R로 배우는 데이터사이언스

의사결정나무와 앙상블

원중호

서울대학교 통계연구소

예측모형 구축 방법론 입문

전통적인 학습과 기계학습

- 전통적인 학습
  - 자료의 관측이나 직관으로 가설 설정
  - 실험 및 통계추론
  - 결론 도출
- 기계학습
  - 데이터를 모은다.
  - 데이터를 분석한다.
  - 새로운 지식을 찾는다.

"Let the data tell something."

기계학습으로 풀 수 있는 문제들

- 심장마비의 원인이 뭐지?
- 인종별로 뭐가 다르지?
- 주가는 왜 움직이지?
- 누가 우리 회사에 도움이 되는 고객이지?
- 개하고 고양이를 어떻게 구별하지?
- 음성을 어떻게 인식하지?
- 바둑에서 어떻게 하면 이기지?

기계학습의 여러 분야

- 지도학습 (supervised learning)
  - 입력변수를 이용해서 출력변수를 예측
  - 예: 회귀분석, 분류분석
  - 응용: 이미지분류, 고장 원인 파악, 암 진단
- 비지도학습 (unsupervised learning)
  - 데이터간의 복잡한 관계를 규명 (출력변수가 없음)
  - 예: 군집분석, 주성분분석, 결합분포추정 (예: 요인분석)
  - 응용: 차원축소, 이미지 압축/생성, Source 분해
- 강화학습
  - 행동에 따라 변화하는 환경에서 최적의 의사결정 방법을 학습
  - 예: multi-armed badit problem, Markov decision process
  - 응용: AlphaGo, 각종 게임, 로봇

지도학습 기본 구조

- 입력 (input, covariate) : $\mathbf{x} \in \mathbb{R}^p$
- 출력 (output, response) : $y \in \mathcal{Y}$
- 학습자료 (training data): $\mathcal{L} = \{(y_i, \mathbf{x}_i), \quad i = 1, ..., n\}$ (관측가능)
- 예측자료 (test data): $\mathcal{T} = \{(Y_i, \mathbf{x}_i), \quad i = 1, ..., N\}$ (관측불가)
- 목적: 예측자료에서 주어진 입력변수 $\mathbf{x}$에 대해서 출력변수 y를 잘 예측하는 함수 (즉, 예측모형) $f^0 : \mathbb{R}^p \to \mathcal{Y}$를 학습자료를 이용하여 추정하는 것
- 예측모형: $\hat{f}(\mathbf{x}) = f(\mathbf{x}, \mathcal{L})$.
- 예측: If new input is $\mathbf{x}$, predict unknown y by $\hat{f}(\mathbf{x})$.
  - y is categorical $\Rightarrow$ classification
  - y is continuous $\Rightarrow$ regression

예측모형 구축 방법론

- 모형선택: 좋은 예측모형이 속해 있을 것으로 예상되는 함수집합 $\mathcal{F}$을 선정
  - 선형모형
  - 의사결정나무
- 모형추정: 선택된 모형 $\mathcal{F}$중에서 학습자료를 잘 예측하는 모형 추정
  - 선형모형:최소제곱 추정량
  - 의사결정나무: 성장 및 가지치기

예측모형 구축시 고려사항

- 과적합을 피해야 한다.
  - “학습자료를 너무 잘 예측하면 예측자료에서 예측력이 떨어진다.’’
- 학습자료에서의 학습오차는 예측자료에서의 예측오차와 많이 다를 수 있음.
- 과적합을 피하는 방법: 추정된 예측모형의 예측자료에서의 예측오차를 추정

예측자료에서의 예측오차의 추정

- 자료의 분할
  - 학습자료를 두개로 분할하여 첫번째 자료에서 예측모형을 추정하고 두번째 자료에서 예측오차를 구한다.
  - 보통 무작위로 분할한다. 예를 들면 전체 학습자료의 70%에서 예측모형을 추정하고 나머지 30%에서 예측오차를 구한다.
- 교차검증 (cross-validation)
  - 학습자료가 크지 않은 경우 자료를 두 벌로 분할하면 자료의 수가 너무 작아서 효율이 떨어짐
  - 자료를 K등분 한후 $K-1$벌의 자료로 예측모형을 추정하고 나머지 한 벌의 자료에서 예측오차를 구함
  - 이러한 과정을 K번 반복하여 K개의 예측오차를 구하고 이를 평균하여 최종 예측오차를 구함.

교차검증 예제 I

- 다음은 입력변수가 2개인 6개의 회귀모형 자료이다. obs 번호로 자료의 집합을 나타내기로 하자. 전체 자료는 $D = \{1, 2, 3, 4, 5, 6\}$로 표기할 수 있다.

obs	y	x_1	x_2
1	1	2	1
2	3	7	0
3	5	3	0
4	3	5	1
5	2	1	1
6	7	3	0

교차검증 예제 II

- 3벌 교차검증을 사용하여 다음 두 모형 중 최적 모형을 선택하는 것이 목표이다.
  - 모형 1: $y = \alpha + \beta_1 x_1$
  - 모형 2: $y = \alpha + \beta_1 x_1 + \beta_2 x_2$

교차검증 예제 III

- 1단계: $D = \{1, 2, 3, 4, 5, 6\}$을 $D_1 = \{1, 2\}$, $D_2 = \{3, 4\}, D_3 = \{5, 6\}$ 세 벌로 나눈다.
- 2단계
  - D_1을 제외한 나머지 D_2, D_3 를 사용하여 회귀계수를 추정한다.
    - 모형 1: $y = 3.50 + 0.25x_1$
    - 모형 2: $y = 10.5 - 1.50x_1 + 7.00x_2$
  - 위에서 추정한 모형의 D_1에 대한 예측오차를 구한다.
    - 모형 1 예측오차: 14.06
    - 모형 2 예측오차: 100.25

