



부산대학교

PUSAN NATIONAL UNIVERSITY

Pruning and Regression Trees

Knowing When to Stop

Growing a Tree, Then Cutting It Back

BX3600087-140 / DS3600087-140 / AB3600087-140

기계학습개론

2026학년도 2학기

오용경

데이터사이언스전공

yongkyung.oh@pusan.ac.kr

Section 1 • Pruning and Regression Trees

Regression Trees

같은 구조로 수치를 예측한다

회귀 트리는 수치를 예측한다

DEFINITION

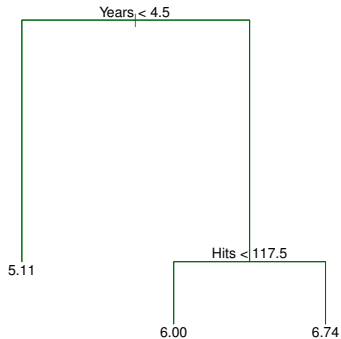
회귀 트리(regression tree)

특성 공간을 여러 구역으로 나누고, 각 구역의 학습 관측값 평균으로 수치형 반응변수를 예측하는 모델

제곱오차를 사용하는 회귀 트리의 기본형

Source: James et al. (2023), Section 8.1.1.

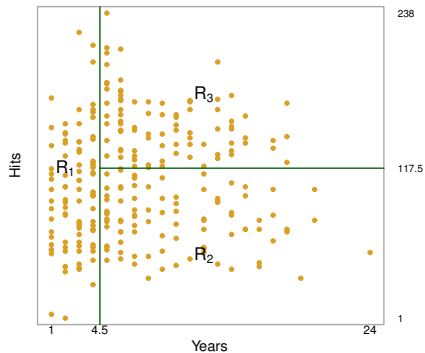
나무 하나를 읽어 보기



경력과 안타 수로 분할하며 잎(끝점, terminal node)의 5.11, 6.00, 6.74는 연봉의 자연로그 예측값

Source: James et al. (2023), Figure 8.1.

나무의 분할을 특성 공간에서 읽는다



두 번의 분할이 평면을 세 구역으로 나누며 각 구역이 나무의 잎 하나에 대응

Source: James et al. (2023), Figure 8.2.

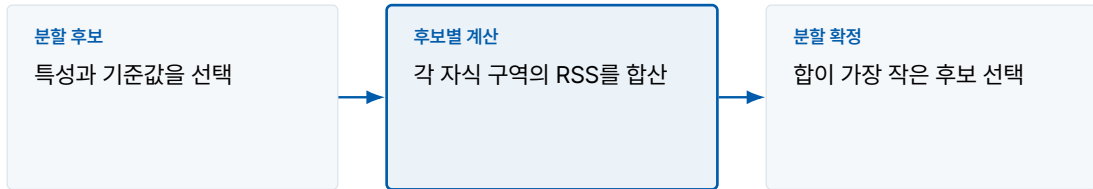
제곱오차를 최소화하는 앞의 예측값

$$\hat{y}_{R_m} = \frac{1}{n_m} \sum_{i: x_i \in R_m} y_i, \quad n_m = \#\{i: x_i \in R_m\}$$

R_m 은 앞에 해당하는 구역, n_m 은 그 구역의 학습 관측 수

같은 앞에 도달한 관측에는 같은 평균값을 예측한다

분할 후 잔차제곱합이 가장 작은 후보를 고른다



Caution 현재 분할의 최적 선택이 전체 나무의 최적 구조를 보장하지는 않는다

서로 다른 관측값 사이에서 기준값을 찾는다

기준값	왼쪽의 y	오른쪽의 y	분할 후 RSS
3.5	1	2, 8, 9	$86/3$
7	1, 2	8, 9	1
11.5	1, 2, 8	9	$86/3$

예제: $x = (2, 5, 9, 14)$, $y = (1, 2, 8, 9)$ 이며 각 구역에서 평균을 예측

범주형 특성의 분할은 구현을 확인한다

관점	회귀 트리에서의 처리
RSS 기준의 범주 분할	현재 학습 구역에서 범주별 반응값 평균을 구하고 그 순서로 분할 후보를 탐색할 수 있다
scikit-learn의 입력	문자 범주를 그대로 받지 않으므로 수치 부호화가 필요하다
데이터 누수 방지	반응값을 이용한 변환은 학습 자료 안에서만 추정한다

Source: James et al. (2023), Section 8.1 · scikit-learn, Decision Trees.

