의사결정나무

2011년 10월 27일

제 1 절 의사결정나무

지도학습 문제에서는 최종모형의 예측력과 해석력이 중요하며, 상황에 따라 예측력 또는 해석력을 보다 더 중시하는 경우가 있다. (고객 유치방안을 예측: 예측력, 신용평가: 해석력)

1.1 의사결정나무의 구성요소

- 뿌리마디 (root node): 시작되는 마디로 전체 자료를 포함
- 자식마디 (child node): 하나의 마디로부터 분리되어 나간 2개 이상의 마디들
- 부모마디 (parent node): 주어진 마디의 상위마디
- 끝마디 (terminal node): 자식마디가 없는 마디
- 중간마디 (internal node): 부모마디와 자식마디가 모두 있는 마디
- 가지 (branch): 뿌리마디로부터 끝마디까지 연결된 마디들
- 깊이 (depth): 뿌리마디부터 끝마디까지의 중간마디들의 수

1.2 의사결정나무의 종류

의사결정나무는

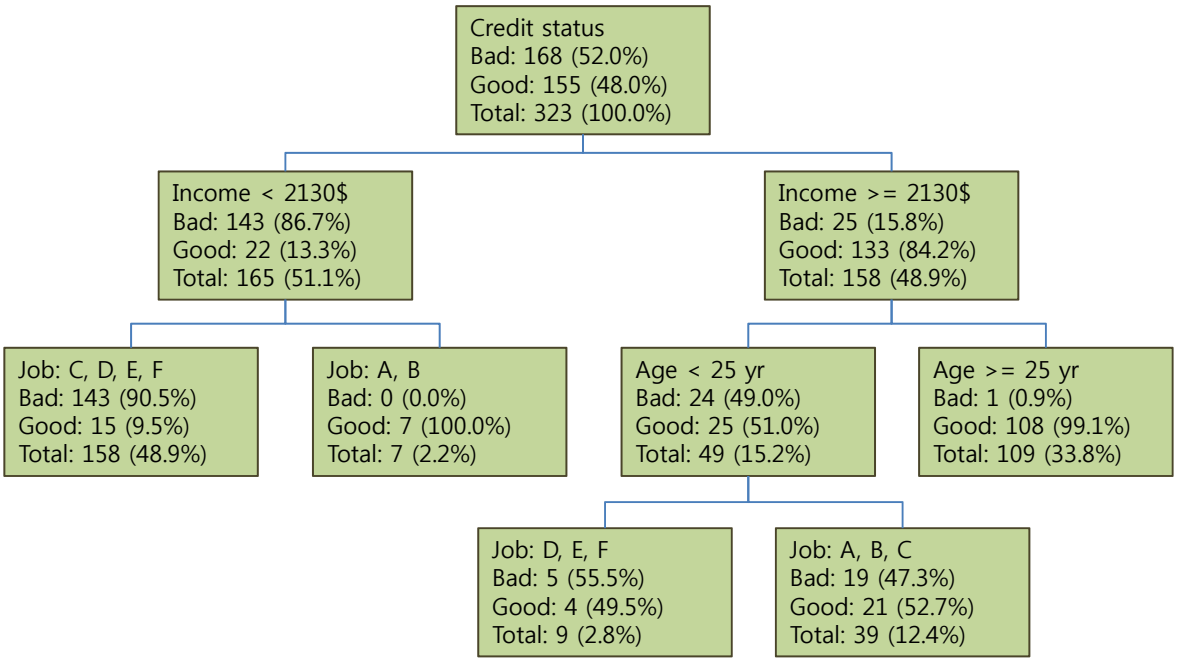


그림 1: 신용자료에 대한 의사결정나무

- 출력변수가 연속형인 회귀나무(regression tree)와
- 범주형인 분류나무(classification tree)로 나눌 수 있다.

1.3 의사결정나무의 형성

의사결정나무에 대하여 다음과 같은 질문을 던질 수 있다.

- 첫째, "뿌리마디의 질문이 왜 소득(income)인가" 라는 질문은 분할기준(splitting rule)의 선택과 관련된다.
- 둘째, "4번, 5번, 7번 마디들은 끝마디인 반면 6번 마디는 왜 중간마디인가" 라는 질문은 분할에 대한 정지규칙(stopping rule)과 관련된다.
- 셋째, "7번 마디에 속하는 자료는 신용상태를 어떻게 결정하여야 하는가" 라는 질문은 각 끝마디에서의 예측값 할당법과 관련된다.

