데이터 마이닝 모델
의사결정트리

History

2019.09.13 내용 전면 추가, 연속형일때 분할기준 추가

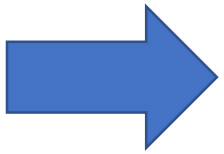
목차

01 데이터 마이닝 기법의 구분

▶ 대표적인 지도학습(supervised learning)

(가) 회귀(예측) - Regression

(나) 분류(Classification)



공통점 : 입력 및 **목표 변수의 값**을 이용하여
주어진 **입력변수에 대한 목표변수의 값을 예측**하는
모형을 개발한다.

01 데이터 마이닝 기법의 구분

▶ 대표적인 지도학습(supervised learning)

(가) 회귀(예측) - Regression

(나) 분류(Classification)

차이점 :

A. 목표 변수의 형태가 회귀의 경우 연속형이다.

B. 분류의 경우는 범주형이다.

02 지도학습, 자율학습, 비정형 분석

지도학습

분류분석

- 판별분석
- 로지스틱 회귀분류
- knn - 최근접이웃기법
- 의사결정나무
- 나이브베이즈분류
- 신경망
- SVM(지지도벡터기계)

회귀분석

- 회귀분석
- knn – 최근접이웃기법
- 신경망
- 평활법

비지도학습

군집 분석

- k-mean(K 평균)
- 계층적 군집분석
- 유한혼합모형
- 이중군집법

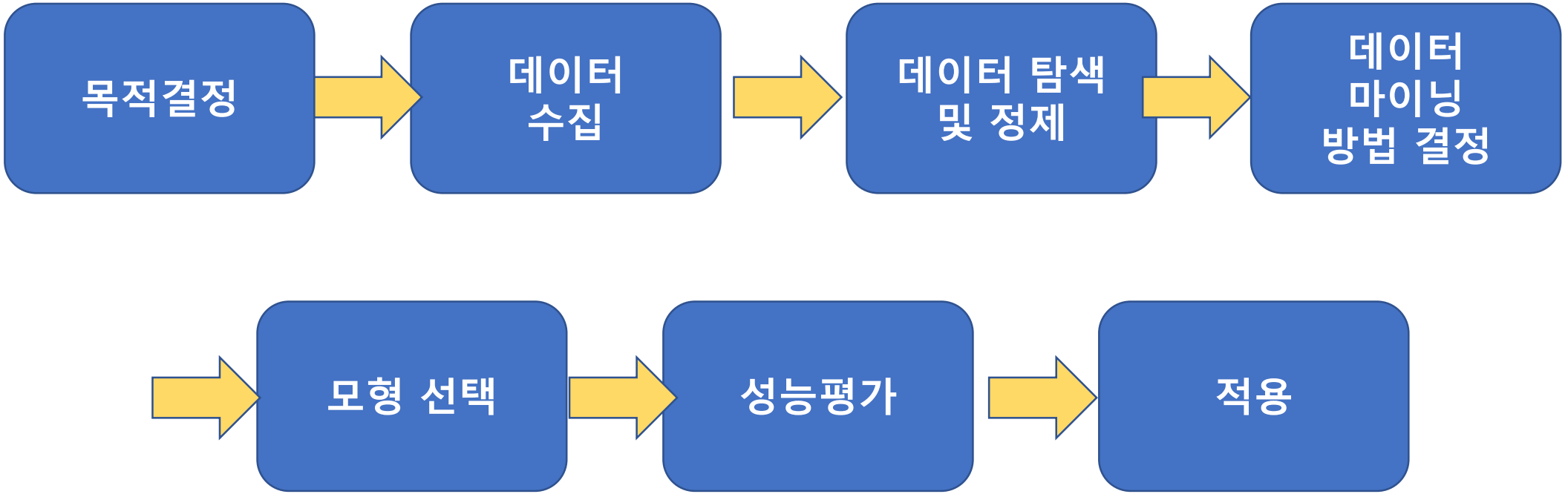
연관성 분석

- 장바구니 분석
- 서열 분석
- 트랜잭션 데이터분석

가중치 결정

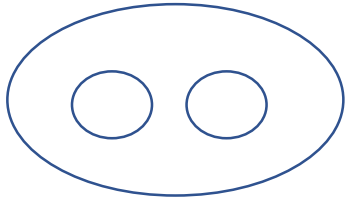
- 신경망

03 데이터 마이닝의 수행단계



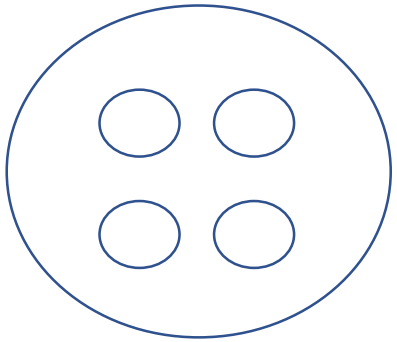
04 분류(Classification)