Checkpoint

Regression Trees

- 앞에서 분류 트리와 회귀 트리는 각각 무엇을 내놓는가
- 구역 안의 예측값은 어떻게 정해지는가
- 분할 후보를 비교할 때 무엇을 최소화하는가

Why Prune

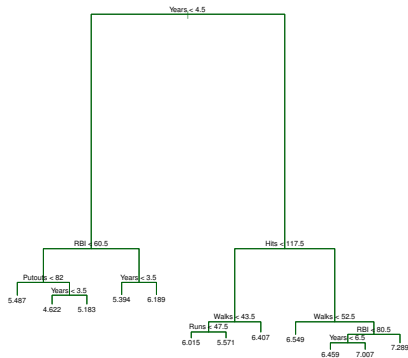
복잡한 나무의 과적합을 설명한다

분할을 늘려도 항상 학습 오차가 0이 되는 것은 아니다

학습 자료의 조건	가능한 결과
관측마다 다른 특성값	관측을 하나씩 분리하면 학습 오차가 0이 될 수 있다
같은 특성값, 다른 반응값	분할만으로 서로 구별할 수 없어 오차가 남는다
새 관측의 예측	학습 오차가 작아도 테스트 오차는 클 수 있다

Caution 학습 오차만으로 일반화 성능을 판단할 수 없다

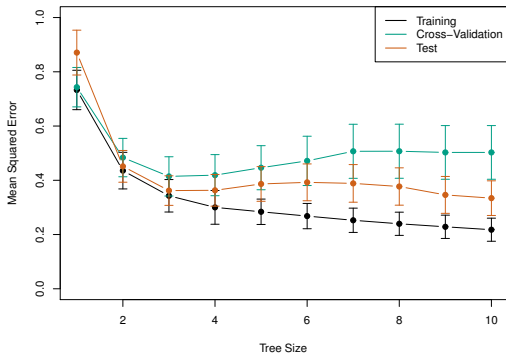
끝까지 자란 나무



가지치기 전의 회귀 트리로, 분할이 많아 예측 경로를 읽기 어렵다

Source: James et al. (2023), Figure 8.4.

앞을 늘리면 두 오차가 갈린다



이 예제에서는 앞이 3개일 때 교차검증 오차가 최소이며, 오차 막대는 평균 오차의 추정이 흔들리는 정도인 표준오차

Source: James et al. (2023), Figure 8.5.

작은 나무의 두 가지 이점

관점	가지치기로 기대하는 변화
예측의 변동	학습 자료의 세부 차이에 덜 민감해질 수 있다
해석의 용이성	분기 수가 적어 예측 규칙을 따라가기 쉽다

너무 작은 나무는 중요한 구조도 놓칠 수 있으므로 검증 성능을 함께 본다

조기 중단과 사후 가지치기의 차이

방법	결정 시점	주의점
조기 중단	나무를 키우는 중	작은 개선 뒤에 올 큰 개선을 놓칠 수 있다
사후 가지치기	큰 나무를 학습한 뒤	큰 나무와 가지치기 경로를 계산해야 한다

Checkpoint

Why Prune

- 학습 오차가 0이어도 과적합을 의심할 수 있는 이유는 무엇인가
- 앞을 늘릴 때 두 오차는 각각 어떻게 움직이는가
- 미리 멈추는 방법의 문제는 무엇인가

Cost Complexity Pruning

얼마나 자를지를 하나의 값으로 조절한다

오차와 잎 수를 함께 최소화한다

$$C_{\alpha}(T) = \underbrace{\sum_{m=1}^{|T|} \sum_{i: x_i \in R_m} (y_i - \hat{y}_{R_m})^2}_{\text{RSS}(T)} + \underbrace{\alpha |T|}_{\text{복잡도 벌점}}, \quad \alpha \geq 0$$

$|T|$ 는 잎 수이며, T 는 이미 학습한 큰 나무의 부분나무 중에서 고른다

여러 잎을 함께 줄일 수 있으므로 RSS 증가량과 줄어드는 잎 수를 함께 비교한다

같은 α 에서 세 부분나무를 비교한다

앞 수	RSS	C_0	C_3	C_9
4	2	2	14	38
2	6	6	12	24
1	14	14	17	23

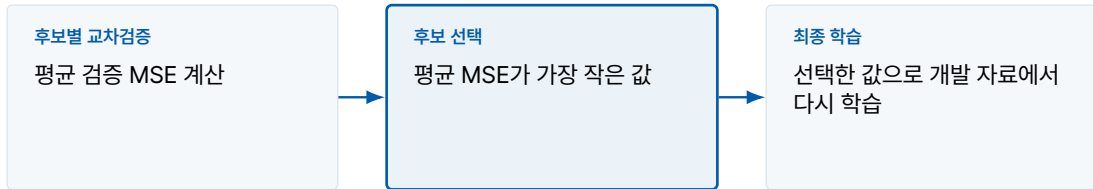
계산용 예제: 이 세 부분나무가 후보라고 가정하면 $\alpha = 0, 3, 9$ 에서 각각 앞 4, 2, 1개를 선택