교차검증 예제 IV

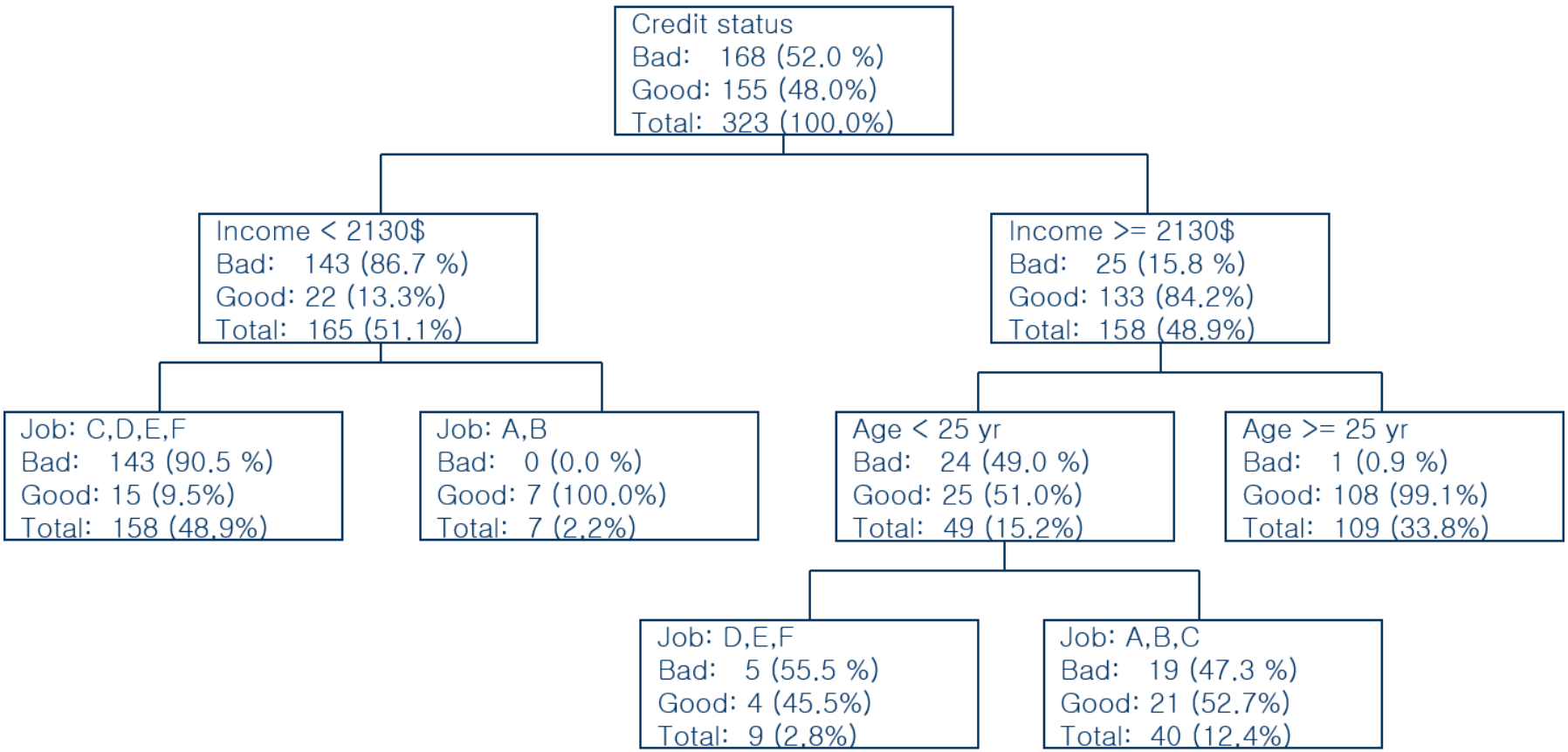
- 3단계
  - D_2을 제외한 나머지 D_1, D_3 를 사용하여 회귀계수를 추정한다.
    - 모형 1: $y = 2.66 + 0.18x_1$
    - 모형 2: $y = 6.27 + 0.24x_1 - 5.08x_2$
  - 위에서 추정한 모형의 D_2에 대한 예측오차를 구한다.
    - 모형 1 예측오차: 3.54
    - 모형 2 예측오차: 24.10
- 4단계
  - D_3을 제외한 나머지 D_1, D_2 를 사용하여 회귀계수를 추정한다.
    - 모형 1: $y = 2.42 + 0.13x_1$
    - 모형 2: $y = 3.17 + 0.20x_1 - 2.10x_2$
  - 위에서 추정한 모형의 D_3에 대한 예측오차를 구한다.
    - 모형 1 예측오차: 17.69
    - 모형 2 예측오차: 10.80

교차검증 예제 V

- 5단계: 2,3,4 단계에서 구한 오차를 모형별로 더한다.
  - 모형 1 총 예측오차: $14.06 + 3.54 + 17.69 = 35.29$
  - 모형 2 총 예측오차: $100.25 + 24.10 + 10.80 = 135.15$
- 6단계 : 오차의 합이 작은 모형 1을 최적모형으로 선택한다.
- 7단계 : 선택된 모형 1의 회귀계수를 전체 자료 D에 대해 추정한다.
  - 최종 모형: $y = 3.12 + 0.10x_1$

의사결정나무

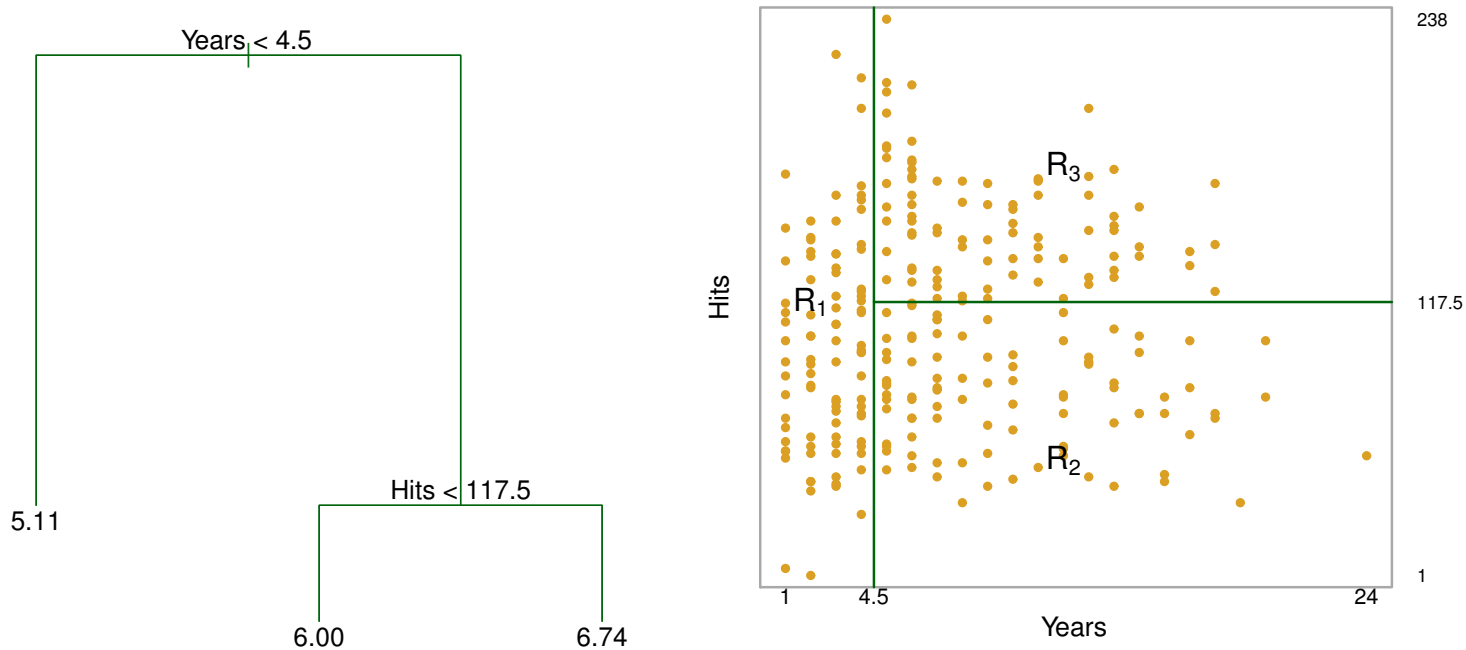
의사결정나무의 예


Credit status
Bad: 168 (52.0 %)
Good: 155 (48.0%)
Total: 323 (100.0%)
Income < 2130$
Bad: 143 (86.7 %)
Good: 22 (13.3%)
Total: 165 (51.1%)
Income >= 2130$
Bad: 25 (15.8 %)
Good: 133 (84.2%)
Total: 158 (48.9%)
Job: C,D,E,F
Bad: 143 (90.5 %)
Good: 15 (9.5%)
Total: 158 (48.9%)
Job: A,B
Bad: 0 (0.0 %)
Good: 7 (100.0%)
Total: 7 (2.2%)
Age < 25 yr
Bad: 24 (49.0 %)
Good: 25 (51.0%)
Total: 49 (15.2%)
Age >= 25 yr
Bad: 1 (0.9 %)
Good: 108 (99.1%)
Total: 109 (33.8%)
Job: D,E,F
Bad: 5 (55.5 %)
Good: 4 (45.5%)
Total: 9 (2.8%)
Job: A,B,C
Bad: 19 (47.3 %)
Good: 21 (52.7%)
Total: 40 (12.4%)