분류
모델



이항분류

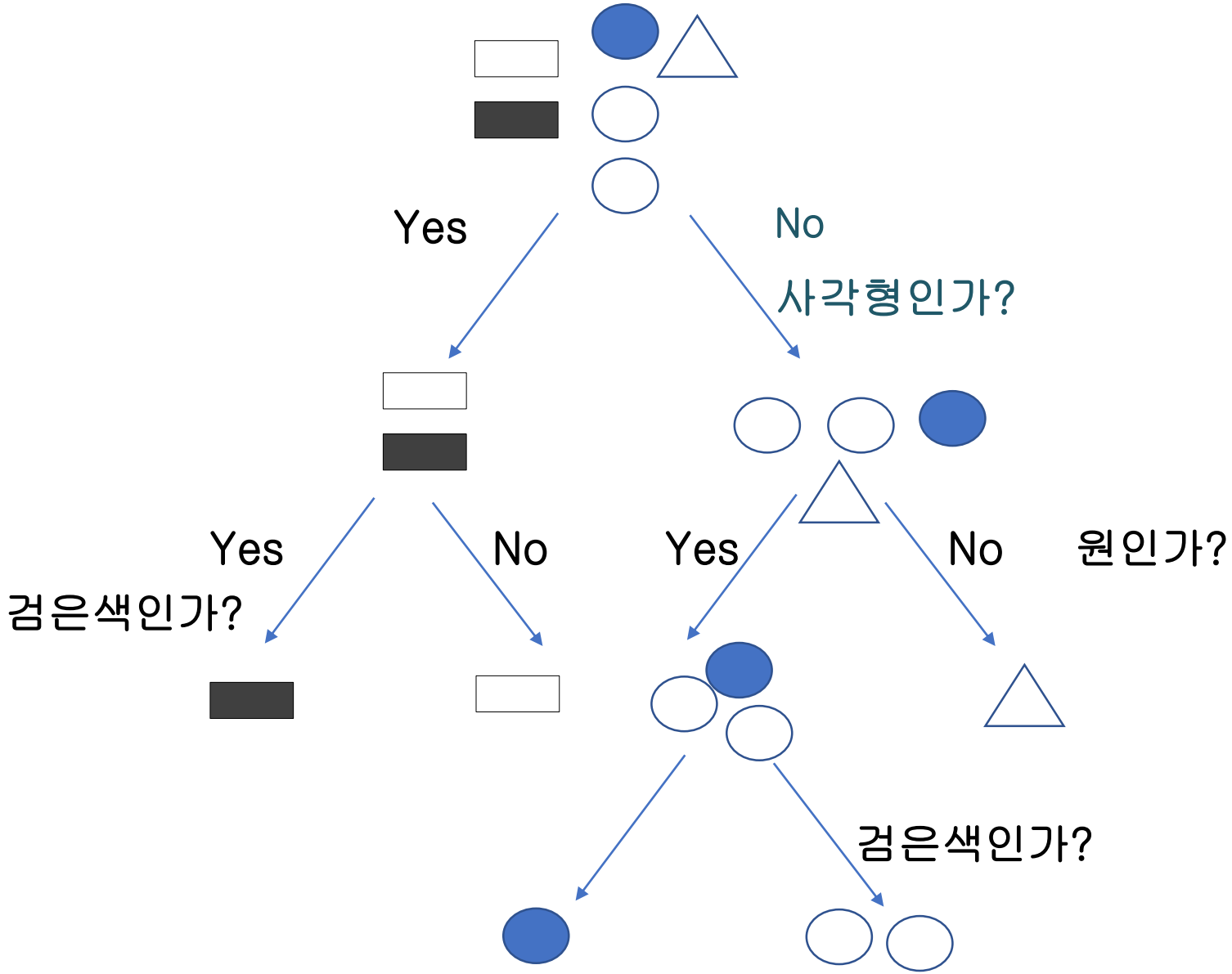
목표 변수의 값의 범주가 2개



다항분류

목표 변수의 값의 범주가 2개 이상

05 의사결정트리



06 Decision Tree 종류

나무모델

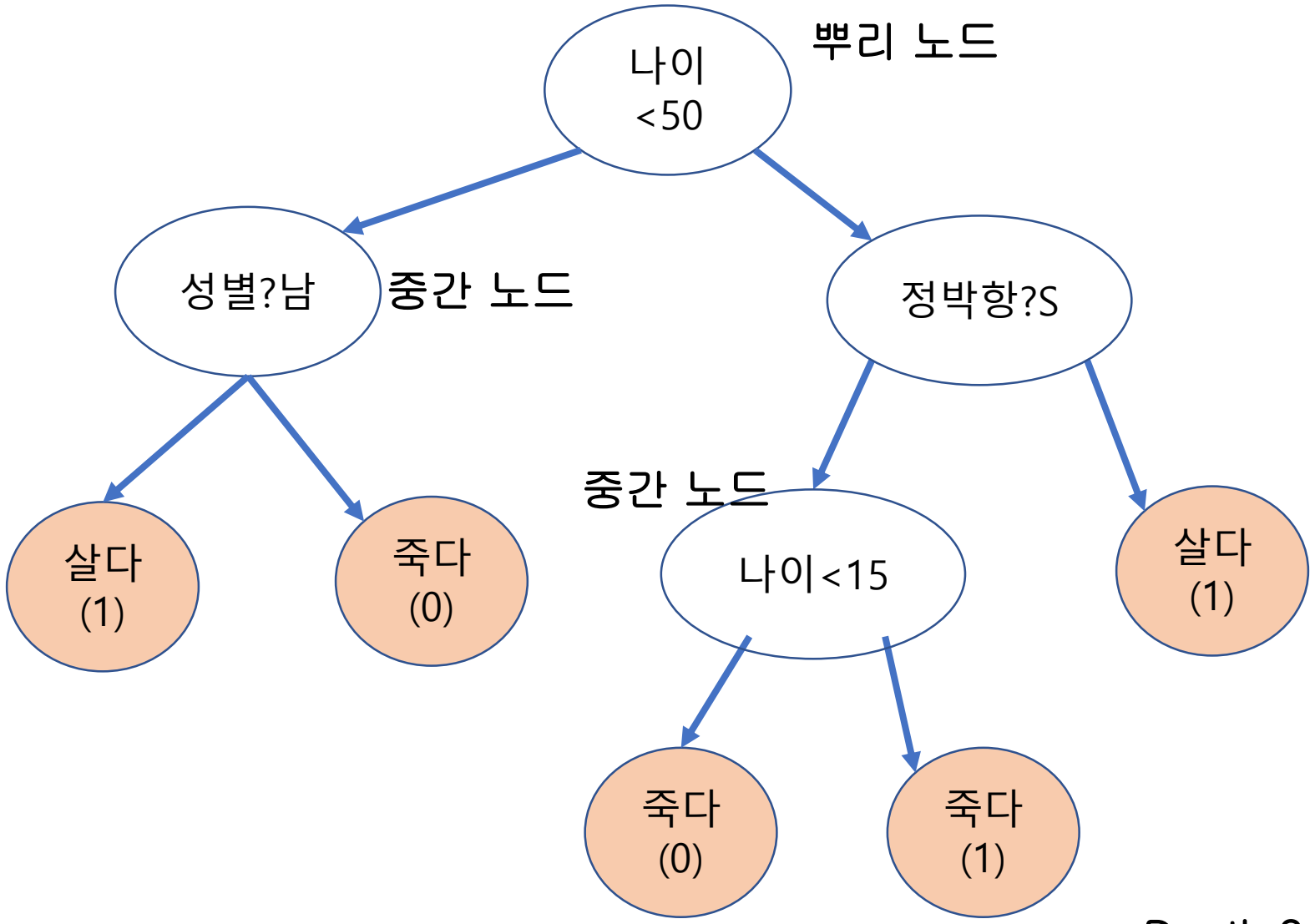
목표변수가 범주형

분류(Classification) 나무

목표변수가 숫자형

회귀(Regression) 나무

07 Decision Tree 구성



변수 선택

– 나무모형에서 뿌리 노드 및 중간 노드의 분할에 사용되는 변수는 매우 유용한 변수로 판단할 수 있다.

깊이(depth)

노드가 분기하여 생성하는 깊이

가지

노드와 노드를 연결한다.

Leaf(잎), Terminal node

맨 끝단에 있는 노드를 말한다.

08 의사결정트리 역사

AID
(automatic interaction detection)
선퀴스트와 모건(1964)

THAID
Morgan and Messenger(1973)

CHAID
(chi-squared a.i.d)
카스(Kass, 1980)

CART
(classification and regression trees)
나무모형(1984)

비모수적 방법

CART
나무모형(1984)

C4.5의 나무모형

기타

FACT(1988) – Quest(1997) – Cruise(2001)

09 의사결정 트리 장단점

장 점	단 점
입력변수의 형태에 관계없이 적용이 가능 명목형, 순서형, 숫자형 여부에 영향을 받지 않음.	연속형 변수인 경우, 분리점 경계에 있는 값은 잘못 예측될 가능성이 커진다.
이해와 해석이 용이하다.	나무구조는 불안정적 첫번째 변수가 어떤 것이 선택되는 것에 따라 나무 구조의 변형이 많이 일어난다. 관측치가 적을 때는 결과가 안정적인 결과가 안나옴.
상호작용을 쉽게 찾아낼 수 있다.	
결측치의 처리가 용이하다. 분할변수의 결측치가 있는 경우, 서로게이트(surrogate)라는 대리 변수를 사용하여 처리	
외부 데이터에 대한 분류 및 예측이 쉽게 이루어진다.	