1.4 의사결정나무의 형성과정

의사결정나무의 형성과정은 크게 성장(growing), 가지치기(pruning), 타당성 평가, 해석 및 예측으로 이루어진다.

- 성장 단계는 각 마디에서 적절한 최적의 분리규칙을 찾아서 나무를 성장시키는 과정으로서 적절한 정지규칙을 만족하면 중단한다.
- 가지치기 단계는 오차를 크게 할 위험이 높거나 부적절한 추론규칙을 가지고 있는 가지 또는 불필요한 가지를 제거한다.
- 타당성 평가 단계에서는 이익도표(gain chart), 위험도표(risk chart), 혹은 시험자료를 이용하여 의사결정나무를 평가하게 된다.
- 해석 및 예측 단계에서는 구축된 나무모형을 해석하고 예측모형을 설정한 후 예측에 적용한다.

1.5 회귀나무의 성장

훈련자료를 (x_i, y_i), $i = 1, \ldots, n$ 로 나타내자. 여기서 $x_i = (x_{i1}, \ldots, x_{ip})$ 이다. 나무모형의 성장과정은 x 들로 이루어지는 입력공간을 재귀적으로 분할하는 과정이다.

각 분할 단계에서는 보통 두 영역으로 분할하는 데, 이를 분할과정(split)이라고 부른다.

$$R_1(j, A) = \{x_j \in A\},\ R_2(j, A^c) = \{x_j \in A^c\}$$

- 분리변수(split variable)가 연속형인 경우: $A = \{x_j \leq s\}$
- 분리변수가 범주형인 경우: 예) 전체 범주가 $\{1, 2, 3, 4\}$ 일 때 $A = \{1, 2, 4\}$ 와 $A^c = \{3\}$ 로 나눌 수 있다.

분할(split)은 변수의 선택, 분리점 및 분리기준에 따라 다양하게 나타날 수 있다.

- 최적 분할의 결정은 아래의 불순도 감소량을 가장 크게 하는 분할이다.

$$\Delta i(t) = i(t) - p_L i(t_L) - p_R i(t_R).$$

단,

$$i(t) = \sum_{i \in t} (y_i - \bar{y}_t)^2$$

- 각 단계에서 최적 분리기준에 의한 분할을 찾은 다음 각 분할에 대하여도 동일한 과정을 반복하게 된다.

1.6 회귀나무의 가지치기

- 그러면 언제까지 나무모형을 성장시킬 것인가? 너무 큰 나무모형은 자료를 과대적합하고 반대로 너무 작은 나무모형은 자료를 과소적합할 위험이 있다.
- 즉, 의사결정나무에서는 나무의 크기를 모형의 복잡도로 볼 수 있으며 최적의 나무크기는 자료로부터 추정하게 된다. 일반적으로 사용되는 방법은 마디에 속하는 자료가 일정 수(가령 5) 이하일 때 분할을 정지하고 비용-복잡도 가지치기(cost-complexity pruning)를 이용하여 성장시킨 나무를 가지치기하게 된다.

비용-복잡도 가지치기(cost-complexity pruning)

- 성장시킨 나무모형 T_0를 가지치기하여 얻을 수 있는 나무모형을 $T \subset T_0$로 나타내자.
- $\tilde{T}$는 끝마디의 집합,
- $|T|$는 T에서의 끝마디 개수,
- 비용-복잡도 가지치기는

$$C_\alpha(T) = \sum_{t \in \tilde{T}} i(t) + \alpha |T|$$

로 정의되며 가지치기는 α에 대하여 $C_\alpha(T)$를 최소화하는 $T_\alpha \subset T_0$를 찾는 문제가 된다.

- 여기서 $\alpha \geq 0$는 나무모형의 크기와 자료에 대한 적합도를 조절하는 조율모수로 α값이 크면(작으면) T_α의 크기는 작아(커)진다. $\alpha = 0$이면 가지치기는 일어나지 않고 T_0를 최종모형으로 준다.

추정값 $\hat{\alpha}$은 자료로부터 흔히 5 또는 10-묶음 교차확인오차로 얻을 수 있다. 가지치기된 최종 모형은 $T_{\hat{\alpha}}$으로 나타낼 수 있고 시험자료가 $x \in R_m$이면 $\hat{y} = \hat{c}_m$으로 예측한다.