의사결정나무 개요 I

- 예측 변수의 공간을 분할하여 분할된 부분에서 반응변수의 값을 예측하는 회귀분석 혹은 분류 방법.
- 예측변수의 공간을 분할하기 때문에 해석이 쉽다.
- 예측성능은 보통 좋지 않다.
- 배깅(bagging), 랜덤포레스트(random forest) 등과 같이 다수의 나무를 합치면 종종 매우 좋은 결과를 나타낸다.
- 반응변수 Y가 범주형인가, 연속형인가에 따라 분류나무, 회귀나무로 나뉜다.

의사결정나무 개요 II

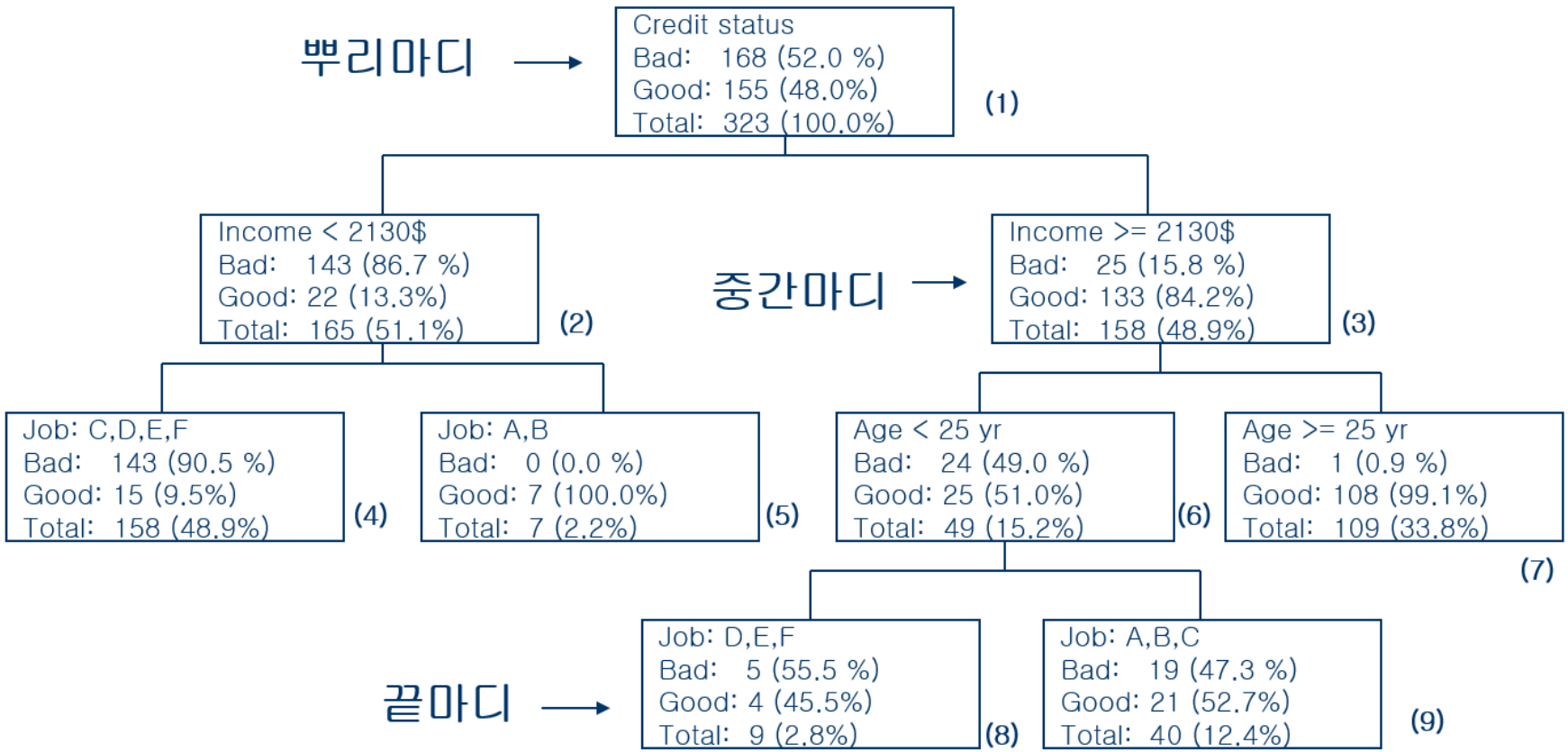


출처: James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). “An Introduction to Statistical Learning.” Springer, New York.

의사결정나무의 구성요소 I

- 뿌리마디(root node): 나무구조가 시작되는 마디로 전체 자료로 이루어져 있다.
- 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): 하나의 마디로부터 끝마디 까지 연결된 일련의 마디들
- 깊이(depth): 뿌리마디부터 끝마디 까지의 중간마디의 수

의사결정나무의 구성요소 II



의사결정나무의 구축방법

- 키우기(growing): 각 마디에서 적절한 최적의 분리규칙을 찾아서 나무를 성장시킨다. 정지규칙을 만족하면 성장을 중단한다.
- 가지치기(pruning): 분류오류를 크게 할 위험이 높거나 부적절한 추론규칙을 가지고 있는 가지를 제거한다.
- 타당성 평가: 이익도표(gain chart)나 위험도표(risk chart) 또는 검증용 자료(test sample), 또는 교차검증(cross validation) 등을 이용하여 의사결정나무를 평가한다.
- 해석 및 예측: 구축된 나무모형을 해석하고 예측모형을 설정한다.

분리규칙

- 각 마디에서 분리규칙은 분리에 사용될 입력변수 (분리변수, split variable)의 선택과 분리가 이루어질 기준 (분리 기준, split criteria)를 정해야 한다.
- 분리변수(X)가 연속형인 경우에는 분리 기준(c)은 하나의 숫자로 주어지며, 일반적으로 분리변수 X가 c보다 작으면 왼쪽 자식마디로 X가 c보다 크면 오른쪽 자식마디로 자료를 분리한다.
- 분리변수가 범주형인 경우에는 분리기준은 전체 범주를 두 개의 부분집합으로 나누는 것이 된다.

