟合自变量与因变量之间的函数
学习是参数空间寻优的过程
在数据空间里环顾四周
近邻法
划分数据空间
决策树CART
随机森林
深度学习
不确定性与不纯度
朴素贝叶斯
另一种概率
逻辑斯蒂回归
人工神经网络
支持向量机SVM

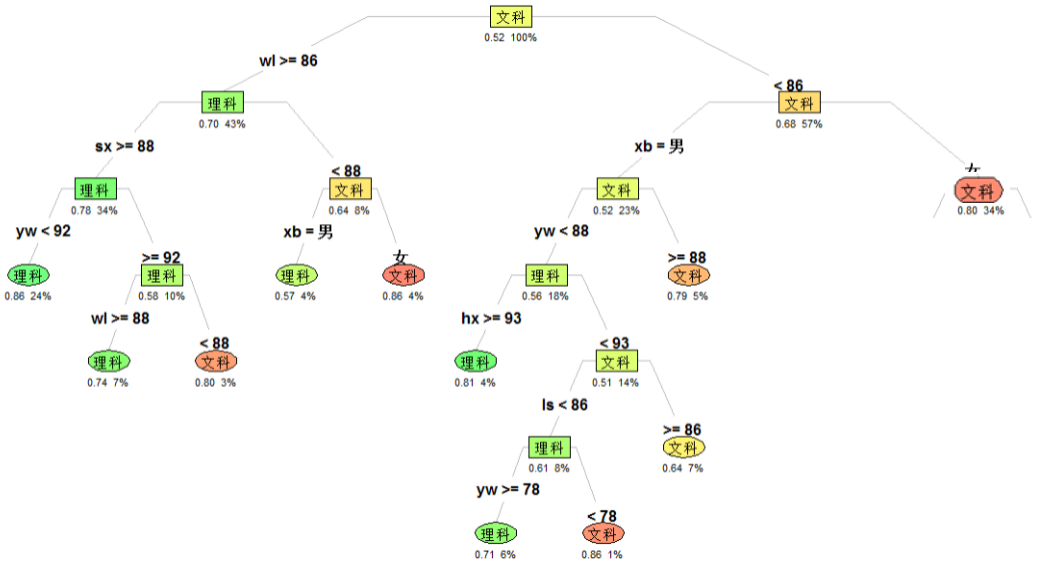
过拟合举例：瓶子的二值分类








决策树剪枝


文科
0.52 100%
wl >= 86
< 86
理科
0.70 43%
文科
0.68 57%
sx >= 88
xb = 男
< 88
文科
0.64 8%
理科
0.78 34%
文科
0.52 23%
文科
0.80 34%
yw < 92
>= 92
xb = 男
yw < 88
>= 88
理科
0.86 24%
理科
0.58 10%
理科
0.57 4%
文科
0.86 4%
理科
0.56 18%
文科
0.79 5%
wl >= 88
< 88
hx >= 93
< 93
理科
0.74 7%
文科
0.80 3%
理科
0.81 4%
文科
0.51 14%
ls < 86
>= 86
理科
0.61 8%
文科
0.64 7%
yw >= 78
< 78
理科
0.71 6%
文科
0.86 1%