분류나무

출력변수가 범주형인 분류나무는 다음절에서 소개될 카이제곱 통계량, 지니지수(Gini index), 엔트로피지수(entropy index) 등을 불순도의 측도로 사용하여 회귀나무와 동일한 방식으로 성장시키게 된다. 분류나무의 가지치기는 흔히 오분류율을 불순도의 측도로 사용하여 회귀나무와 동일한 방식으로 실시하여 최종 분류나무모형 $T_{\hat{\alpha}}$ 을 얻게 된다. $\hat{p}_{mk}$ 를 최종모형의 영역 R_m 에 속하는 자료중 출력변수의 범주가 k 인 자료의 비율이라 하자. $x \in R_m$ 이면 그 예측값은 $\hat{y} = \arg\max_k \hat{p}_{mk}$ 로 주어진다. 즉, 분류나무는 예측값을 각 마디에서 다수결(majority vote) 원칙으로 정하는 것이다.

제 2 절 불순도의 여러가지 측도

이 절에서는 의사결정나무의 성장 단계에서 최적 분리기준을 정하는데 사용되는 불순도의 측도에 대하여 알아 보자. 특히 분류나무에서 사용되는 카이제곱(χ^2) 통계량, 지니지수, 엔트로피지수 등에 대하여 살펴보기로 한다.

자료가 아래의 표와 같이 분리된다고 하자.

	Good	Bad	Total
Left	32 (56)	48 (24)	80
Right	178 (154)	42 (66)	220
Total	210	90	300

표의 값은 실제도수(O)와 괄호안의 값인 기대도수(E)를 보여준다. 예를 들어 Left의 Good 셀의 기대도수는 $80 \times 210/300 = 56$ 으로 구하며 다른 셀들에서도 마찬가지 방법으로 기대도수를 구할 수 있다.

카이제곱 통계량은

각 셀에 대한 ((기대도수$-$ 실제도수)의 제곱 / 기대도수)의 합

으로 정의되며, 카이제곱통계량이 최대가 되는 분리를 사용한다. 이 표에서 카이제곱통계량은

$$\frac{(56-32)^2}{56} + \frac{(24-48)^2}{24} + \frac{(154-178)^2}{154} + \frac{(66-42)^2}{66} = 46.75$$

으로 계산된다.

지니지수는

$$2\left(\mathbb{P}(\text{Left에서 Good})\mathbb{P}(\text{Left에서 Bad})\mathbb{P}(\text{Left})\right.$$
$$\left.+\mathbb{P}(\text{Right에서 Good})\mathbb{P}(\text{Right에서 Bad})\mathbb{P}(\text{Right})\right)$$

로 정의되며, 지니지수가 최소가 되는 분리를 선택한다. 앞의 표에 대하여 지니지수를 구하면

$$2\left(\frac{32}{80}\times\frac{48}{80}\times\frac{80}{300}+\frac{178}{220}\times\frac{42}{220}\times\frac{220}{300}\right)=0.355$$

으로 주어진다.

엔트로피지수는

$$\text{엔트로피지수} = \text{엔트로피}(\text{Left})\mathbb{P}(\text{Left}) + \text{엔트로피}(\text{Right})\mathbb{P}(\text{Right})$$

이다. 여기서

$$\begin{aligned}\text{엔트로피}(\text{Left}) &= -\mathbb{P}(\text{Left에서 Good})\log_2\mathbb{P}(\text{Left에서 Good})\\ &\quad -\mathbb{P}(\text{Left에서 Bad})\log_2\mathbb{P}(\text{Left에서 Bad})\end{aligned}$$

로 정의되며, 오른쪽 마디에 대한 엔트로피도 이와 유사하게 정의될 수 있다. 앞의 표에서 엔트로피지수를 구하면

$$-\left(\frac{32}{80}\log_2\left(\frac{32}{80}\right)+\frac{48}{80}\log_2\left(\frac{48}{80}\right)\right)\frac{80}{300}$$
$$-\left(\frac{178}{220}\log_2\left(\frac{178}{220}\right)+\frac{42}{220}\log_2\left(\frac{42}{220}\right)\right)\frac{220}{300}=.7747$$

를 얻을 수 있다.

예제

다음의 자료에 대하여 지니지수를 이용하여 최적의 분리를 찾아보기로 하자.