순수도

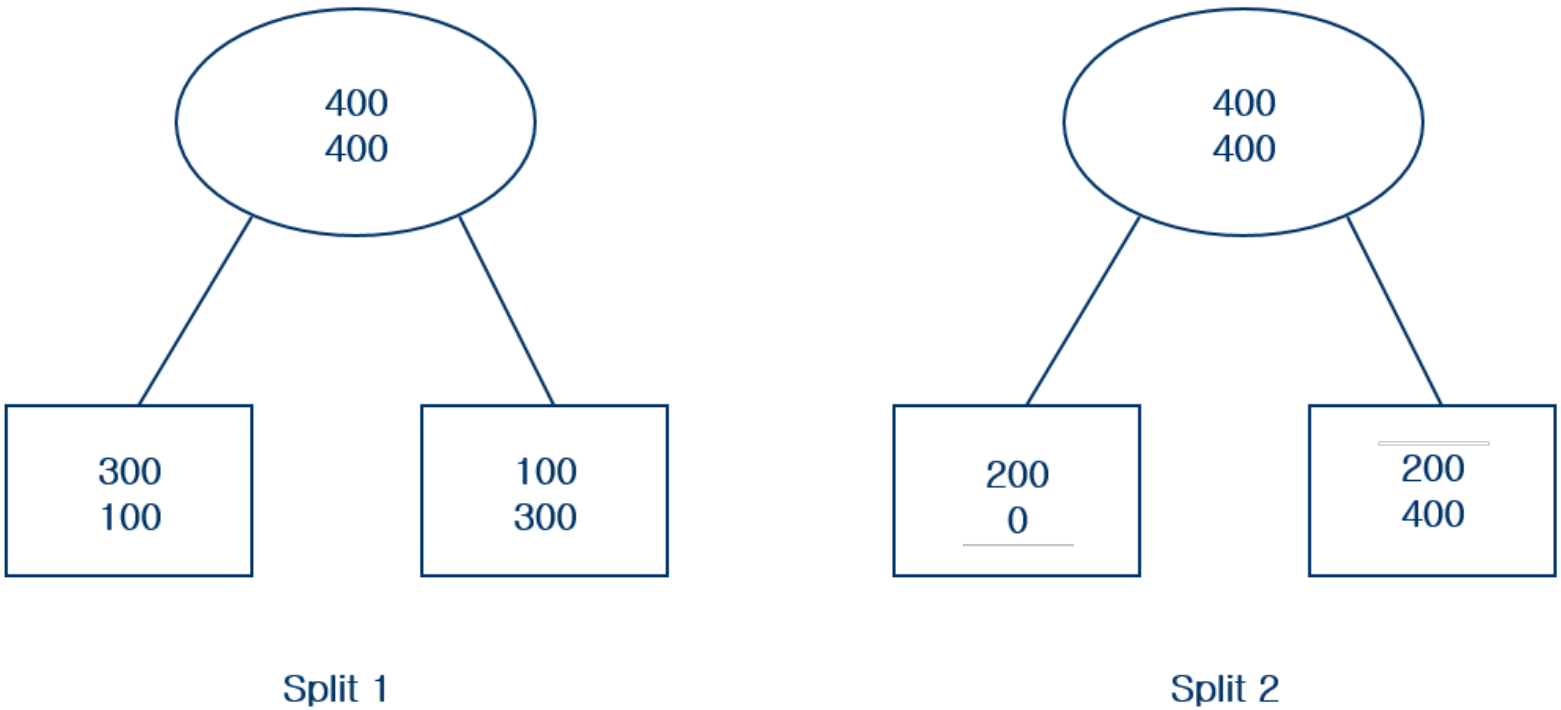
- 각 마디에서 분리변수와 분리기준은 목표변수의 분포를 가장 잘 구별해주는 쪽으로 정한다. 목표변수의 분포를 얼마나 잘 구별하는가에 대한 측정치로 순수도(purity) 또는 불순도(impurity)를 사용한다.
- 예를 들어 그룹 0과 그룹 1의 비율이 45%와 55%인 마디는 각 그룹의 비율이 90%와 10%인 마디에 비하여 순수도가 낮다 (또는 불순도가 높다)라고 이야기 한다.
- 각 마디에서 분리변수와 분리 기준의 설정은 생성된 두 개의 자식마디의 순수도의 합이 가장 큰 분리변수와 분리기준을 선택한다.

순수도의 조건 I

- 의사결정나무는 각 마디에서 분리에 의해 커진다.
- 분리는 분리 후에 자식마디가 부모마디보다 순수하도록 이루어진다.
- 자식마디가 부모마디보다 순수하다는 것은 분리 후에 각 마디 안에 있는 자료의 구성이 어느 한 그룹에만 해당하는 비율이 높다는 것이다.
- 분리를 할 때, 분리변수의 선택은 불순도의 감소를 최대로 만드는 변수를 선택한다.

순수도의 조건 II

- 불순도의 측정으로 가장 쉽게 생각할 수 있는 것이 오분류율이다.
- 그러나 다음의 그림을 보면, split 1과 split 2에서 오분류율은 200/800으로 같지만, 다음 단계의 분리를 생각하면 split 2가 더 바람직하다.



순수도의 조건 III

- 이렇듯, 한 마디에서 오분류율을 감소시키는 것이 나무 전체의 오분류율을 감소시키는 것이 아니다.
- 이러한 점에서, 단순히 오분류율을 불순도로 생각하는 것은 바람직하지 않다.
- 적절한 불순도는 두 개의 자식마디 중 어느 한쪽의 오분류율이 아주 작은 경우에 적어지는 것이 바람직하다.
- 이러한 관점에서 split 2가 더 작은 불순도를 갖는다고 말할 수 있다.

불순도 측정량

- 분류모형
  - 카이제곱 통계량(chi-square statistics)
  - 지니지수(Gini index)
  - 엔트로피지수(entropy index)
- 회귀모형
  - 분산분석에 의한 F-통계량(F-statistics)
  - 분산의 감소량
  - 모든 분리변수와 분리기준 쌍에서 불순도를 가장 작게 하는 변수와 기준을 선택하고 이를 이용하여 분리를 수행한다.

예제

- 주어진 분리변수와 분리기준에 의하여 다음의 표를 작성할 수 있다.

	good	bad	total
left	32	48	80
right	178	42	220
total	210	90	300

지니지수 (Gini index)

- 한 집단의 지니지수는 다음과 같이 구한다:
  지니지수 = (good의 확률) x (good이 아닐 확률) + (bad의 확률) x (bad가 아닐 확률)
- 앞의 표에서 기존 지니지수 = $\frac{210}{300}\frac{90}{300} + \frac{90}{300}\frac{210}{300} = 0.42$
- 분리기준을 따랐을 경우:
  - (지니지수) = $\frac{80}{300}$(좌측 지니지수) + $\frac{220}{300}$(우측 지니지수)
    - (좌측 지니지수) = $\frac{32}{80}\frac{48}{80} + \frac{48}{80}\frac{32}{80} = 0.48$
    - (우측 지니지수) = $\frac{178}{220}\frac{42}{220} + \frac{42}{220}\frac{178}{220} = 0.3089$
    - (지니지수) = $\frac{80}{300}(0.48) + \frac{220}{300}(0.3089) = 0.3545$

엔트로피 지수 (entropy index)

- 엔트로피는 다음과 같이 구한다:
  (엔트로피) = -(good의 확률) × log(good의 확률) + (bad의 확률) × log(bad의 확률)
- 앞의 표에서
  (기존 엔트로피) = $-\frac{210}{300}\log\frac{210}{300} - \frac{90}{300}\log\frac{90}{300} = 0.6109$
- 분리기준을 따랐을 경우:
  - (엔트로피) = $\frac{80}{300}$(좌측 엔트로피}) + $\frac{220}{300}$(우측 엔트로피)
    - (좌측 엔트로피) = $-\frac{32}{80}\log\frac{32}{80} - \frac{48}{80}\log\frac{48}{80} = 0.6730$
    - (우측 엔트로피) = $-\frac{178}{220}\log\frac{178}{220} - \frac{42}{220}\log\frac{42}{220} = 0.4875$
    - (엔트로피) = $\frac{80}{300}(0.6730) + \frac{220}{300}(0.4875) = 0.5369$

분리방법: 예제

- 아래의 자료에 대해 지니지수를 이용한 최적의 분리를 찾아보자.