10 의사 결정 트리 분할 방법(분류)

- 자식노드의 순수도가 가장 높은지는 분할 규칙을 채택하는 것이 나무모형의 기본적 아이디어

$$\text{지니지수(t)} = 1 - \sum_{J=1}^{J} p(j|t)^2$$

$$\text{분할 개선도} = \text{지니지수(t)} - \frac{N_{t_1}}{N_t} * \text{지니지수}(t_1) - \frac{N_{t_2}}{N_t} * \text{지니지수}(t_2)$$

10 분류(Classification)나무모형의 분할 방법

$$\text{지니지수(t)} = 1 - \sum_{J=1}^{J} p(j|t)^2$$

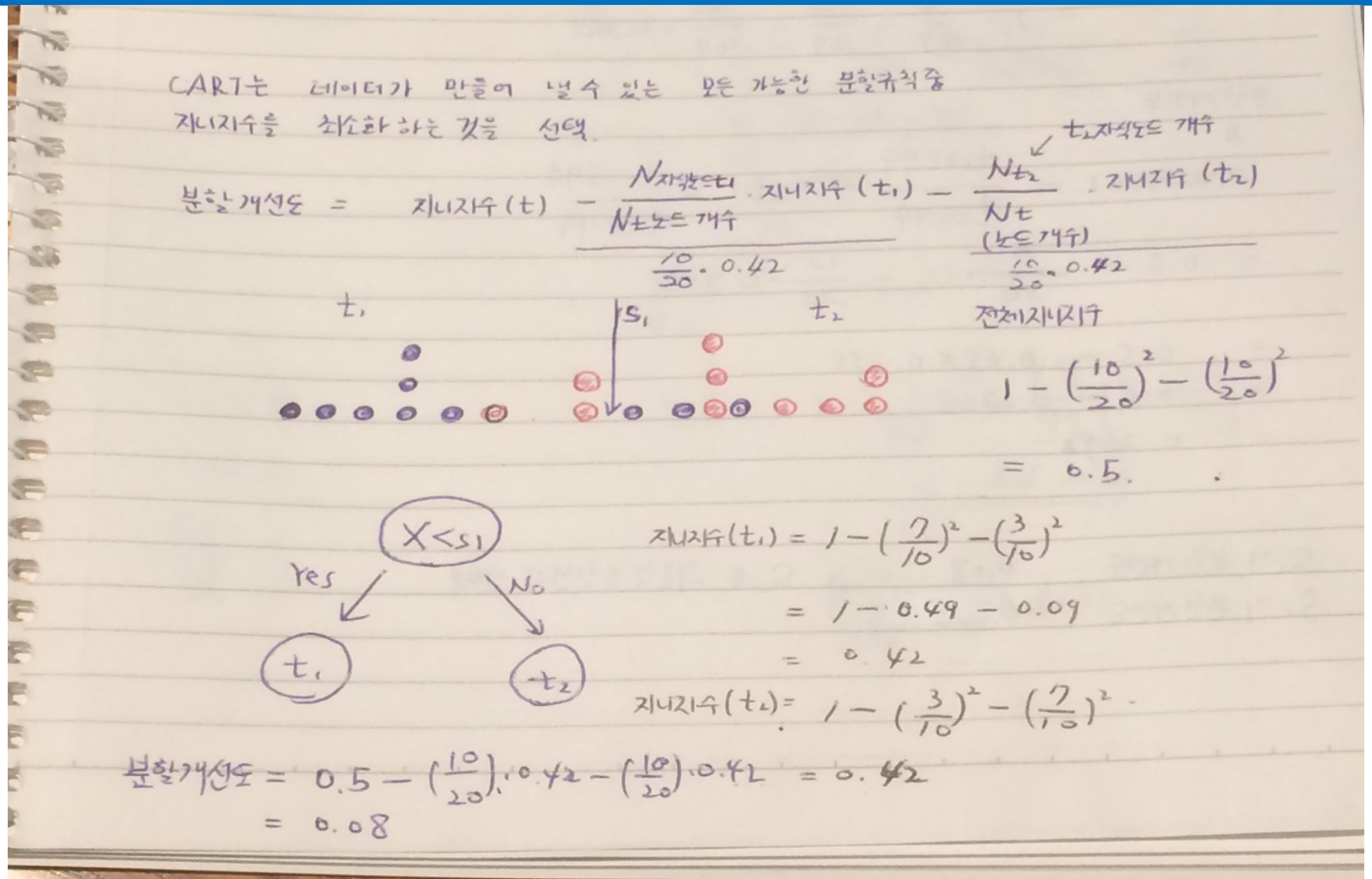
(Case 1) 만약 t 노드 내 관측치들의 소속이 여러 집단에 걸쳐 섞여 있다면 p(j|t)는 모두 유사한 값을 갖는다.
그리고 **지니지수는 상대적으로 커진다.**