α 가 변해도 같은 나무가 선택되는 구간이 있다

α 구간	선택되는 잎 수	목적함수
$0 \leq \alpha < 2$	4	$2 + 4\alpha$
$2 < \alpha < 8$	2	$6 + 2\alpha$
$\alpha > 8$	1	$14 + \alpha$

앞 예제의 경계 $\alpha = 2, 8$ 에서는 인접 후보가 동률이며, 잎은 한 번에 여러 개 줄어들 수 있다

교차검증으로 α 를 선택한다



Caution 테스트 자료는 α 선택에 사용하지 않고 마지막 평가에만 사용한다

학습 오차와 교차검증 오차는 역할이 다르다

관측한 결과	해석
잎 수를 늘릴수록 학습 RSS 감소	잎마다 따로 평균을 두므로 학습 자료에 더 잘 맞는다
잎 3개에서 교차검증 오차 최소	교차검증 오차는 더 큰 나무에서 더 줄지 않았다
다른 자료에서 최적 잎 수 변화	잎 3개가 모든 문제의 정답이라는 뜻은 아니다

나무의 복잡도에도 벌점을 부과한다

모델	적합 오차	복잡도 벌점
ridge regression	잔차제곱합	$\lambda \sum_j \beta_j^2$
회귀 트리	잔차제곱합	$\alpha T $

ridge regression은 절편을 제외한 계수 크기, 회귀 트리는 잎 수에 벌점

비슷한 검증 성능이면 복잡도도 함께 본다

앞 수	평균 검증 MSE	표준오차
3	0.42	0.06
6	0.41	0.07
10	0.50	0.08

설명용 예제: 앞 6개의 평균 오차가 가장 작지만 앞 3개와의 차이는 0.01

Caution 표준오차보다 차이가 작다는 사실만으로 두 모델의 성능이 같다고 입증되지는 않는다

Checkpoint

Cost Complexity Pruning

- 식의 두 항은 각각 무엇을 재는가
- α 를 아주 크게 하면 어떤 나무가 남는가
- α 는 무엇으로 고르는가

Section 4 • Pruning and Regression Trees

Reading a Tree

규칙과 중요도로 나무를 읽는다

각 경로는 조건과 예측값으로 읽는다

경로의 조건	예측하는 로그 연봉
경력 < 4.5년	5.11
경력 ≥ 4.5년, 안타 수 < 117.5	6.00
경력 ≥ 4.5년, 안타 수 ≥ 117.5	6.74

로그 연봉: 천 달러 단위 연봉(Salary)에 자연로그를 취한 값

Source: James et al. (2023), Figure 8.1.

특성 중요도는 가중 불순도 감소량을 합산한다

앞의 네 관측 예에서 첫 분할은 $d = (50 - 1)/4 = 12.25$ 이고, 이 특성만 쓰면 중요도는 1이다

$$d_t = \frac{n_t}{N} \left[l_t - \frac{n_L}{n_t} l_L - \frac{n_R}{n_t} l_R \right], \quad v_j = \frac{\sum_{t: \text{feature}(t)=j} d_t}{\sum_t d_t}$$

t 는 분기점, d_t 는 그 분기점의 가중 불순도 감소량, v_j 는 특성 j 의 중요도다. 회귀에서 불순도 l 는 구역 내 MSE이고 $(n_t/N) \cdot \text{MSE}$ 를 더하면 RSS/N 이므로 RSS 기준 분할과 같다. N 은 전체 학습 관측 수

scikit-learn은 전체 중요도의 합을 1로 맞추며, 분할이 없으면 모두 0이다

특성 중요도는 인과관계를 설명하지 않는다

잘못된 해석	가능한 해석
중요도가 크므로 결과의 원인이다	학습한 나무에서 가중 불순도를 많이 줄였다
중요도가 낮으므로 쓸모가 없다	비슷한 정보를 가진 다른 특성이 먼저 선택되었을 수 있다
이 순위는 항상 유지된다	학습 자료나 설정에 따라 순위가 달라질 수 있다