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

분리방법: 예제

1. Temperature를 기준으로 분리
   1) left node={hot}, right node={mild, cold}

	N	P	total
left	3	1	4
right	3	7	10
total	6	8	14

- Gini index$= \frac{4}{14}\frac{3}{4}\frac{1}{4} + \frac{10}{14}\frac{3}{10}\frac{7}{10} = 0.2036$

분리방법: 예제

1. Temperature를 기준으로 분리
   2) left node={mild}, right node={hot, cold}

	N	P	total
left	1	5	6
right	5	3	8
total	6	8	14

- Gini index$= \frac{6}{14}\frac{1}{6}\frac{5}{6} + \frac{8}{14}\frac{5}{8}\frac{3}{8} = 0.1934$

분리방법: 예제

1. Temperature를 기준으로 분리
   3) left node={cold}, right node={hot, mild}

	N	P	total
left	2	2	4
right	4	6	10
total	6	8	14

- Gini index$= \frac{4}{14}\frac{2}{4}\frac{2}{4} + \frac{10}{14}\frac{4}{10}\frac{6}{10} = 0.2429$

분리방법: 예제

2. Humidity를 기준으로 분리
   1) left node={high}, right node={normal}

	N	P	total
left	3	4	7
right	3	4	7
total	6	8	14

- Gini index$= \frac{7}{14}\frac{3}{7}\frac{4}{7} + \frac{7}{14}\frac{3}{7}\frac{4}{7} = 0.2449$

분리방법: 예제

3. Windy를 기준으로 분리
   1) left node={false}, right node={true}

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

- Gini index$= \frac{8}{14}\frac{4}{8}\frac{4}{8} + \frac{6}{14}\frac{2}{6}\frac{4}{6} = 0.2381$

분리방법: 예제

- 1, 2, 3의 결과를 종합하여 불순도가 가장 작은 분리를 선택한다.
- 따라서, 1번 Temperature를 기준으로 마디를 분리하며, 이 때 분리 기준은 {mild} 와 {hot, cold}가 된다.

회귀모형에서 불순도의 측정

- 오른쪽 자식마디와 왼쪽자식마디의 평균의 차이를 검정하는 t-통계량의 유의확률이 가장 작은 분리변수와 분리기준을 사용하여 분리를 수행한다.
- 왼쪽자식마디의 자료의 분산과 오른쪽 자식마디의 자료의 분산의 합이 가장 작은 분리를 선택한다.

정지규칙

- 현재의 마디가 더 이상 분리가 일어나지 못하게 하는 규칙이다.
- 규칙의 종류로는
  - 모든 자료가 한 그룹에 속할 때
  - 마디에 속하는 자료가 일정 수 이하일 때
  - 불순도의 감소량이 아주 작을 때
  - 뿌리마디로부터의 깊이가 일정 수 이상일 때 등이 있다.

가지치기 (pruning)

- 지나치게 많은 마디를 가지는 의사결정나무는 새로운 자료에 적용할 때 예측오차가 매우 클 가능성이 있다.
- 성장이 끝난 나무의 가지를 적당히 제거하여 적당한 크기를 갖는 나무모형을 최종적인 예측모형으로 선택하는 것이 예측력의 향상에 도움이 된다.
- 적당한 크기를 결정하는 방법은 평가용 자료(validation data)를 사용하거나 교차검증을 이용하여 예측오차를 구하고 이 예측오차가 가장 작은 나무모형을 선택한다.

가지치기 (pruning) 과정

- 주어진 나무 T와 양수 α에 대하여 비용 복잡도(cost complexity)는 다음과 같이 정의 된다.

$$\text{비용복잡도}(\alpha) = \text{나무 } T\text{의 오분류율} + \alpha|T|$$

  여기서 $|T|$는 나무 T의 끝마디의 개수이다.
- 일반적으로 나무가 커지면 (즉 $|T|$가 커지면) 오분류율을 줄게 된다. 하지만 $\alpha|T|$이 증가하여 비용복잡도는 항상 감소하지 않는다.
- 나무성장과정을 통하여 생성된 큰 나무 T_m 대하여, 주어진 α에 대하여 비용복잡도(α)를 최소로 하는 T_m의 부분나무를 $T(\alpha)$라 하자. 일반적으로 α가 크면 $T(\alpha)$의 크기가 작아진다.

의사결정나무의 장점

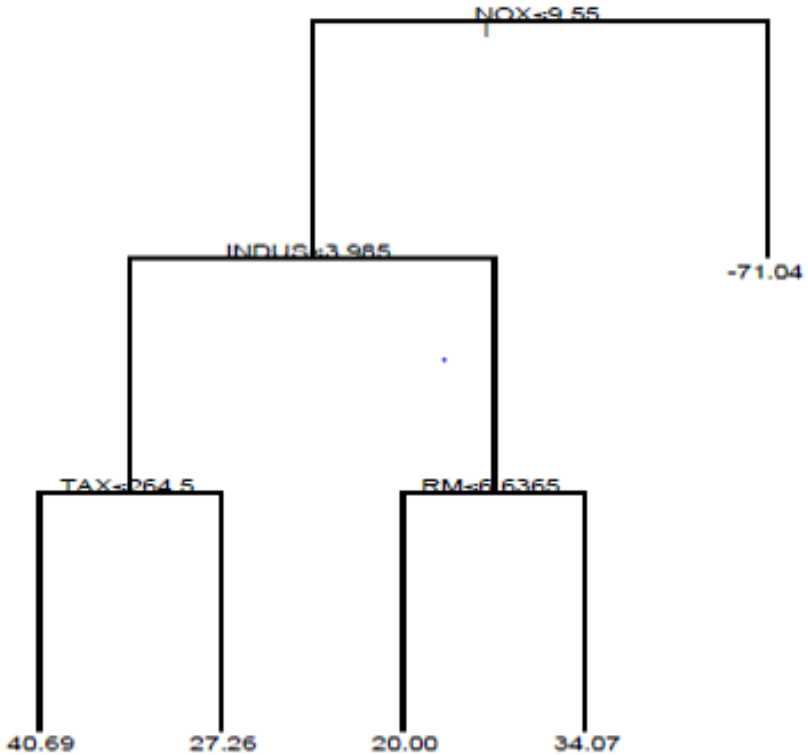
- 이해하기 쉬운 규칙을 생성시켜 준다.
- 분류작업이 용이하다.
- 연속형변수와 범주형 변수를 모두 다 취급할 수 있다.
- 가장 좋은 변수를 명확히 알아낸다.
- 이상치에 덜 민감하다.
- 모형의 가정 (선형성, 등분산성 등)이 필요 없다. 즉, 비모수적 모형이다.

