多元统计第三次作业

蒋文馨

2020/6/2

目录

1 Best Split for Cleveland

9.2 The discussion of the way to choose the best split for a classification tree in Section 9.2 used the entropy function as the impurity measure. Use the Gini index as an impurity measure on the Cleveland heart-disease data and determine the best split for the age variable (see Table 9.2); draw the graphs of i(τ_L) and i(τ_R) for the age variable and the goodness of split (see Figure 9.3). Determine the best split for all the variables in the data set (see Table 9.3).

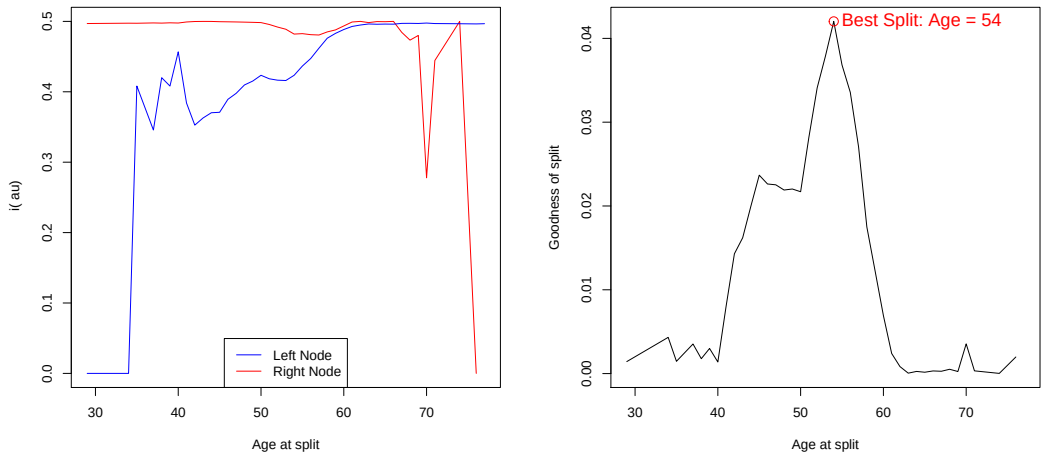
```
library(readr)
cleveland = read_table2("cleveland_.txt")#这份文件把原文件所有的"都删了
chd = cleveland[, -15]#删去无关信息
chd$diag[chd$diag == 'buff'] = 0
chd$diag[chd$diag == 'sick'] = 1

#gini index
p = sum(chd$diag == chd$diag[1]) / length(chd$diag)
i.tau = 2 * p * (1 - p)

#split using age
chd2 = cbind(chd$age, chd$diag)#取出待分割特征和分类标签
split.f = sort(unique(chd2[, 1]))#取出分割点
t = length(split.f)

n = matrix(0, 2, 2)
delta = tau.r = tau.l = numeric(t)
```

```
for (i in 1:t) {
  c.left = matrix(chd2[chd2[, 1] <= split.f[i], ], ncol = 2)#分到左子树
  c.right = matrix(chd2[chd2[, 1] > split.f[i], ], ncol = 2)#分到右子树
  n = matrix(c(
    sum(c.left[, 2] == 0),
    sum(c.right[, 2] == 0),
    sum(c.left[, 2] == 1),
    sum(c.right[, 2] == 1)
  ), ncol = 2)
  n.row = rowSums(n); n.col = colSums(n); n.sum = sum(n)
  p = n[1, 1] / n.row[1]
  tau.l[i] = 2 * p * (1 - p)
  p = n[2, 1] / n.row[2]
  tau.r[i] = 2 * p * (1 - p)
  delta[i] = i.tau - n.row[1] / n.sum * tau.l[i] - n.row[2] / n.sum * tau.r[i]
}
```



(a) i(tauL) and i(tauR)

(b) Goodness of Split

图 1: Best Split for Age using Gini Index

```
#split using all variables
best.split = data.frame(row.names = c('delta.i', 'split'))
for (j in 1:13) {
  chd2 = cbind(chd[, j], chd$diag)#取出待分割特征和分类标签
  split.f = sort(unique(chd2[, 1]))#取出分割点
  t = length(split.f)
```

```
  n = matrix(0, 2, 2)
  delta = tau.r = tau.l = numeric(t)
  for (i in 1:t) {
    #连续变量
    if (typeof(split.f[1]) == 'double') {
      c.left = as.matrix(chd2[chd2[, 1] <= split.f[i],], ncol = 2)#分到左子树
      c.right = as.matrix(chd2[chd2[, 1] > split.f[i],], ncol = 2)#分到右子树
    }
    else{
      #分类变量
      c.left = as.matrix(chd2[chd2[, 1] == split.f[i],], ncol = 2)#分到左子树
      c.right = as.matrix(chd2[chd2[, 1] != split.f[i],], ncol = 2)#分到右子树
    }
    n = matrix(c(
      sum(c.left[, 2] == 0),
      sum(c.right[, 2] == 0),
      sum(c.left[, 2] == 1),
      sum(c.right[, 2] == 1)
    ), ncol = 2)
    n.row = rowSums(n)
    n.col = colSums(n)
    n.sum = sum(n)
    p = n[1, 1] / n.row[1]
    tau.l[i] = 2 * p * (1 - p)
    p = n[2, 1] / n.row[2]
    tau.r[i] = 2 * p * (1 - p)
    delta[i] = i.tau - n.row[1] / n.sum * tau.l[i] - n.row[2] / n.sum * tau.r[i]
  }
  max.delta = which.max(delta)
  best.split[1, j] = round(delta[max.delta], 5)
  best.split[2, j] = split.f[max.delta]
}
names(best.split) = names(chd)[1:13]
```