代价-复杂度剪枝

定义损失函数：

$$C_\alpha(T) = C(T) + \alpha|T|$$

其中$C(T)$为对训练数据的预测误差，$|T|$为叶节点个数

参数$\alpha \geq 0$用于权衡训练数据拟合程度与模型的复杂度：将α从小增到大，不同的α对应着不同的最优子树。

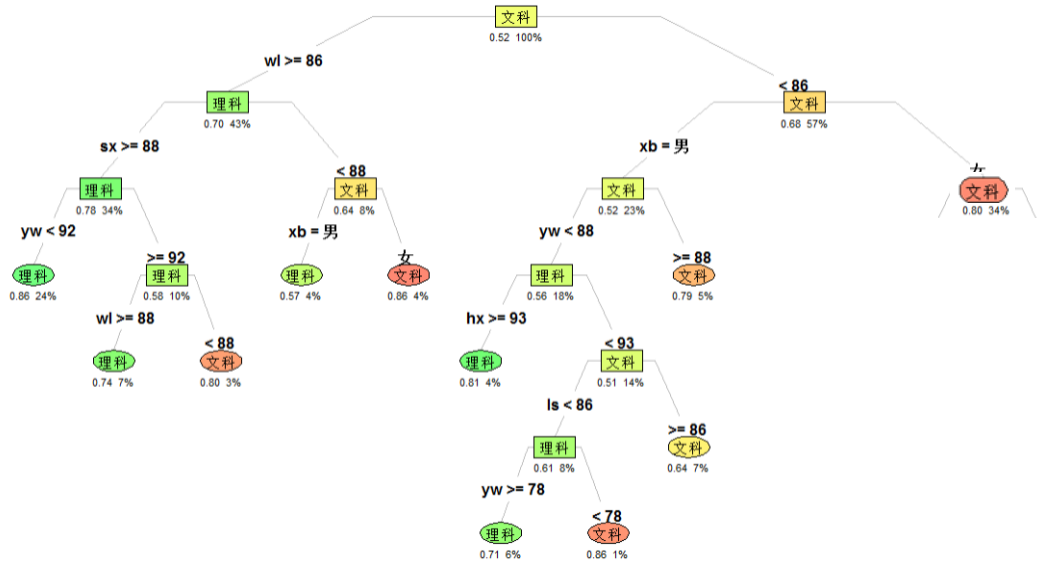
当α小的时候，最优子树偏大。特别地，$\alpha = 0$时，整体树是最优的；当α大的时候，最优子树偏小。增大到某个特定值时，根节点组成的单节点树最优

更准确的讲：α落入不同区间时，对应着不同的最优子树。

$0 = \alpha_0 < \alpha_1 < \alpha_2 < \cdots < \alpha_n < \infty$，产生一序列的区间：$[\alpha_i, \alpha_{i+1}), i = 1, 2, \cdots, n$，

当$\alpha \in [\alpha_i, \alpha_{i+1}), i = 1, 2, \cdots, n$时，对应最优子树序列$\{T_0, T_1, \cdots, T_n\}$

代价-复杂度剪枝



$\alpha = \alpha_0$ $\alpha = \alpha_1$ $\alpha = \alpha_2$ $\alpha = \alpha_3$

代价-复杂度剪枝

对T_0的任意内部节点t：

以t为单节点树的损失函数：$C_\alpha(t) = C(t) + \alpha$

以t为根节点的子树T_t的损失函数：$C_\alpha(T_t) = C(T_t) + \alpha|T_t|$

当$\alpha = 0$或α充分小时，有不等式：$C_\alpha(T_t) < C_\alpha(t)$

当α增大时，在某一α有：$C_\alpha(T_t) = C_\alpha(t)$。

当α进一步增大时，不等式将反向

当$\alpha = \frac{C(t)-C(T_t)}{|T_t|-1}$，$T_t$与$t$具有相同的损失函数，而$t$的节点少，因而$t$比$T_t$更可取，应对$T_t$进行剪枝

代价-复杂度剪枝

对T_0的任意内部节点t，计算：

$$g(t) = \frac{C(t) - C(T_t)}{|T_t| - 1}$$

在T_0中剪去$g(t)$最小的T_t，将得到的子树作为T_1，同时将最小的$g(t)$作为α_1，T_1为区间$[\alpha_1, \alpha_2)$的最优子树

如此剪枝下去，直至得到根节点，得到与$0 = \alpha_0 < \alpha_1 < \alpha_2 < \cdots < \alpha_n$相对应的序列子树$\{T_0, T_1, \cdots, T_n\}$