Temperature	Humidity	Windy	Class
Hot	High	False	N
Hot	High	True	N
Hot	High	False	P
Mild	High	False	P
Cold	Normal	False	P
Cold	Normal	True	N
Cold	Normal	True	P
Mild	High	False	N
Cold	Normal	False	N
Mild	Normal	False	P
Mild	Normal	True	P
Mild	High	True	P
Hot	Normal	False	N
Mild	High	True	P

```
rain=c("Hot" , "High" , "False" , "N",
"Hot" , "High" , "True" , "N",
"Hot" , "High" , "False" , "P",
"Mild" , "High" , "False" , "P",
"Cold" , "Normal" , "False" , "P",
"Cold" , "Normal" , "True" , "N",
"Cold" , "Normal" , "True" , "P",
"Mild" , "High" , "False" , "N",
"Cold" , "Normal" , "False" , "N",
"Mild" ,"Normal" , "False" , "P",
"Mild" , "Normal" , "True" , "P",
"Mild" , "High" , "True" , "P",
"Hot" , "Normal" , "False" , "N",
"Mild" , "High" , "True" , "P")
rain=data.frame(matrix(rain, ncol=4, byrow=T))
```

```
names(rain)=c("Temperature", "Humidity", "Windy", "Class")

library(tree)
tm=tree(Class~., rain, split ="deviance", mincut = 1, minsize = 2, mindev = 0.01)
summary(tm)

Classification tree:
tree(formula = Class ~ ., data = rain, split = "deviance", mincut = 1,
    minsize = 2, mindev = 0.01)
Number of terminal nodes:  7
Residual mean deviance:  1.584 = 11.09 / 7
Misclassification error rate: 0.2857 = 4 / 14

plot(tm, type = "uniform")#"proportional"
text(tm)
```

1. Temperature를 기준으로 분리하는 경우

   - Left={Hot}, Right = {Mild, Cold}일 때

	N	P	계
Left	3	1	4
Right	3	7	10
계	6	8	14

$$\text{지니지수} = 2\left(\frac{1}{4}\times\frac{3}{4}\times\frac{4}{14}+\frac{3}{10}\times\frac{7}{10}\times\frac{10}{14}\right)=0.4071.$$

   - Left={Mild}, Right = {Hot, Cold}일 때

	N	P	계
Left	1	5	6
Right	5	3	8
계	6	8	14

$$\text{지니지수} = 2\left(\frac{1}{6}\times\frac{5}{6}\times\frac{6}{14}+\frac{5}{8}\times\frac{3}{8}\times\frac{8}{14}\right) = 0.3869.$$

- Left={Cold}, Right = {Hot,Mild}일 때

	N	P	계
Left	2	2	4
Right	4	6	10
계	6	8	14

$$\text{지니지수} = 2\left(\frac{2}{4}\times\frac{2}{4}\times\frac{4}{14}+\frac{5}{10}\times\frac{6}{10}\times\frac{10}{14}\right) = 0.4860.$$

2. Humidity를 기준으로 분리하는 경우
   Humidity는 High와 Normal의 두 가지 값만 가지므로 Left를 둘중 어느 한 범주로 잡아도 동일한 결과를 준다. 따라서 편의상 Left={High}, Right = {Normal}라 하자.

	N	P	계
Left	3	4	7
Right	3	4	7
계	6	8	14

$$\text{지니지수} = 2\left(\frac{3}{7}\times\frac{4}{7}\times\frac{7}{14}+\frac{3}{7}\times\frac{4}{7}\times\frac{7}{14}\right) = 0.4897.$$

3. Windy를 기준으로 분리하는 경우
   Humidity와 마찬가지로 범주가 둘이므로 Left={False}, Right = {True}라 하자.

	N	P	계
left	4	4	8
right	2	4	6
계	6	8	14

$$\text{지니지수} = 2\left(\frac{4}{6}\times\frac{2}{6}\times\frac{6}{14}+\frac{4}{8}\times\frac{4}{8}\times\frac{8}{14}\right) = 0.4762.$$

모든 가능한 분리에 대하여 결과를 종합해보면 Temperature에 대하여 Left = {Mild}, Right = {Hot, Cold}로 분리하는 것이 지니지수 측면에서 가장 좋다. 각자 연습삼아 카이제곱 통계량이나 엔트로피지수에 대하여 최적 분리를 찾아보고 지니지수와 동일한 분리를 주는지 확인해 보기 바란다.

제 3 절 여러가지 의사결정나무 알고리즘

다음은 흔히 사용되는 의사결정나무 알고리즘이다.

- CART(classification and regression trees)
  이 장에서 소개된 방법으로 **?**이 개발하였고 가장 널리 사용되는 의사결정나무 알고리즘이다. 불순도의 측도로서 출력변수가 범주형인 경우 지니지수를 이용하고 출력변수가 연속형인 경우에는 분산을 이용하여 이진분리(binary split)를 한다. 개별 입력변수뿐만 아니라 입력변수들의 선형결합들중에서 최적의 분리를 찾을 수도 있다.