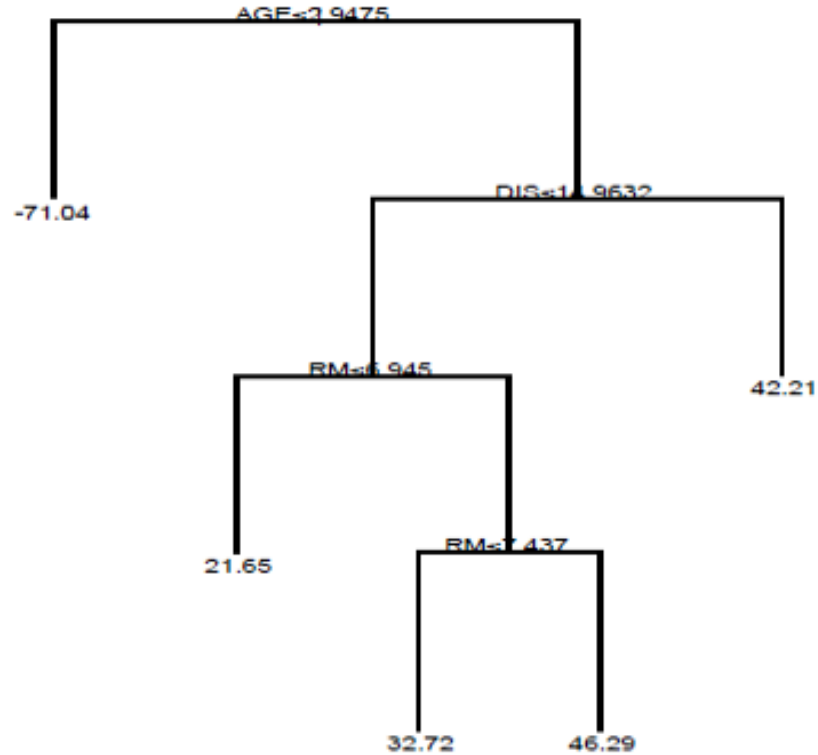
의사결정나무의 단점

- 목표변수가 연속형인 회귀모형에서는 그 예측력이 떨어진다.
- 나무가 너무 깊은 경우에는 예측력의 저하뿐 아니라 해석도 하기가 쉽지 않다.
- 계산량이 많을 수 있다.
- 비사각영역에서 문제가 있다.
- 결과가 불안정하다.
- 선형성 또는 주효과의 결여

의사결정나무의 단점의 예

- Boston Housing 자료로부터 부트스트랩을 이용하여 추출한 두 벌의 표본에 대해 노드 개수 5로 적합한 의사결정나무





앙상블법

앙상블법 소개

- 앙상블법이란 하나의 자료에 대해서 여러 개의 예측모형을 만든 후, 이를 결합하여 최종예측모형을 만드는 방법을 통칭한다.
- 앙상블법의 예
  - Bagging (Breiman, 1996)
  - Boosting (Freund and Schapire, 1997)
  - Random Forest (Breiman, 2004)
- 실증적으로 앙상블 방법이 의사결정나무 보다 훨씬 좋은 예측력을 갖는 것이 밝혀졌다.

Bagging

- **B**ootstrap **agg**regat**ing**
- $\mathcal{L}$:학습자료
- 알고리즘
  1. 원자료 $\mathcal{L}$로부터 복원추출하여 B개의 부트스트랩 표본 $\{\mathcal{L}^{(b)}, b = 1, \ldots, B\}$을 만든다.
  2. 각각의 부트스트랩 표본에 대해서 예측모형 $\{f(\mathbf{x}, \mathcal{L}^{(b)}), b = 1, \ldots, B\}$을 구축한다.
  3. y가 연속형변수이면 평균예측모형 $f_B(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B} f(\mathbf{x}, \mathcal{L}^{(b)})$를 사용하고
  4. y가 범주형이면 다수결(majority vote) 방법을 이용하여 다음과 같이 배깅예측모형을 만든다: $f_B(\mathbf{x}) = \text{argmax}_j \#\{b : f(\mathbf{x}, \mathcal{L}^{(b)}) = j\}$.

예측력 비교

- Bagging classification trees

Data set	#sample	#var	#class	$\bar{e}_S$	$\bar{e}_B$	Decrease
waveform	300	21	3	29.0	19.4	33%
heart	1395	16	2	10.0	5.3	47%
breast cancer	699	9	2	6.0	4.2	30%
ionosphere	351	34	2	11.2	8.6	23%
diabetes	1036	8	2	23.4	18.8	20%
glass	214	9	6	32.0	24.9	22%
soybean	307	35	19	14.5	10.6	27%

예측력 비교

- Bagging regression trees

Data set	$\bar{e}_S$	$\bar{e}_B$	Decrease
Boston Housing	19.1	11.7	39%
Ozone	23.1	18.0	22%
Friedman #1	11.4	6.2	46%
Friedman #2	30,800	21,700	30%
Friedman #3	.0403	.0249	38%

배깅의 원리 I

- D를 주어진 자료라 하고 X, Y를 미래의 관측치라 하자. $f(X, D)$를 D로 구축한 추정량의 X에서의 값, 즉 $\hat{Y}$이라 하자. 또한, 평균추정량을

$$f_A(X) = \mathbb{E}_D f(X, D)$$

와 같이 표기하자.

- 젠센의 부등식을 이용하면

$$\mathbb{E}_{(X,Y)}\mathbb{E}_D(Y - f(X, D))^2 \geq \mathbb{E}_{(X,Y)}(Y - f_A(X))^2$$

가 성립하는 것을 알 수 있다.

- 자료 D의 분포를 부트스트랩분포로 근사를 한다고 생각하면 f_A는 배깅추정량과 비슷하다. 따라서, 우변은 배깅추정량의 예측오차와 비슷하다. 좌변은 f라는 추정방법이 평균적으로 갖는 예측오차이다. 여기서 평균은 주어진 자료 D와 미래의 자료 (X, Y)에 대해 이루어 졌다.

배깅의 원리 II

- 이를 해석하면 배깅추정량의 예측오차는 주어진 자료로 한 번 구축한 추정량 $f(X, D)$ 보다 평균적으로 좋다고 해석할 수 있다.
- 예측오차의 차이는

$$\mathbb{E}_{(X,Y)}\mathbb{E}_D(Y - f(X, D))^2 - \mathbb{E}_{(X,Y)}(Y - f_A(X))^2 = \mathbb{E}_{(X,Y)}\mathrm{Var}_D f(X, D)$$

  라는 것이 알려져있다.
- 따라서 배깅에 의해서 예측력이 향상되는 예측모형은 분산은 크고 편이는 작아야 한다는 것을 알 수 있다.
- 즉, 일부러 과적합된 예측모형에 배깅을 적용하면 큰 효과를 볼 수 있다.
- 이 원리를 의사결정나무에 적용하면, 가지치기를 하지 않은 나무모형이 배깅하고 가장 잘 어울린다.

배깅의 원리 III

- 의사결정나무 구축에서 가장 시간이 많이 드는 부분이 가지치기인데, 그 이유는 모형선택(얼마나 많은 가지를 처낼 것인지를 결정)이 필요하기 때문이다.
- 즉, 배깅은 최적의 의사결정나무 구축보다 계산속도가 빠르다!

