데이터마이닝 (**D**ata **M**ining)

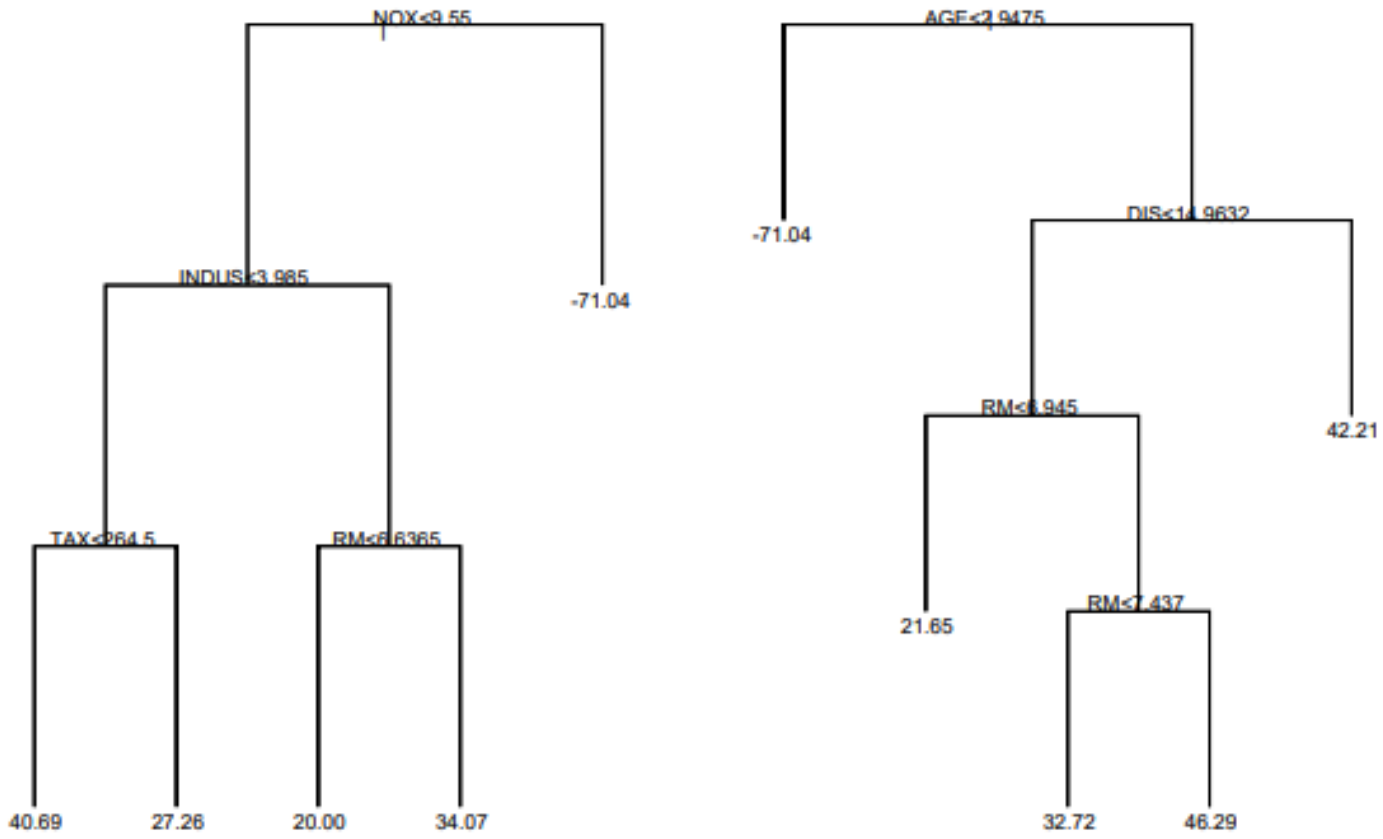
Chapter 1. 앙상블(emsemble) 기법

학습방법의 불안정성

- 학습자료의 작은 변화에 의해 예측모형이 크게 변하는 경우
  - 안정 : 1-차 근방, 최소제곱추정에 의한 선형회귀
  - 불안정 : 선형회귀에서의 최적 부분집합 선택, 의사결정나무 등 예측모형 구축 시 불연속적인 의사결정

- 학습방법의 불안정성은 예측력을 저하시키며 예측모형의 해석을 어렵게 함

학습방법의 불안정성


NOX<9.55
INDUS<3.965
-71.04
TAX<264.5
RM<6.6365
40.69
27.26
20.00
34.07
AGE<2.9475
-71.04
DIS<14.9632
RM<6.945
42.21
21.65
RM<7.437
32.72
46.29

배깅 알고리즘

- Bagging : **B**oostrap **Agg**regat**ing**
- 주어진 자료에 대하여 여러 개의 부스트랩(bootstrap) 자료를 생성하고 각 부스트랩 자료에서 예측모형을 만든 후 결합하여 최종 예측모형을 만듬.
- $\mathcal{L} = \{(x_i, y_i)\}_{i=1}^{n}$: 훈련자료
- 알고리즘
  1. B 개의 부스트랩 자료 $\mathcal{L}^{*(b)},\ b = 1, \cdots, B$를 만든다.
  2. 각 부스트랩 자료 $\mathcal{L}^{*(b)}$에 대해서 예측모형 $f^{(b)}(x)$를 구축한다.
  3. B개의 예측모형을 결합하여 최종 모형 $\hat{f}$을 만든다. 최종모형을 만드는 방법은
     a. 회귀모형인 경우 $\hat{f} = \sum_{b=1}^{B} f^{(b)}(x)/B$ 와 같이 평균을 취한다.
     b. 분류모형인 경우 $\hat{f} = \arg max_k \left(\sum_{b=1}^{B} I(f^{(b)}(x) = k)\right)$와 같이 투표(voting)를 한다.