最后，通过交叉验证的方法，从这个子树序列中找到最好的那个子树，完成剪枝

（机器学习的模型，大多是通过实验的方法、而非理论的方法进行择优^-^）

代价-复杂度剪枝

输入：完全生长的决策树T_0

输出：最优决策树T_α

(1) 设$k=0$，$T=T_0$

(2) $\alpha=\infty$

(3) 自下而上对各内部节点t计算$C(T_t)$，$|T_t|$以及：

$$g(t)=\frac{C(t)-C(T_t)}{|T_t|-1}$$

$$\alpha=\min(\alpha,\ g(t))$$

(4) 对$g(t)=\alpha$自内部节点t进行剪枝，并对叶节点t以多数表决法决定其类，得到树T

(5) 设$k=k+1,\alpha_k=\alpha,T_k=T$

(6) 如果T不是由根节点及两个叶节点组成的树，则回到步骤（2）；否则令$n=k$

(7) 采用交叉验证法在子树序列$T_0,T_1,\cdots,T_n$中选取最优子树T_α

R语言实现

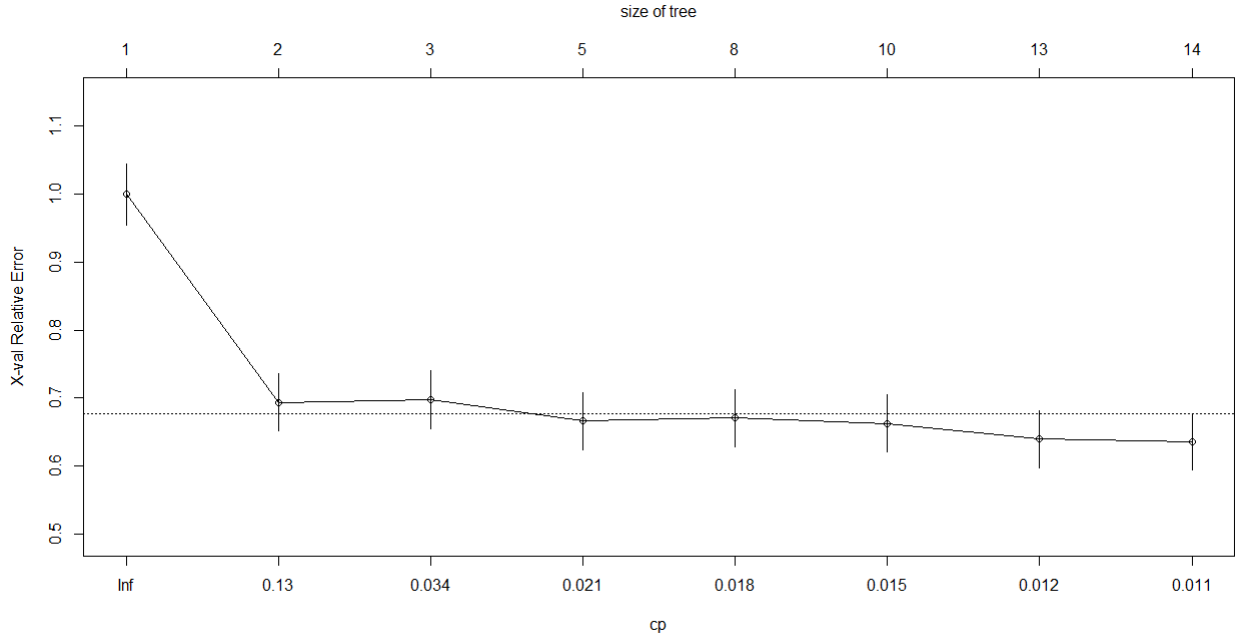
```
printcp(imodel, digits = 2)
#> Classification tree:
#>   rpart(formula = wlfk ~ ., data = cjb[train_set_idx, ])
#>
#> Variables actually used in tree construction:
#>   [1] hx ls sw sx wl wy xb
#>
#> Root node error: 266/542 = 0.49
#>
#> n= 542
```

R语言实现

```
#>       CP nsplit rel error xerror  xstd
#> 1 0.349      0      1.00   1.00 0.045
#> 2 0.050      1      0.65   0.69 0.042
#> 3 0.023      2      0.60   0.70 0.042
#> 4 0.019      4      0.55   0.67 0.042
#> 7 0.012     12      0.42   0.66 0.042
#> 8 0.010     13      0.41   0.65 0.042
```

```
#查看不同cp值情况下交叉验证的结果
plotcp(imodel)
```

R语言实现


size of tree
1
2
3
5
8
10
13
14
X-val Relative Error
0.5
0.6
0.7
0.8
0.9
1.0
1.1
Inf
0.13
0.034
0.021
0.018
0.015
0.012
0.011
cp