(Case 2) 만약 t 노드 내 관측치들이 한 집단에 집중되어 있다면, 지니지수는 0에 가까운 값을 갖는다.

순수도

중간노드를 분할하는 규칙을 찾을 때 자식노드의 순수도가 가장 높아지는 규칙을 채택.

10 분류(Classification)나무모형의 분할-지니계수



전체 지니 지수

분할하기 전의 파란공의 비율, 빨간공의 비율을 이용하여 구한 지수

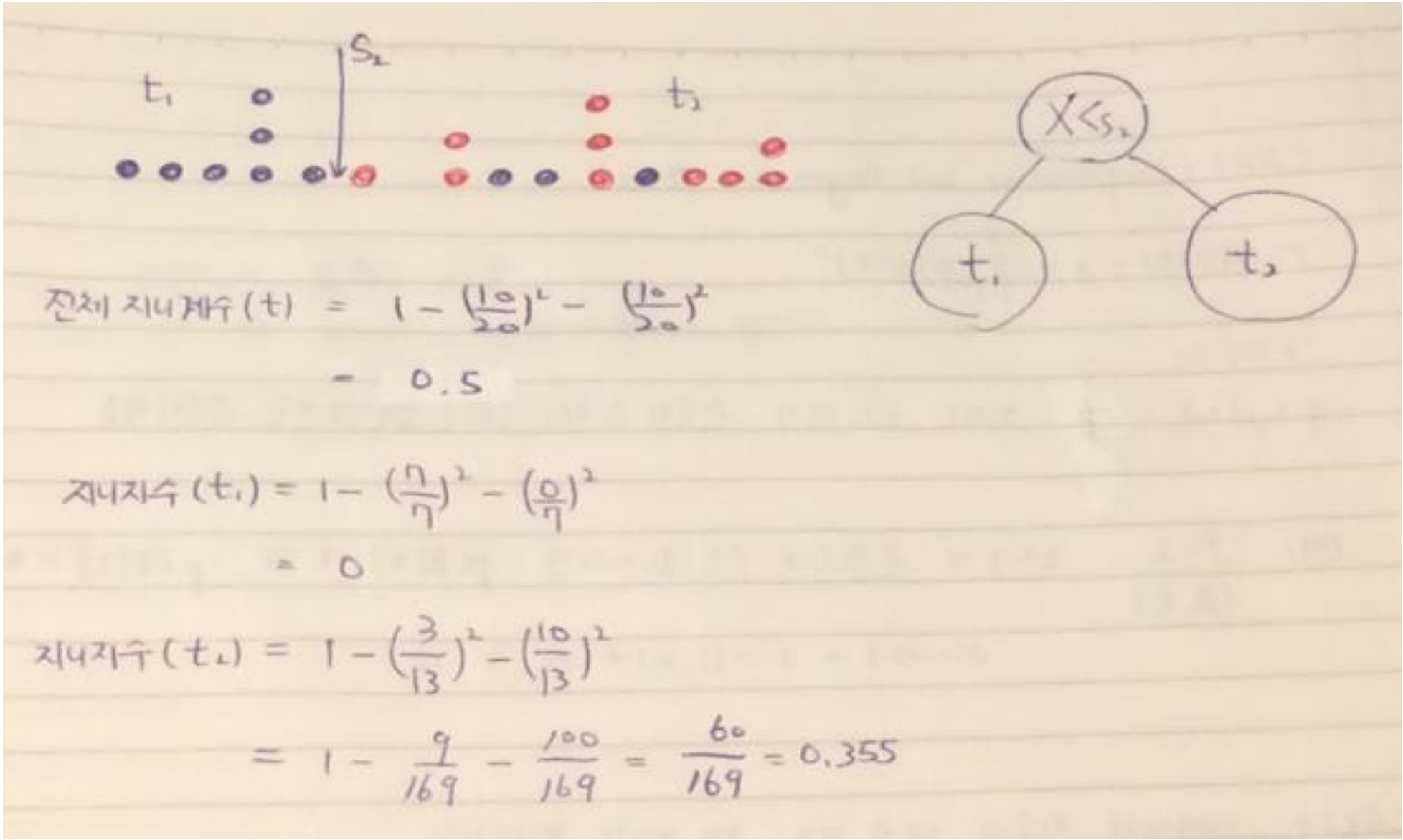
분류의 변수 분할점

분할개선도가 가장 큰 것을 택한다.