- C4.5와 C5.0
  호주의 연구원 **?**에 의하여 개발되었고 초기버전은 ID 3 (iterative dichotomizer 3)이다. CART와는 다르게 각 마디에서 다지분리(multiple split)가 가능하며 범주형 입력변수에 대해서는 범주의 수만큼 분리가 일어난다. 불순도의 측도로는 엔트로피지수를 사용한다.

- CHAID(chi-squared automatic interaction detection)
  ?이 제안한 방법으로 **?**의 AID의 후신으로 볼 수 있다. 가지치기를 하지 않고 적당한 크기에서 나무모형의 성장을 중지하며 입력변수가 반드시 범주형 변수이어야 한다. 불순도의 측도로는 카이제곱 통계량을 사용한다.

제 4 절 의사결정나무의 특징

의사결정나무(특히 CART)는 if-then 형식의 이해하기 쉬운 규칙을 생성하며, 분류작업이 용이하고, 연속형 변수와 범주형 변수를 모두 취급할 수 있으며, 모형에 대한 가정(예: 선형회귀의 선형성, 등분산성 등)이 필요 없는 비모수적 방법이라는 장점이 있다. 또한 가장 설명력이 있는 변수에 대하여 최초로 분리가 일어나는 특징을 가진다.

의사결정나무의 단점은 다음과 같다. 출력변수가 연속형인 회귀모형에서는 그 예측력이 떨어진다. 일반적으로 복잡한 나무모형은 예측력이 저하되고 해석 또한 어려우며, 상황에 따라 계산량이 많을 수도 있으며, 베이즈 분류경계가 사각형(rectangle)이 아닌 경우에는 좋지 않은 결과를 줄 수 있다는 것이다. 특히 자료에 약간의 변화가 있는 경우에 전혀 다른 결과를 줄 수도 있는 (즉, 분산이 매우 큰) 불안정한 방법이다. 참고로 **??**장에서 소개하는 배깅(bagging)과 같은 앙상블(ensemble) 알고리즘을 적용하여 의사결정나무의 분산을 줄일 수도 있다.

제 5 절 R 예제

이 절에서는 R. A. Fisher의 붓꽃(iris)자료에 대하여 의사결정나무 모형을 적합한다. 붓꽃자료는 꽃받침의 길이(sepal length), 꽃받침의 폭(sepal width), 꽃잎의 길이(petal length), 꽃잎의 폭(petal width), 붓꽃의 품종(setosa, versicolor, virginica)으로 이루어졌다. 그림 2는 입력변수들에 대한 산점도 행렬로서 각 품종을 표시한 것이다. 붓꽃자료에 대하여 붓꽃의 품종을 출력변수로 하고 나머지는 입력변수로 하는 다음과 같이 의사결정나무를 적합하였다.

```
> library(MASS)
> library(tree)
> data(iris)
> library(MASS)
> library(tree)
> data(iris)
> plot(iris[,1:4], col=as.integer(iris$Species),
+    pch=substring((iris$Species),1,1))
> ir.tr = tree(Species ~., iris)
> summary(ir.tr)

Classification tree:
tree(formula = Species ~ ., data = iris)
Variables actually used in tree construction:
[1] "Petal.Length" "Petal.Width"  "Sepal.Length"
Number of terminal nodes:  6
```