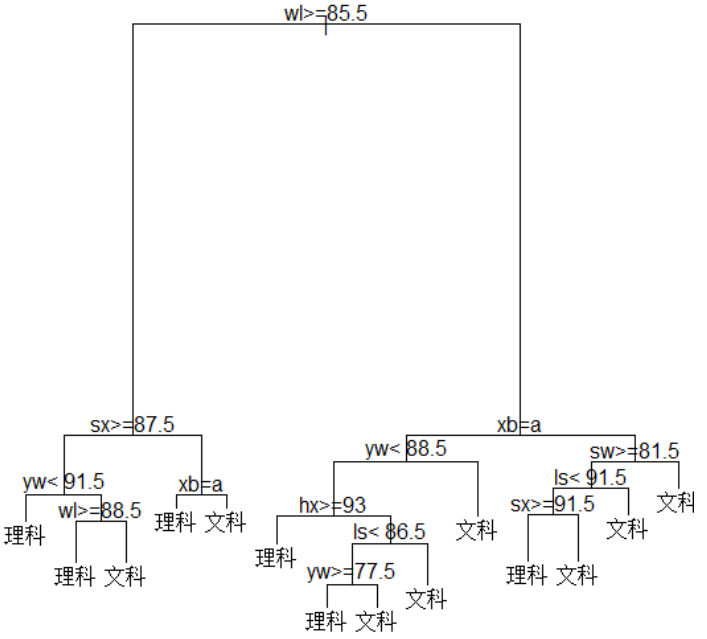
R语言实现

```
imodel$cptable
#>           CP nsplit rel error    xerror       xstd
#> 1 0.34883721      0 1.0000000 1.0000000 0.04502822
#> 6 0.01291990      9 0.4612403 0.6434109 0.04157683
#> 7 0.01162791     12 0.4224806 0.6356589 0.04143566
#> 8 0.01000000     13 0.4108527 0.6356589 0.04143566
#剪枝的一般方法
opt <- which.min(imodel$cptable[, "xerror"])
cp <- imodel$cptable[opt, "CP"]
imodel_pruned <- prune(imodel, cp = cp)
```

R语言实现

#绘制决策树的基本方法

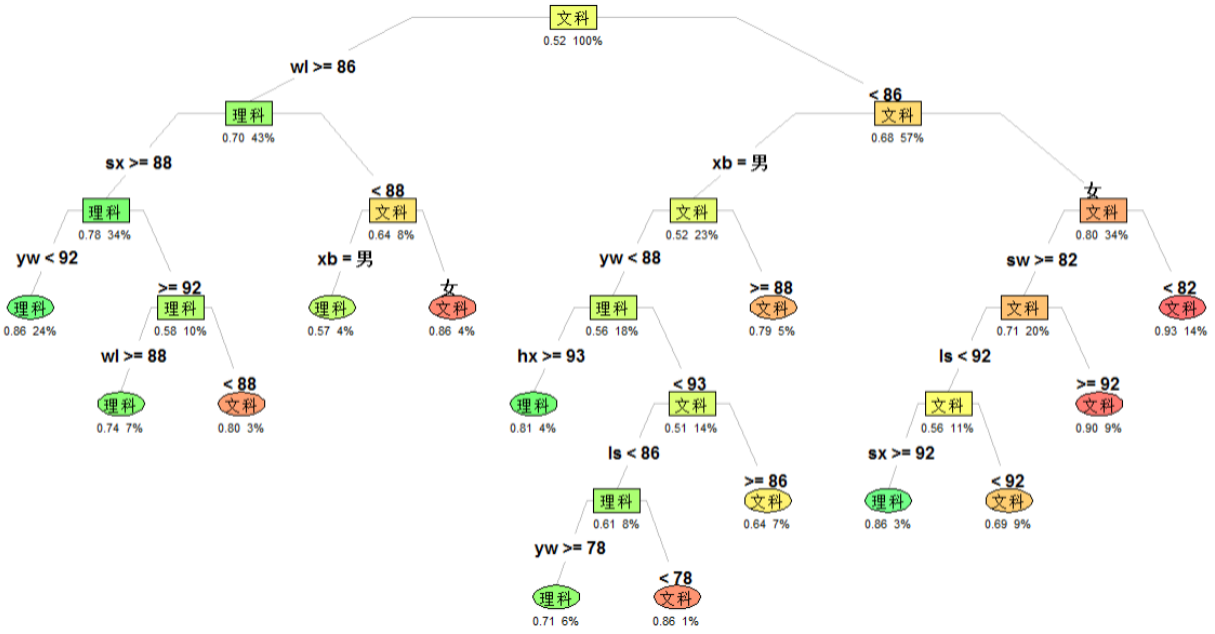
```
plot(imodel)
text(imodel)
```



R语言实现

```
rpart.plot(imodel_pruned,
    type=4, fallen=F,
    branch=0.5, round=0,
    leaf.round=2, clip.right.labs=T,
    cex = 0.85, under.cex=0.75,
    box.palette="GnYlRd",
    branch.col="gray",
    branch.lwd=2,
    extra=108,
    under=T,split.cex=0.8)
```