지니지수

같은 집단의 데이터가 많을 수록 지니지수의 값은 적어진다.

10 분류(Classification)나무모형의 분할-지니계수



분할점을 S2로 변경

(1) 먼저 전체 지니지수를 구하고

(2) 각각의 노드별 지니지수를 구한다.

(3) 분할 계선도를 최종적으로 확인한다.

(4) 개선도가 가장 큰 분할지점으로 선택한다.

10 분류(Classification)나무모형의 분할-지니계수

지니지수(t_2) $= 1 - \left(\frac{3}{13}\right)^2 - \left(\frac{10}{13}\right)^2$

$= 1 - \frac{9}{169} - \frac{100}{169} = \frac{60}{169} = 0.355$

분할 기준

⇓

지니지수(t) $-$ $\frac{t_1 \text{노드의수}}{t\text{노드의수}} \cdot$ 지니계수(t_1) $-$ $\frac{t_2 \text{노드의수}}{t\text{노드의수}}$

$= 0.5 - \frac{7}{20} \cdot 0 - \frac{13}{20} \cdot 0.355$

$= 0.5 - 0.65 \times 0.355$

$= 0.5 - 0.2308$

$= 0.2692$

S_1의 분할 기준 0.08

S_2의 분할 기준 0.2692

⇒ S_2를 최종적 분할점으로 선택

10 나무모형의 분할-회귀(Regression)

- CART 분류 나무 방법과 유사한 과정을 거쳐 구축된다.
- 다른점은 분할을 위해 사용했던 불순도 함수 대신에 분산 함수를 사용.

$$\text{불순도(t)} = \frac{1}{N} * \sum_{y_i \in t}(y_i - \bar{y}_t)^2$$

t노드에 속하는 데이터 각각의 값들에 대한 분산

y_i : 데이터 관측치, $\bar{y}_t$: 데이터 관측치의 표본평균

$$\text{분할 개선도} = \text{지니지수(t)} - \frac{N_{t_1}}{N_t} * \text{지니지수}(t_1) - \frac{N_{t_2}}{N_t} * \text{지니지수}(t_2)$$

10 나무모형의 분할-회귀(Regression)

- CART 분류 나무 방법과 유사한 과정을 거쳐 구축된다.
- 다른점은 분할을 위해 사용했던 불순도 함수 대신에 분산 함수를 사용.

$$\text{불순도(t)} = \frac{1}{N} * \sum_{y_i \in t} (y_i - \bar{y}_t)^2$$

$$\text{분할 개선도} = \text{불순도(t)} - \frac{N_{t1}}{N_t} * \text{불순도}(t_1) - \frac{N_{t2}}{N_t} * \text{불순도}(t_2)$$

11 불순도 함수의 종류

지니지수(t) = $1 - \sum_{J=1}^{J} p(j|t)^2$

엔트로피(t) = $- \sum_{J=1}^{J} p(j|t) * \log_2 p(j|t)$

디비언스(t) = $- 2 * \sum_{J=1}^{J} n_j \log_2(p(j|t))$

분류오분류율(t) = $1 - \max\{ p(1|t), p(2|t), \cdots \cdots , p(J|t) \}$

11 불순도 함수의 종류

알고리즘	평가지수	비고
ID3	Entropy	
C4.5	엔트로피 (Information Gain)	
C5.0	엔트로피 (Information Gain)	C4.5와 거의 비슷
CHAID	카이제곱(범주), F검정 (수치)	통계적 접근
CART	Gini index(범주), 분산의 차이(수치)	통계적 접근

12 분할 규칙(QUEST 방법)

(1) 변수 선택

연속형 입력변수와 목표 변수를 이용하여 분산분석(Oneway ANOVA)를 수행하고 P-값을 구한다.

범주형 입력변수도 목표변수와의 분할표를 생성하고, 카이제곱검정에 의해 P-value를 구한다.

가장 작은 p-value를 갖는 변수가 가장 유의한 변수이므로 해당변수를 분할변수로 선택

(2) 분할점 선택

(1) CART의 분할점 선택 방법을 그대로 적용

(2) 연속형 입력변수의 경우, 2차 판별분석방법을 사용하여 분할점 선택. 범주형의 경우, Crimcoord라는 방법을 이용하여 범주형->연속형으로 변형 후, 2차 판별분석 사용.

(3) 장점

(1) QUEST는 연산속도가 빠르다. CART의 약점을 보완하였다.

http://bitly.kr/G4YqS0C : public.dhe.ibm.com 사이트 알고리즘 설명

13 분할 규칙(CRUISE 방법)

(1) 변수를 선택하고, 선택된 변수에 대해서 분할점을 선택한다.