랜덤포레스트

- 랜덤포레스트는 여러 개의 의사결정나무를 무작위로 만든 후, 이를 결합하여 최종 예측모형을 만드는 방법이다.
- 배깅은 랜덤포레스트의 특수한 경우인데, 일반적인 랜덤포레스트는 배깅보다 더 많은 무작위성을 사용한다.
- 랜덤포레스트의 장점은
  - 예측력이 배깅보다 좋으며
  - 이상치에 강건하고
  - 계산속도가 상대적으로 빠르며
  - 사용하기 쉽다. (초보자도 쉽게 사용 가능)

알고리즘

- RF1: 나무를 성장시킬 때, 각 노드에서 변수를 임의로 뽑아서 사용한다.
- RF2: 각 노드에서 m개의 변수를 임의로 뽑고, 이 변수들 중 가장 불순도를 크게 감소시키는 변수로 나무를 성장시킨다.
- RF-L
  - L개의 변수를 임의로 뽑는다.
  - L개의 변수를 이용하여 F개의 선형결합을 임의로 만든다 (가중치를 임의로 정한다).
  - F개의 새로운 변수중 가장 불순도를 크게 감소시키는 변수로 나무를 성장시킨다.
- 가지치기는 사용하지 않고, 부트스트랩 표본을 사용한다.

- 모든 변수를 사용한다면 RF2는 배깅과 같다.

예측력 비교

Data Set	Single	Bagging	AdaBoost	RF1	RF1-L
waveform	29.0	19.4	18.2	17.2	16.1
breast cancer	6.0	5.3	3.2	2.9	2.9
ionosphere	11.2	8.6	5.9	7.1	5.7
diabetes	23.4	18.8	20.2	24.2	23.1
glass	32.0	24.9	22.0	20.6	23.5

이상치 강건성

- 5%의 자료를 임의로 뽑아서 속한 그룹을 임의로 바꾼다.
- 오차율의 증가량 (%)

Data Set	AdaBoost	RF1
breast cancer	43.2	1.8
ionosphere	27.7	3.8
diabetes	6.8	1.8
glass	1.6	0.4

랜덤포레스트 성능의 직관적 근거

- 주어진 변수 중 한 개의 변수가 매우 중요하고 나머지 변수들은 중간 정도의 중요성을 가진다고 하자. 배깅을 하면 모든 나무들이 매우 중요한 변수부터 분할을 하게될 것이다. 그러면 나무들이 서로 비슷해지고 나무를 이용한 추정치 사이에 상관계수들이 커질 것이다. 그런데 $m < p$개의 변수만 고려한다면 매우 중요한 변수로 분할을 시작하지 않을 확률이 $\frac{p-m}{p}$이나 될 것이다. 이는 부트스트랩 나무들 사이의 상관성을 줄여서 평균의 분산을 줄여준다.

$$\text{Var}\left(\frac{X_1 + X_2}{2}\right) = \frac{1}{4}\left[\text{Var}(X_1) + \text{Var}(X_2) + 2\text{Cov}(X_1, X_2)\right]$$

  - $\text{Cov}(X_1, X_2)$가 작아지면 $\text{Var}\left(\frac{X_1+X_2}{2}\right)$가 작아진다.

랜덤포레스트의 장점

- 조율모수가 전혀 없어서 초보자도 쉽게 사용할 수 있다.
- 계산면에서도 랜덤포레스트는 매우 효율적인데, 완벽한 분산처리가 가능하기 때문이다.

해석

- 의사결정나무의 선형결합을 어떻게 해석할 수 있을까?
- 입력변수의 상대적 중요도(relative importance)와 부분의존성 도표(partial dependency plot)를 이용

상대적 중요도 I

- 주어진 나무 T에 대해서 $s(T,k)$를 k번째 노드에서 어미노드와 자식노드들 사이의 불순도 측정치의 차이라 하자 (불순도의 차이가 클수록 중요한 변수임).
- $v(m,k)$는 m번째 나무의 k번째 노드에 사용된 변수라 하자.
- 변수 j의 상대적 중요도는 다음과 같이 구한다.

$$RI_j = \sum_{m=1}^{M} \sum_{k=1}^{|T_m|} s(T_m,k) I(v(m,k)=j).$$

```
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

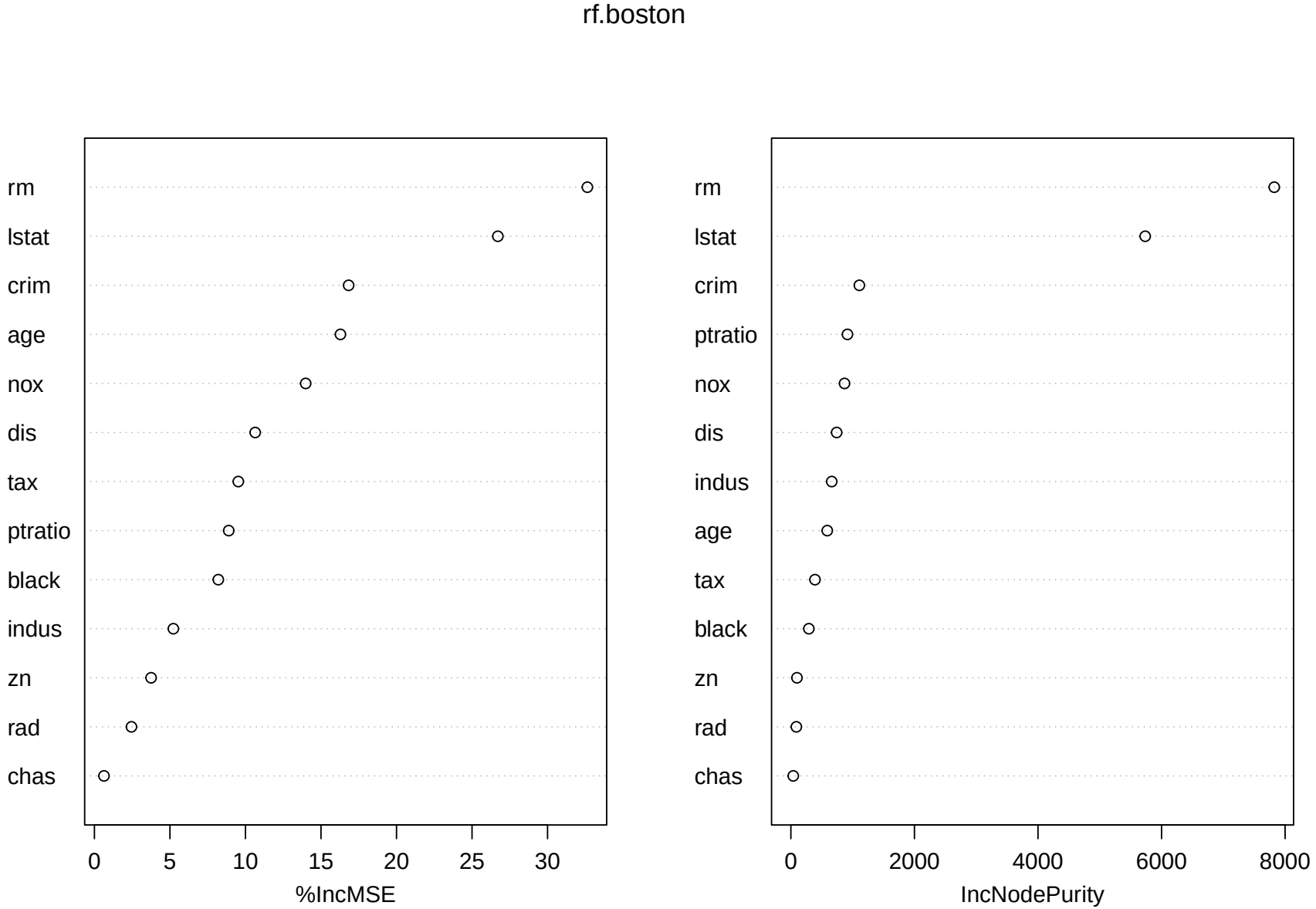
상대적 중요도 II