R语言实现


文科
0.52 100%
wl >= 86
< 86
理科
0.70 43%
文科
0.68 57%
sx >= 88
< 88
xb = 男
女
理科
0.78 34%
文科
0.64 8%
文科
0.52 23%
文科
0.80 34%
yw < 92
>= 92
xb = 男
女
yw < 88
>= 88
sw >= 82
< 82
理科
0.86 24%
理科
0.58 10%
理科
0.57 4%
文科
0.86 4%
理科
0.56 18%
文科
0.79 5%
文科
0.71 20%
文科
0.93 14%
wl >= 88
< 88
hx >= 93
< 93
ls < 92
>= 92
理科
0.74 7%
文科
0.80 3%
理科
0.81 4%
文科
0.51 14%
文科
0.56 11%
文科
0.90 9%
ls < 86
>= 86
sx >= 92
< 92
理科
0.61 8%
文科
0.64 7%
理科
0.86 3%
文科
0.69 9%
yw >= 78
< 78
理科
0.71 6%
文科
0.86 1%

R语言实现

```
#规则导出
library(rattle)
rules <- asRules(imodel_pruned, compact=TRUE)
#> R  7 [22%,0.90] sx< 85.5 xb=女
#> R 11 [11%,0.85] sx>=85.5 wl< 86.5 ls>=92.5
#> R 51 [ 4%,0.79] sx< 85.5 xb=男 hx>=83 wy>=81.5 sx< 76.5
#> R 13 [ 5%,0.75] sx< 85.5 xb=男 hx< 83
#> R 21 [ 3%,0.75] sx>=85.5 wl< 86.5 ls< 92.5 sw< 80.5
#> R 19 [ 3%,0.75] sx>=85.5 wl>=86.5 ls>=95.5 sw< 92.5
#> R 24 [ 5%,0.28] sx< 85.5 xb=男 hx>=83 wy< 81.5
```

R语言实现

```
sp <- Sys.time() #记录开始时间
cat("\n[Start at:", as.character(sp))
library(rpart)
for(i in 1:length(kfolds)) {
  curr_fold <- kfolds[[i]] #当前这一折
  train_set <- cjb[-curr_fold, ] #训练集
  test_set <- cjb[curr_fold, ] #测试集
  imodel_kfold <- rpart(wlfk~., train_set) #模型训练
  opt <- which.min(imodel_kfold$cptable[, "xerror"])
  cp <- imodel_kfold$cptable[opt, "CP"]
  imodel_kfold <- prune(imodel_kfold, cp = cp)
```

R语言实现

```
  predicted_train <- predict(imodel_kfold,
                             train_set, type = "class")
  imetrics("rpart", "Train", predicted_train, train_set$wlfk)
  predicted_test <- predict(imodel_kfold,
                            test_set, type = "class")
  imetrics("rpart", "Test", predicted_test, test_set$wlfk)
}
ep <- Sys.time()
cat("\tFinised at:", as.character(ep), "]\n")
cat("[Time Ellapsed:\t",
    difftime(ep, sp, units = "secs"), " seconds]\n")
```

R语言实现

```
#> 21   rpart Train 0.7859195   0.2140805
#> 22   rpart  Test 0.6923077   0.3076923
#> 23   rpart Train 0.8160920   0.1839080
#> 24   rpart  Test 0.7435897   0.2564103
#> 25   rpart Train 0.7600575   0.2399425
#> 26   rpart  Test 0.7692308   0.2307692
.....
#> 37   rpart Train 0.8106169   0.1893831
#> 38   rpart  Test 0.6623377   0.3376623
#> 39   rpart Train 0.8177905   0.1822095
#> 40   rpart  Test 0.7012987   0.2987013
```

谢谢聆听

Thank you

教师个人联系方式

艾新波

手机：13641159546

QQ： 23127789

微信：13641159546

E-mail: 13641159546@126.com

axb@bupt.edu.cn

地址：北京邮电大学科研楼917室

课程网址：https://github.com/byaxb/RDataAnalytics