(2) QUEST 방법의 단점인 변수 선택 부분을 개선하고, 변수간 상호작용을 좀 더 적극적 반영함.

(1) 변수 선택

A. 분산분석 대신에 카이제곱분할표 검정을 사용.

(왜? 분산분석을 사용하면 집단 간 분포가 다르지만, 평균과 분산이 유사한 경우에는 해당 변수 선택이 어렵다)

B. 연속형 입력변수에 대해서 사분위수를 이용한 범주화를 수행. 카이제곱분할표 검정 결과 유의한 변수 분할 변수 선택 (변수간 상호 작용의 유의성도 고려됨). REF(Kim and Loh, 2001) 논문 참조

(2) 분할점 선택

A. CART 분할점 선택방법 적용 가능.

B. 연속형 – 1차 판별분석 사용 분할점 선택, 범주형 – Crimcoord 방법으로 범주형을 연속형으로 변형 후, 1차 판별 분석 방법 사용. 1차 판별 분석을 수행하기 전에 박스-콕스 변환(Box-Cox transformation)를 수행. 분할점이 수행된 이후에 분할점에 대해서 박스-콕스 역변환을 통해 원자료 단위로 변환해 주어야 함.

13 분할 규칙(CRUISE 방법)

(1) 변수를 선택하고, 선택된 변수에 대해서 분할점을 선택한다.

(2) QUEST 방법의 단점인 변수 선택 부분을 개선하고, 변수간 상호작용을 좀 더 적극적 반영함.

(3) 장점

CRUISE 방법은 결측치가 있는 경우에 결측치를 처리하는 다양한 방법을 제시.

14 의사 결정 트리의 크기 선택하기

지나치게 많은 노드와 가지를 가진 나무모형은

(1) 해석이 복잡하다.

(2) 과적합 문제(새로운 자료에 잘 맞지 않음)

그렇다면 이를 어떻게 적절하게 제어할 수 있을까?

14 의사 결정 트리의 크기 선택하기

(1) 분할 정지 방법

(2) 가지 치기(pruning) 방법

14 의사 결정 트리의 크기 선택하기

(1) 분할 정지 방법

CHAID 방법에서 사용. 나무 모형에서 나무 구조를 만들어 갈 때, 단계마다 분할이 꼭 필요한지 통계적 유의성을 이용하여 평가한다.

분할이 필요하면 분할을 하고, 분할이 필요하지 않다면 분할을 정지한다.

따라서, 나중에 정말 유의한 것에 대한 발견이 안될 수 있다.

(2) 가지 치기(pruning) 방법

14 의사 결정 트리의 크기 선택하기

(1) 분할 정지 방법

CHAID 방법에서 사용. 나무 모형에서 나무 구조를 만들어 갈 때, 단계마다 분할이 꼭 필요한지 통계적 유의성을 이용하여 평가한다.

분할이 필요하면 분할을 하고, 분할이 필요하지 않다면 분할을 정지한다.

따라서, 나중에 정말 유의한 것에 대한 발견이 안될 수 있다.

(2) 가지 치기 방법

가치 치기 방법은 단계마다 분할의 유의성을 평가하지 않고, 일단 계속적으로 분할해 가도록 허용.

분할을 계속한 이후에 결과적으로 마지막 구해진 나무를 최대 나무(maximal tree)라고 한다.

최대 나무 구조 중에서 불필요한 가지. 즉 적절하지 않은 마디를 제거하여 적당한 크기의 나무구조를 갖는 나무 모형을 최종 모형으로 선택한다.

15 가지 치기(Pruning) 방법

(1) CART, C4.5, QUEST, CRUISE 방법이 채택

(2) 연산이 오래 걸린다.

(3) 분할 정지방법 보다 더 우월한 나무구조를 찾을 수 있다.

15 가지 치기(Pruning) 방법

(1) CART, C4.5, QUEST, CRUISE 방법이 채택

(2) 연산이 오래 걸린다.

가지 치기 방식 방법이 나무 구조를 아주 크게 만들고 나서 가지치기를 해야 하므로 연산이 오래 걸린다.

(3) 분할 정지방법 보다 더 우월한 나무구조를 찾을 수 있다.

15 가지 치기(Pruning) 방법

(1) CART, C4.5, QUEST, CRUISE 방법이 채택

(2) 연산이 오래 걸린다.

가지 치기 방식 방법이 나무 구조를 아주 크게 만들고 나서 가지치기를 해야 하므로 연산이 오래 걸린다.

(3) 분할 정지방법 보다 더 우월한 나무구조를 찾을 수 있다.