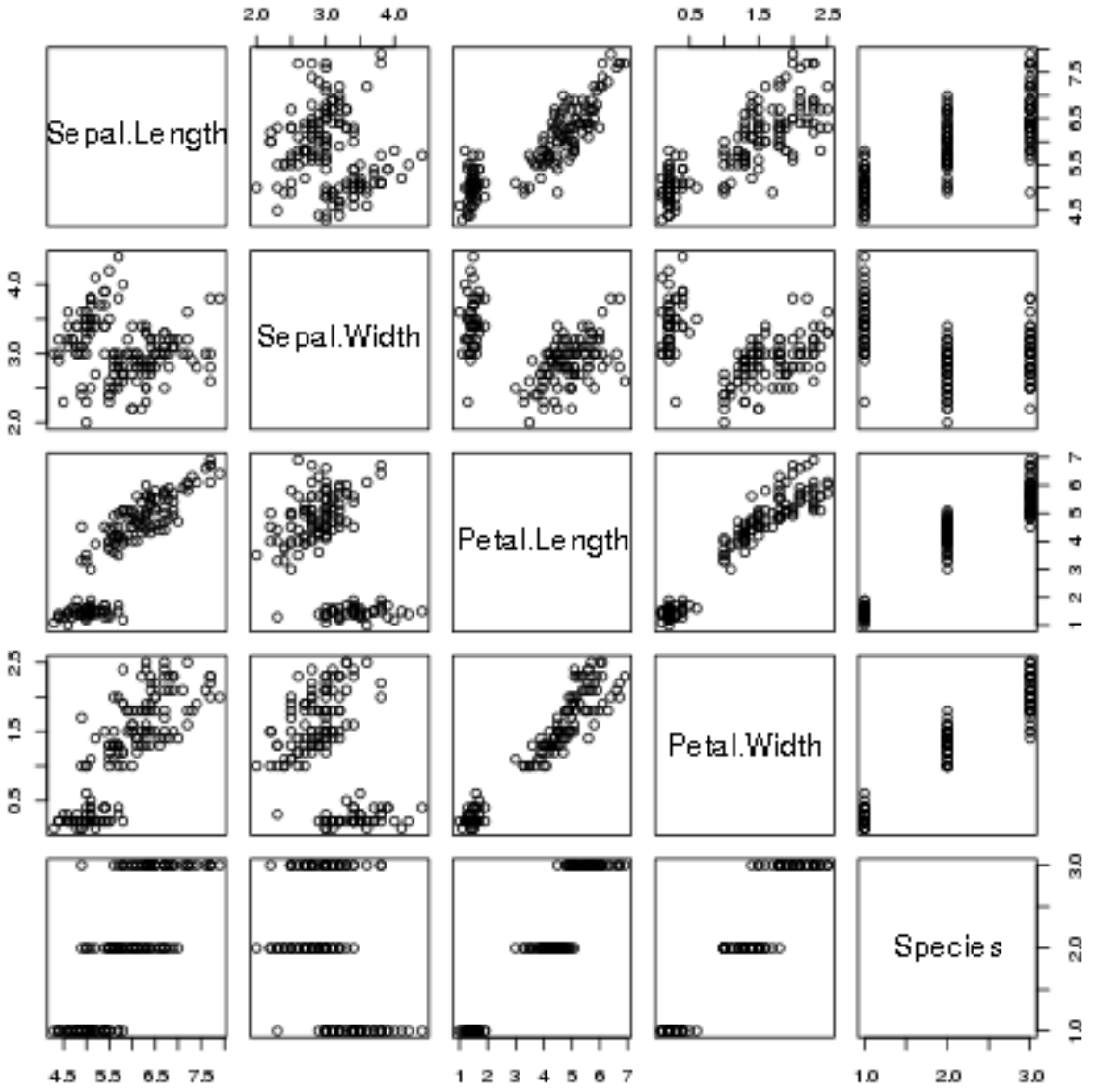


그림 2: 붓꽃 자료에 대한 산점도 행렬

```
Residual mean deviance:  0.1253 = 18.05 / 144
Misclassification error rate: 0.02667 = 4 / 150
> ir.tr
node), split, n, deviance, yval, (yprob)
      * denotes terminal node

1) root 150 329.600 setosa ( 0.3333 0.3333 0.3333 )
 2) Petal.Length<2.45 50   0.000 setosa (1.0000 0.0000 0.0000) *
 3) Petal.Length>2.45 100 138.600 versicolor (0.0000 0.5000 0.5000)
   6) Petal.Width<1.75 54  33.320 versicolor (0.0000 0.9074 0.0925)
     12) Petal.Length<4.95 48   9.721 versicolor (0.0000 0.9791 0.0208)
      24) Sepal.Length<5.15 5   5.004 versicolor (0.0000 0.8000 0.2000) *
      25) Sepal.Length>5.15 43   0.000 versicolor (0.0000 1.0000 0.0000) *
    13) Petal.Length>4.95 6   7.638 virginica (0.0000 0.3333 0.6666) *
   7) Petal.Width>1.75 46   9.635 virginica (0.0000 0.0217 0.9782)
    14) Petal.Length<4.95 6   5.407 virginica (0.0000 0.1666 0.8333) *
    15) Petal.Length>4.95 40   0.000 virginica (0.0000 0.0000 1.0000) *
>
> plot(ir.tr)
> text(ir.tr, all = T)
```

여기서 yprob은 마디내의 클래스별 분포를 나타내며 그림 3은 가지치기 이전의 나무모형을 보여준다. 다음과 같이 snip.tree 함수로 마디 7과 12를 제거한 후 그 결과를 ir.tr1로 저장하였다.

```
> ir.tr1 = snip.tree(ir.tr, nodes = c(12, 7))
> plot(ir.tr1)
> text(ir.tr1, all = T)
>
> par(pty = "s")
> plot(iris[, 3],iris[, 4], type="n",
+ xlab="petal length", ylab="petal width")
> text(iris[, 3], iris[, 4], c("s", "c", "v")[iris[, 5]])
> partition.tree(ir.tr1, add = TRUE, cex = 1.5)
```