```
library(MASS)
set.seed(1)
train <- sample(1:nrow(Boston), nrow(Boston)/2)
boston.test <- Boston[-train,"medv"]
rf.boston <- randomForest(medv~., data=Boston, subset=train, mtry=6,
                          importance=TRUE)
yhat.rf <- predict(rf.boston, newdata=Boston[-train,])
mean((yhat.rf - boston.test)^2)  # prediction error
```

```
## [1] 19.43268
```

```
varImpPlot(rf.boston)
```

상대적 중요도 III


rf.boston
rm
lstat
crim
age
nox
dis
tax
ptratio
black
indus
zn
rad
chas
0
5
10
15
20
25
30
%IncMSE
rm
lstat
crim
ptratio
nox
dis
indus
age
tax
black
zn
rad
chas
0
2000
4000
6000
8000
IncNodePurity

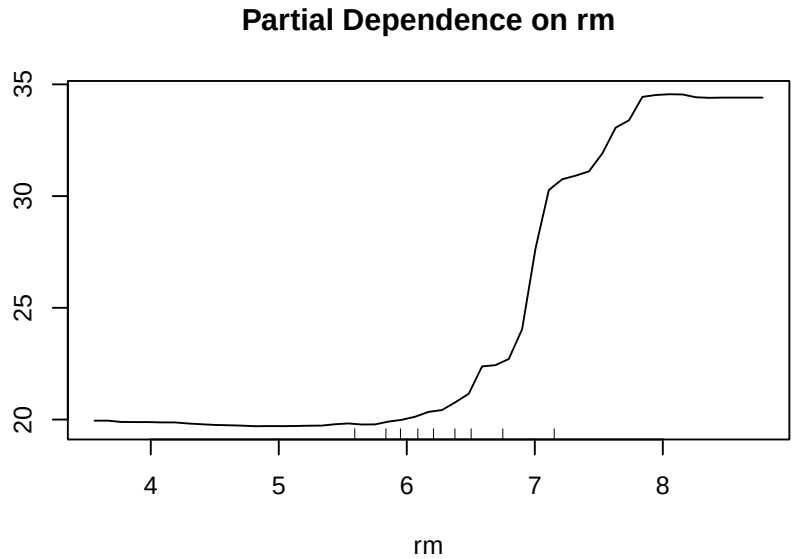
부분의존성 도표 I

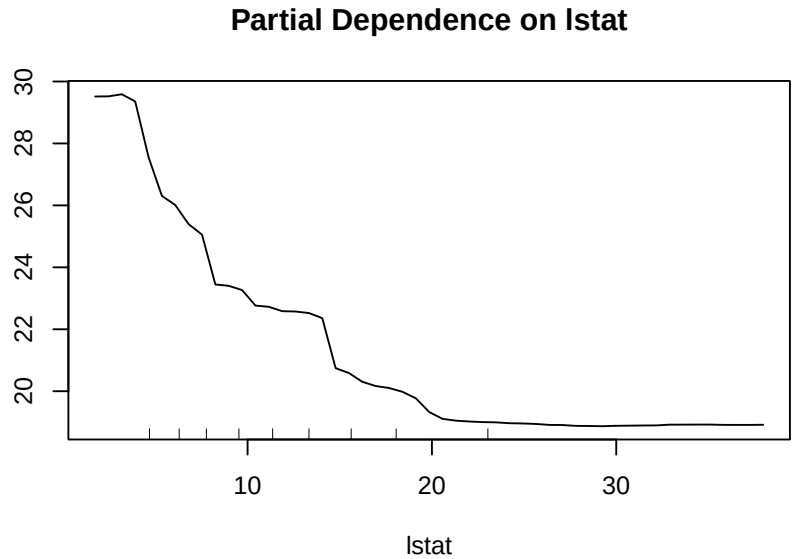
- 주어진 앙상블 모형 $F(\mathbf{x})$에서 j번째 변수의 부분의존성 도표 다음과 같이 정의된다.

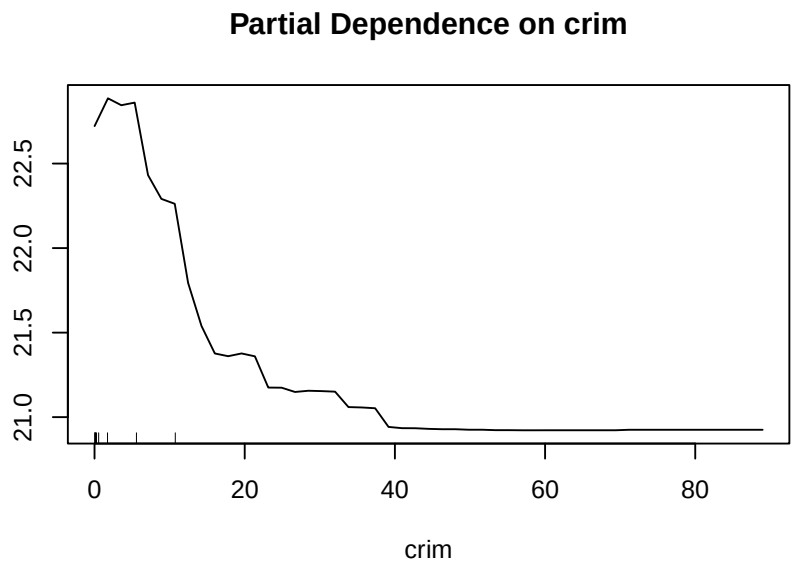
$$f_j(x_j) = \frac{1}{n}\sum_{i=1}^{n} F(x_{i1}, \cdots, x_{i(j-1)}, x_j, x_{i(j+1)}, \cdots, x_{ip}).$$

```
imp <- importance(rf.boston)
impvar <- rownames(imp)[order(imp[, 1], decreasing=TRUE)]
op <- par(mfrow=c(2, 2))
for (i in seq_along(impvar[1:4])) {
    partialPlot(rf.boston, Boston, impvar[i], xlab=impvar[i],
                main=paste("Partial Dependence on", impvar[i]))
}
```

부분의존성 도표 II


Partial Dependence on rm
35
30
25
20
4
5
6
7
8
rm


Partial Dependence on lstat
30
28
26
24
22
20
10
20
30
lstat


Partial Dependence on crim
22.5
22.0
21.5
21.0
0
20
40
60
80
crim


Partial Dependence on age
23.0
22.8
22.6
22.4
22.2
0
20
40
60
80
100
age

참고문헌

[1] Friedman, J., Hastie, T., & Tibshirani, R. (2008). “The Elements of Statistical Learning”, 2nd Ed. Springer, New York.

[2] 박창이 외 4인 (2011). “R을 이용한 데이터마이닝”. 교우사.

[3] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). “An Introduction to Statistical Learning.” Springer, New York.