배깅 알고리즘

- 배깅 알고리즘은 매우 단순하지만 불안정한 학습방법의 예측력을 획기적으로 향상시킴

Breiman (1996) 논문의 예시 중 일부

자료	표본수	변수개수	클래스 개수	$\bar{e}_S$	$\bar{e}_B$	오차 감소율
waveform	300	21	3	29.1	19.3	34 %
heart	1395	16	2	4.9	2.8	43 %
breast cancer	699	9	2	5.9	3.7	37%
ionosphere	351	34	2	11.2	7.9	29%
diabets	768	8	2	25.3	23.9	6%
glass	214	9	6	30.4	23.6	22%
soybean	683	35	19	8.6	6.8	21%

$\bar{e}_S$: 최적의 단일 의사결정나무의 예측오차, $\bar{e}_B$: 배깅의 예측오차

배깅 알고리즘

- 최적의 의사결정나무 구축시 가장 어려운 부분은 가지치기
- 배깅에서는 가지치기를 하지 않은 최대로 성장한 의사결정나무를 사용
- 가지치기를 하지 않음으로써 오히려 하나의 최적 의사결정나무를 구축하는 것보다 계산량이 작을 수 있음

배깅 알고리즘

- 훈련자료 $\mathcal{L}$을 이용하여 구축된 예측모형 $\hat{f}(x) = f(x, \mathcal{L})$
- 예측모형 $f(x, \mathcal{L})$에 대한 평균예측모형 $f_A(x) = E_{\mathcal{L}} f(x, \mathcal{L})$
- **정리**

  $(X,\ Y)$를 $\mathcal{L}$과 독립인 미래의 관측값이라 하자. 제곱손실함수 $L(y,\ a) = (y-a)^2$에 대하여$\hat{f}(x)$과 $f_A(x)$의 기대손실 R와 R_A를 다음과 같이 정의한다.

  $$R = E_{(X,Y)} E_{\mathcal{L}} L\big(Y,\ f(X, \mathcal{L})\big), \qquad R_A = E_{(X,Y)} L\big(Y,\ f(X, \mathcal{L})\big)$$

  그러면 항상 $R \geq R_A$

배깅 알고리즘

- 배깅은 주어진 예측모형의 평균예측모형 구하는 것
- 배깅은 예측모형의 편의(bias)에는 영향을 미치지 않고 분산에만 영향을 미침
- 배깅을 적용하기에 적합한 예측모형은 편의가 없고 분산이 큰 모형
- 일반적으로 과대적합된 모형이 편의가 작고 분산이 크며, 의사결정나무에 배깅을 적용할 때 가지치기를 하지 않는 이유는 과대적합을 통해 배깅의 효과를 극대화하기 위함

부스팅

- 예측력이 약한 예측모형(weak learner)들을 결합하여 강한 예측모형을 만듬
- 예) 경마에서 우승하는 말을 맞추기 위한 전략

부스팅

- AdaBoost(Adaptive Boost) 알고리즘

1. 가중치 $w_i = \frac{1}{n},\ i = 1,\ \cdots,\ n$를 초기화
2. $m = 1,\ \cdots,\ M$에 대하여 다음의 과정을 반복
   a. 가중치 w_i를 이용하여 분류기 $f_m(x) \in \{-1,\ 1\}$를 적합한다.
   b. err_m를 다음과 같이 계산한다.
   c. $err_m = \frac{\sum_{i=1}^{n} w_i I(y_i \neq f_m(x_i))}{\sum_{i=1}^{n} w_i}$
   d. $c_m = \log((1 - err_m)/err_m)$로 설정한다.
   e. 가중치 w_i를 $w_i = w_i \exp(c_m\ I(y_i \neq f_m(x_i)))$로 업데이트한다.
3. 단계 2에서 얻은 M개의 분류기를 결합하여 최종 분류기 $sign(\sum_{m=1}^{M} c_m f_m(x))$를 얻음

부스팅

- 단계 2의 c, d
  - $\widehat{f_m}$: 랜덤한 추측보다 조금 좋은 예측력을 갖는다고 하면 f_m의 오분류율은 0.5보다 작게 되므로 c의 $c_m > 0$
  - d에서 각 관측치에 할당되는 가중치가 $\widehat{f_m}$에 의해서 오분류된 관측치에서는 증가하고 정분류된 관측치는 기존의 값과 같음

    >> 매 반복마다 오분류된 관측치의 가중치는 증가시키고 정분류된 가중치는 감소시키면서 예측모형을 만듬

부스팅

- 약한 학습기 $\widehat{f_m}$의 오분류율이 항상 $0.5 - \gamma\ (\gamma > 0)$ 이면 훈련오차는 지수적으로 빠르게 0으로 수렴
- AdaBoost 알고리즘의 본래 목적은 훈련오차를 빨리 줄이기 위한 것이나 예측오차도 감소한다는 것이 경험적으로 입증

부스팅의 예측오차

자료	단일 의사결정나무	Adaboost	감소율	Bagging
waveform	29.0	18.2	37 %	19.4
breast cancer	6.0	3.2	47 %	5.3
ionosphere	11.2	5.9	47 %	8.6
diabets	23.4	20.2	14 %	18.8
glass	32.0	22.0	31 %	24.9