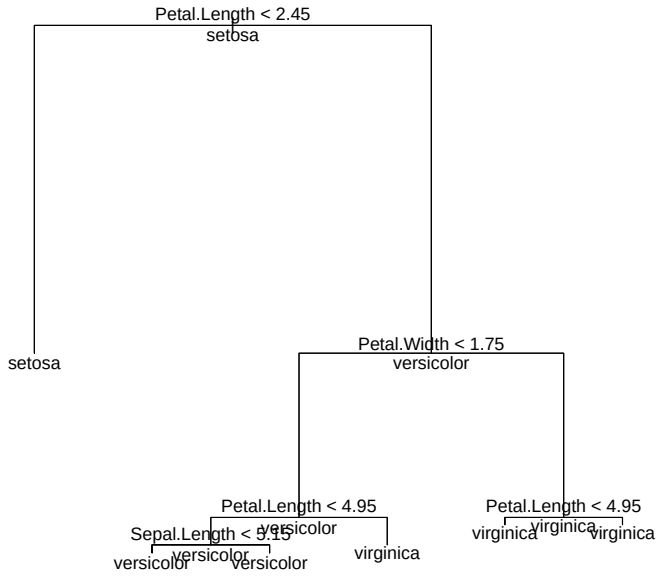


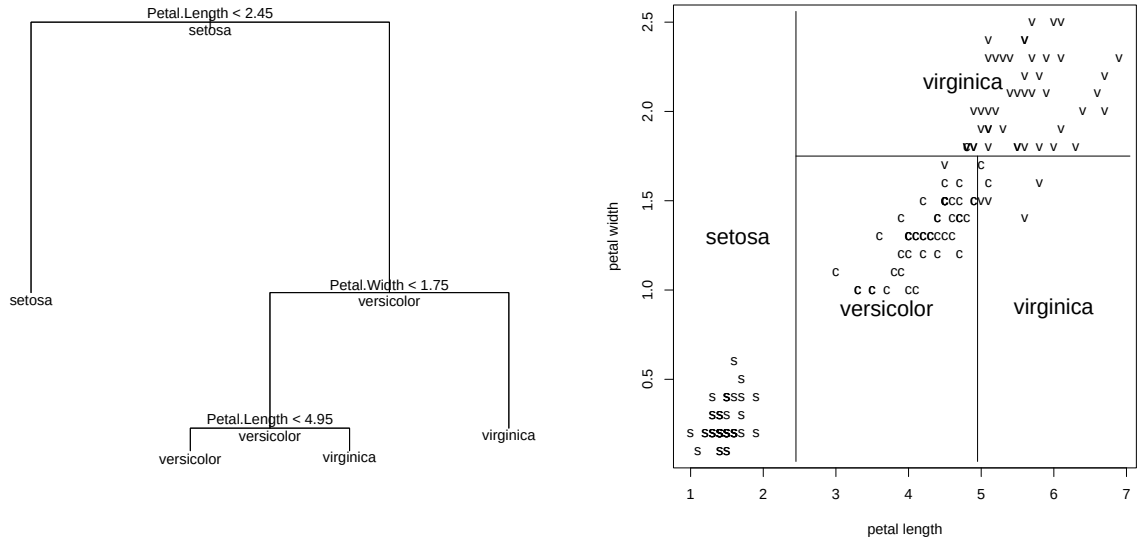
그림 3: 붓꽃 자료에 대한 가지치기 이전의 나무모형

그림 4는 가지치기 이후의 나무모형과 꽃잎 길이와 폭을 축으로 나무모형에 의한 영역 분할을 보여준다. 다음은 오분류율을 이용한 가지치기를 보여준다. 그림 3에서 끝마디의 수를 4로 하여 가지치기한 결과 그림 5 (b)의 나무모형을 얻었다. 이 나무모형은 앞에서 수동으로 마디 7과 12를 가지치기한 그림 4 (a)와 동일하다.

```
> ir.tr2 = prune.misclass(ir.tr)
> plot(ir.tr2)
> fin.tr = prune.misclass(ir.tr, best=4)
> plot(fin.tr)
> text(fin.tr, all = T)
```

`prune.misclass` 함수에서 best의 값을 지정하지 않는 경우 끝마디의 수(size), deviance, k(cost-complexity measure)를 출력한다. 이 경우 k가 최소가 되는 끝마디를 선택할 수 있다.