Best split for all the variables:

```
##              age  gender      cp trestbps      chol   fbs restecg   thatach
## delta.i  0.04203 0.04044 0.12675 9.99e-03   0.01099 1e-05 0.01436   0.09027
## split   54.00000    male  asympt 1.42e+02 271.00000   fal    norm 147.00000
##           exang oldpeak   slope      ca    thal
## delta.i 0.08975  0.0823 0.07401 0.11836 0.13954
## split      true  1.6000      up 0.00000    norm
```

注：类别变量 *chd$cp* 有 *4* 种取值，共有 *4(*利用一种类别分割*)+3(*利用两种类别分割*)=7* 种分割方法。上述代码无法计算后 *3* 种分割的 *delta i*。修改代码，单独计算后发现，其分割效果均不如单独使用 *asympt*。故最终分割方案如上表所示。

2 Resubstitution Error Rate

9.4 Consider the following two examples. Both examples start out with a root node with 800 subjects of which 400 have a given disease and the other 400 do not. The first example splits the root node as follows: the left node has 300 with the disease and 100 without, and the right node has 100 with the disease and 300 without. The second example splits the root node as follows: the left node has 200 with the disease and 400 without, and the right node has 200 with the disease and 0 without. Compute the resubstitution error rate for both examples and show they are equal. Which example do you view as more useful for the future growth of the tree?

```
# first example
n = 800; ld = 300; lh = 100; rd = 100; rh = 300
rt.1 = (min(ld, lh) + min(rd, rh)) / n
#second example
n = 800; ld = 200; lh = 400; rd = 200; rh = 0
rt.2 = (min(ld, lh) + min(rd, rh)) / n
# whether the resubstitution error rate for examples are equal
rt.1 == rt.2
```

```
## [1] TRUE
```

The second example is more useful for the future growth of the tree, for the right node of which is already a terminal node, which means that we only need to classify 600 subjects in left node.

3 Appropriate-size Classification Tree for Vehicle Data

9.8 Construct the appropriate-size classification tree for the vehicle data(see Section 8.7).

```
library(readr)
vehicle3 = read_table2("vehicle3_.txt")
vehicle3$class = factor(vehicle3$class)

library(rpart)
library(rpart.plot)
out = rpart(class ~ . - pam, vehicle3)
plotcp(out)
```