Caution 중요도만으로 특성을 바꿨을 때 결과가 어떻게 변할지는 알 수 없다

비슷한 특성의 중요도는 분산되거나 한쪽에 집중될 수 있다

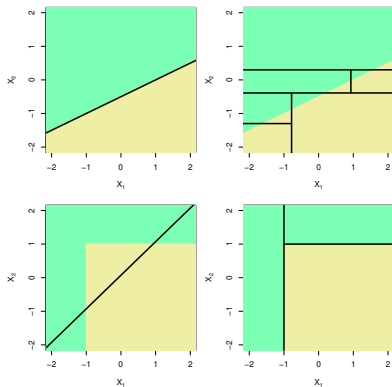
분할에서의 선택	중요도에 나타날 수 있는 결과
두 특성이 번갈아 선택됨	불순도 감소량이 두 특성에 분산된다
한 특성이 주로 선택됨	다른 특성의 중요도가 낮아진다
학습 자료가 조금 달라짐	선택되는 특성과 중요도 순위가 바뀔 수 있다

의사결정나무의 장점에는 적용 조건이 있다

관점	장점	주의점
해석	작은 나무는 경로를 읽기 쉽다	큰 나무는 규칙이 복잡해진다
단위	보통 표준화 없이 분할할 수 있다	수치 입력과 부호화는 필요하다
결측·범주	이를 처리하는 나무 구현이 있다	지원 여부는 구현마다 다르다
표현력·안정성	비선형 규칙을 학습할 수 있다	자료 변화에 구조가 민감하고, 단일 나무의 예측 정확도는 다른 방법보다 낮은 경우가 많다

Source: James et al. (2023), Section 8.1.4 · scikit-learn, Decision Trees.

비스듬한 경계에는 더 많은 분할이 필요하다



경계의 형태와 모델

- 위: 비스듬한 경계는 선형 모델이 간단히 표현한다. 나무는 여러 분할로 근사한다.
- 아래: 축에 나란한 경계는 나무가 적은 분할로 표현한다.

Source: James et al. (2023), Figure 8.7.

회귀 트리의 핵심 용어

용어	의미
회귀 트리	각 잎의 학습 관측값 평균으로 수치를 예측하는 나무
가지치기	학습한 큰 나무의 부분나무를 선택하는 절차
α	잎 수에 부과하는 벌점의 계수
특성 중요도	특성별 가중 불순도 감소량을 합산해 상대 크기로 나타낸 값

학습 자료에서 가지치기 경로를 계산한다

```
1 from sklearn.tree import DecisionTreeRegressor
2 tree = DecisionTreeRegressor(random_state=0)
3 path = tree.cost_complexity_pruning_path(X_dev, y_dev)
4 alphas = path.ccp_alphas
5 impurities = path.impurities
```

X_{dev} , y_{dev} 는 테스트 자료를 제외한 개발 자료

Caution scikit-learn의 벌점은 평균 불순도 단위이므로 앞의 RSS 식의 α 를 N 으로 나눈 값이다

교차검증으로 선택한 나무를 테스트 자료에 적용한다

```
1 from sklearn.model_selection import GridSearchCV
2 search = GridSearchCV(
3     DecisionTreeRegressor(random_state=0),
4     {"ccp_alpha": alphas},
5     scoring="neg_mean_squared_error", cv=5)
6 search.fit(X_dev, y_dev)
7 y_pred = search.best_estimator_.predict(X_test)
```

앞 쪽의 가지치기 경로에서 얻은 α 들을 후보로 쓴다. 개발 자료에서 선택·재학습한 뒤 테스트 자료에 적용한다

음의 MSE 점수를 최대화하는 것은 MSE를 최소화하는 것과 같다

학습으로 추정하는 값과 검증으로 선택하는 값

대상	결정 방법	사용 자료
분할 특성과 기준값	불순도 감소량을 기준으로 학습	학습 자료
앞의 예측값	앞 안의 관측값 평균	학습 자료
α	교차검증 오차를 비교해 선택	개발 자료의 교차검증 분할

복잡도 벌점만으로 테스트 성능을 알 수는 없다

단계	판단하는 내용
학습	주어진 복잡도 설정에서 예측 규칙을 추정한다
검증	후보 설정의 일반화 성능을 비교한다
테스트	선택을 마친 모델의 성능을 별도 자료에서 평가한다

Checkpoint

Reading a Tree

- 잎이 셋인 나무에서 규칙은 몇 개인가
- 서로 비슷한 특성이 있으면 중요도를 어떻게 해석해야 하는가
- 가지치기 경로의 α 가운데 무엇을 교차검증으로 고르는가

Can Class Distributions Tell Us the Class?

Next Lecture: Naive Bayes

- 베이즈 정리와 사전확률
- 범주별 조건부 분포
- 사후확률로 범주를 선택하는 법