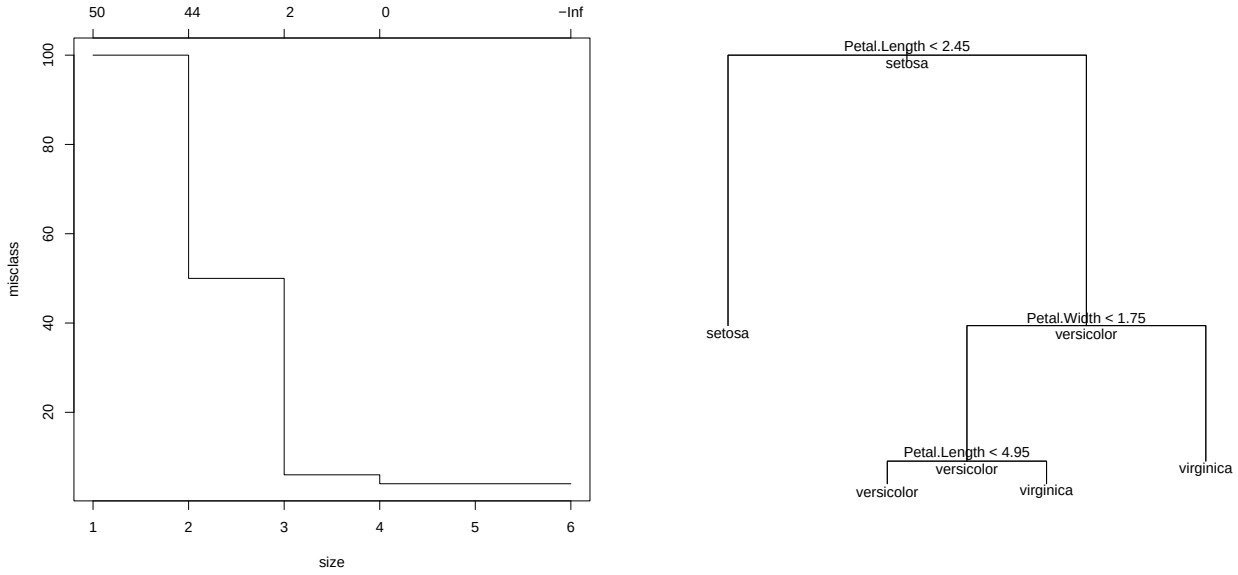
```
> ir.tr2
```



(a) 나무모형 (b) 분할영역

그림 4: 붓꽃 자료에 대하여 가지치기를 적용한 나무모형과 분할영역

```
$size
[1] 6 4 3 2 1
$dev
[1]   4   4   6  50 100
$k
[1] -Inf    0    2   44   50
$method
[1] "misclass"
attr(,"class")
[1] "prune"         "tree.sequence"
```



(a) CV (b) 가지치기 이후

그림 5: 붓꽃 자료에 대하여 오분류율을 이용한 나무모형의 가지치기

연습문제

1. `mlbench` 패키지의 Vehicle, Glass, BreastCancer 자료에 대하여 적절한 크기의 나무모형을 적합하라.

2. 다음의 코드를 이용하여 자료를 생성하라.

```
x1 = rnorm(300)
x2 = rnorm(300)
error = 0.1 * rnorm(300)
y1 = 1 + (2 * x1) + (3 * x2) + error
```

   (a) `lm` 함수로 선형회귀모형을 적합하라. 또한 `rpart` 함수로 나무모형을 적합하고 `text` 함수로 나무모형의 결과를 출력하라. 회귀모형과 나무모형의 결과를 실제 자료가 생성된 모형과 비교하라. 적합된 모형이 자료를 잘 적합하는지 살펴보기 위해 `predict` 함수로 적합된 값과 실제값을 비교해 보아라.

   (b) 다음과 같이 이항 인자(factor) 변수를 입력으로 하는 반응변수 y를 생성한다.

```
x11 = (x1 > 0)
x22 = (x2 > 0)
y = 1 + (2 * x11) + (3 * x22) + error
```

(a)와 동일한 방식으로 선형회귀와 나무모형을 적합하고 그 결과를 비교하라.

(c) (a)와 (b)는 각각 자료가 선형 가법 모형과 가법 계단함수(step function) 모형에서 생성되는 경우에 해당된다. 모형의 적합도와 해석력을 기준으로 했을 때 어떤 경우에 나무모형이 선형회귀보다 더 좋은가?