분할을 하기 전에 유의성을 검증했을 때, 유의하지 않은 분할이지만,

분할을 수행한 이후에 보았을때,

분할의 유의성이 있을 수 있을 수 있다. 분할 한 이후에 유의성을 찾아낼 수 있다는 장점이 있다.

16 가지 치기 (Pruning) 이론

어떤 기준을 가지고 가지 치기를 해야 할까?

(1) 최종 노드가 너무 많으면 해석이 어렵다.

(2) 가지치기를 과도하게 진행하면 노드가 너무 적어 오분류율(정답 못맞출 확률)이 증가한다.

16 가지 치기 (Pruning) 이론

(1) CART – 두가지 측면을 고려한 비용 복잡함수 이용

16 가지 치기 (Pruning) 이론

(1) CART – 두가지 측면을 고려한 비용 복잡함수 이용

비용 복잡 함수(α) = 오분률율(T) + α * |T|

T : 임의의 나무 모형

|T| : 나무모형 T의 최종 노드의 개수

α : 나무모형의 복잡도에 따른 벌점 모수(cost-complexity parameter)

16 가지 치기 (Pruning) 이론

(1) CART – 두가지 측면을 고려한 비용 복잡함수 이용

비용 복잡 함수(α) = 오분률율(T) + α * |T|

T : 임의의 나무 모형

|T| : 나무모형 T의 최종 노드의 개수

α : 나무모형의 복잡도에 따른 벌점 모수(cost-complexity parameter)

(A) 나무 모형 크기가 커지면

- 오분류율(T)는 감소(과적합)
- α * |T| 값은 증가

16 가지 치기(Pruning) 이론

비용 복잡 함수(α) = 오분률율(T) + α * |T|

T : 임의의 나무 모형

|T| : 나무모형 T의 최종 노드의 개수

α : 나무모형의 복잡도에 따른 벌점 모수(cost-complexity parameter)

(A) 나무 모형 크기가 커지면

- 오분류율(T)는 감소(과적합)
- α * |T| 값은 증가

(B) 나무 모형 크기가 작아지면(가지치기로)

- 오분류율(T)는 증가
- α * |T| 값은 감소

16 가지 치기(Pruning) 이론

비용 복잡 함수(α) = 오분률율(T) + α * |T|

T : 임의의 나무 모형

|T| : 나무모형 T의 최종 노드의 개수

α : 나무모형의 복잡도에 따른 벌점 모수(cost-complexity parameter)

(A) 나무 모형 크기가 커지면

- 오분류율(T)는 감소(과적합)
- α * |T| 값은 증가

(B) 나무 모형 크기가 작아지면(가지치기로)

- 오분류율(T)는 증가
- α * |T| 값은 감소

(C) α의 값

- $\alpha = 0$이면 규모가 큰 나무구조가 좋다.
- α값이 매우 크다면, 규모가 적은 나무가 좋다.

16 가지 치기(Pruning) 이론

비용 복잡 함수(α) = 오분률율(T) + α * |T|

T : 임의의 나무 모형

|T| : 나무모형 T의 최종 노드의 개수

α : 나무모형의 복잡도에 따른 벌점 모수(cost-complexity parameter)

(C) α의 값

- $\alpha = 0$이면 규모가 큰 나무구조가 좋다.
- α값이 매우 크다면, 규모가 적은 나무가 좋다.

결론적으로, 적절한 α 값에 대한 주관적 선택이 필요해진다.

16 가지 치기(Pruning) 이론

비용 복잡 함수(α) = 오분률율(T) + α * |T|

그러면 어떻게 어떻게 할까?

16 가지 치기(Pruning) 이론

비용 복잡 함수(α) = 오분률율(T) + α * |T|

CART에서는 α 값에 대응하는 나무에 따라..

α 값이 0에 해당되는 나무 T0, α 값이 0.1에 해당되는 나무 T1... 여러 개를 생성.

10-fold **교차 타당성 방법**에 의해 T0, T1, T2.., 중에 Err가 가장 적은 친구를 선택

Err(T*) = min {Err(T0), Err(T1), …, Err(T3),..}

Err(T*)는 T나무구조에 대한 10-fold 교차 타당성 오분류율

가장 적은 오분류율을 보이는 T*을 선택한다.

16 가지 치기(Pruning) 이론

비용 복잡 함수(α) = 오분률율(T) + α * |T|

10-fold 교차 타당성 방법에 의해 T0, T1, T2.., 중에 Err가 가장 적은 친구를 선택

Err(T*) = min {Err(T0), Err(T1), …, Err(T3),..}

Err(T*)는 T나무구조에 대한 10-fold 교차 타당성 오분류율

가장 적은 오분류율을 보이는 T*을 선택한다.

=> **QUEST, CRUISE** 방법도 동일하게 채택