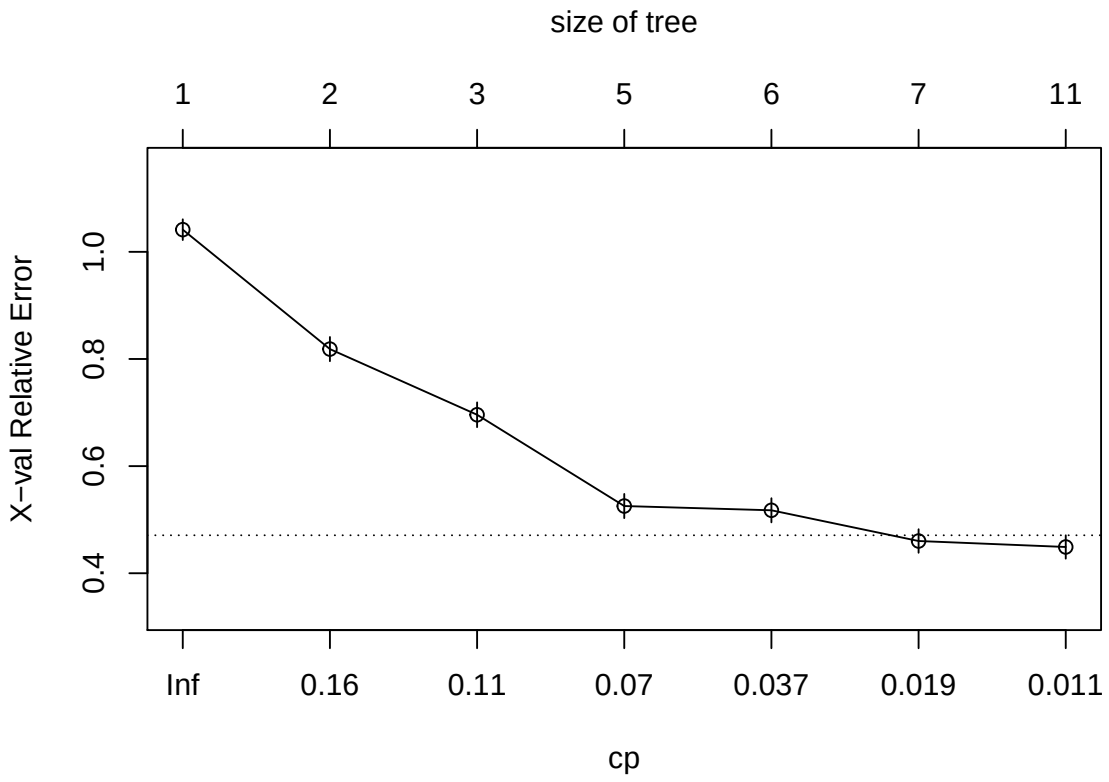


图 2: Relative Error of Vehicle Data

```
out$cptable
```

```
##           CP nsplit rel error    xerror       xstd
## 1 0.20541401      0 1.0000000 1.0414013 0.01939968
## 2 0.12101911      1 0.7945860 0.8184713 0.02261548
## 3 0.09554140      2 0.6735669 0.6958599 0.02314502
## 4 0.05095541      4 0.4824841 0.5254777 0.02259108
## 5 0.02707006      5 0.4315287 0.5175159 0.02252764
## 6 0.01273885      6 0.4044586 0.4601911 0.02196502
## 7 0.01000000     10 0.3535032 0.4490446 0.02183329
```

```
out.prune = prune(out, cp = out$cptable[7, 1])
rpart.plot(out.prune, type = 1, extra = 1)
```

注：由图 *2* 可知，应该取 *size=7*，而 *cptable* 只有 *7* 行，所以直接 *rpart.plot(out, type = 1, extra = 1)* 也是可以的。

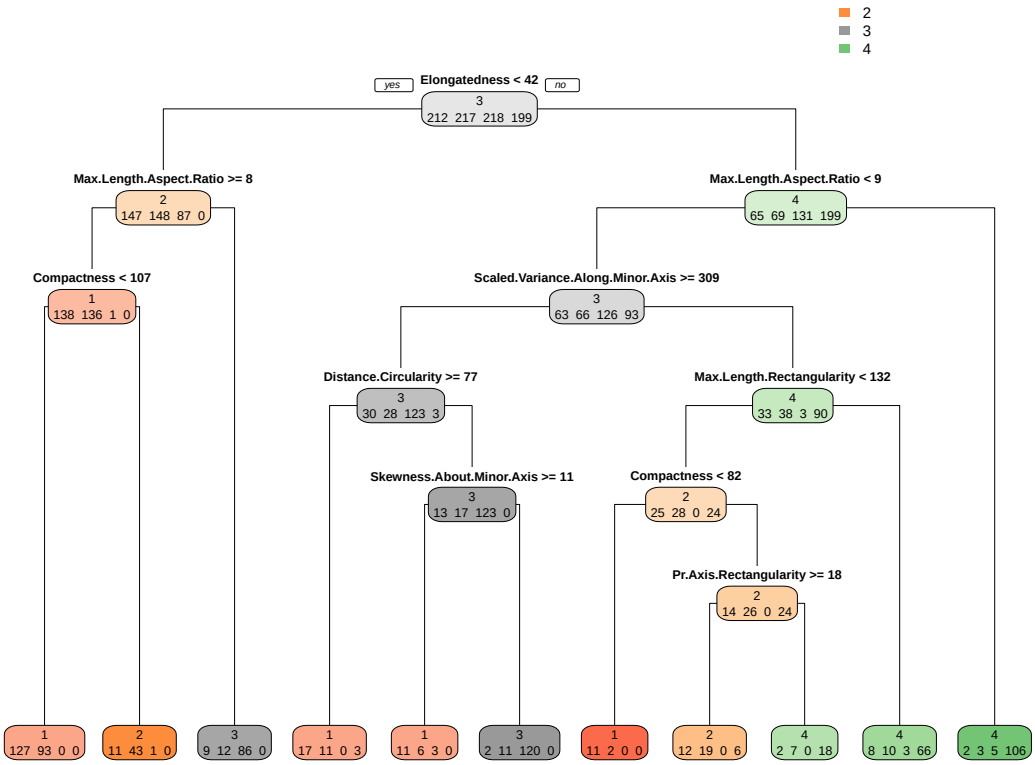


图 3: Pruned Classifiction Tree of Vehicle Data: A pruned classification tree for the vehicle data. There are 18 input variables, 846 observations, and four classes of vehicle models: opel (red), saab (orange), bus (grey), and van (green), whose numbers at each node are given by a/b/c/d, respectively. There are 10 splits and 11 terminal nodes in this tree. The resubstitution error rate is 0.354.