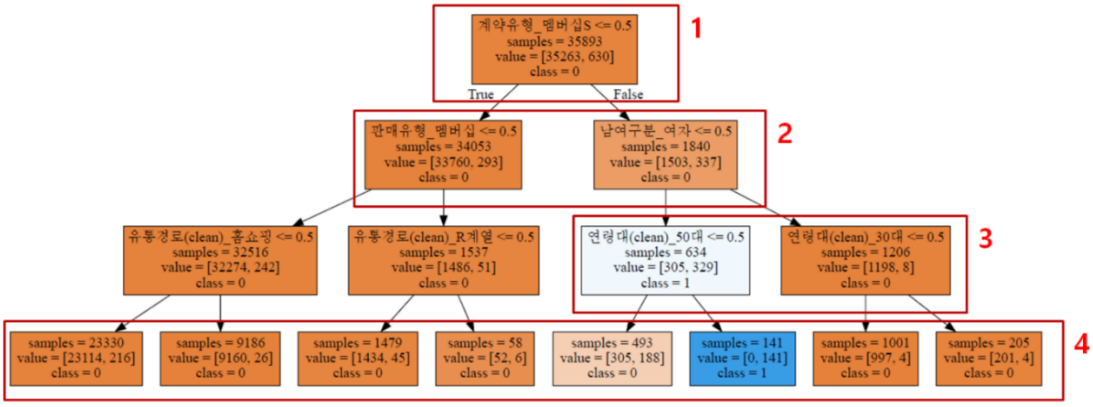
Decision tree

1. Decision tree?

   설명변수들의 규칙, 관계, 패턴 등으로 목표변수를 분류하는 나무구조의 모델을 만들고, 설명변수들의 관측값을 대입하며 목표변수를 분류/예측하는 지도학습 기법이다. 즉, 의사결정 규칙을 나무구조로 나타내어 전체 자료를 몇 개의 소집단으로 분류(classification)하거나 예측(prediction)을 수행하는 분석방법이라고 할 수 있다.

   다른 머신러닝 모델들에 비해 이해하기가 비교적 쉬우며, 정확도도 높은 편이라 자주 사용되는 모델 중 하나이다.

2. 구조 및 용어



Root node : 나무 구조가 시작되는 node (그림의 1번)

Child node : 상위의 node에서 분리된 node (그림의 2번은 1번의 child node)

Parent node : child node의 상위 node (그림의 1번은 2번의 parent node)

Internal node : 끝 node가 아닌 나무 중간에 있는 node (그림 3번)

Terminal node (or leaf) : 나무 각 줄기의 끝에 위치한 node(그림 4번)

Branch : 하나의 node로부터 끝 node까지 연결된 일련의 node들

Depth : Branch를 이루고 있는 node의 층 수 (위의 그림은 총 4층)

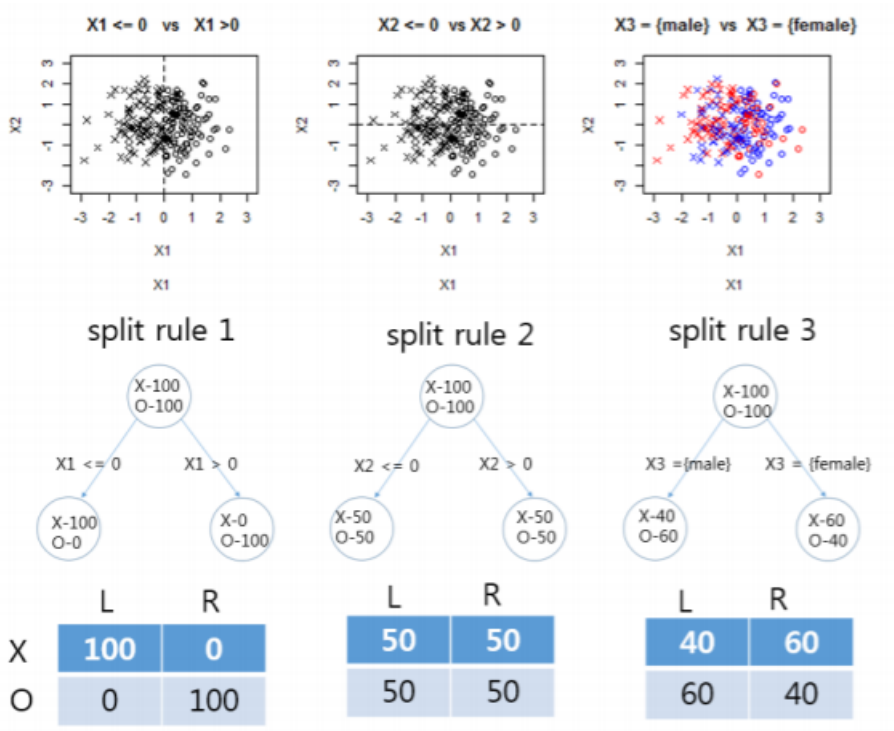
3. 분류 방법(split rule)

Root node에서 시작해 불순도를 가장 낮게 만드는 분류기준을 찾는다. 기준을 만족하면 왼쪽, 만족하지 못하면 오른쪽으로 보내 2개의 child node로 분류한다.

불순도를 계산하는 방법에는 Chi-square statistic, Deviance, Misclassification error, Gini index, Entropy index 등이 있는데 주로 Gini index와 Entropy index가 사용된다.

분류나무(classification tree) : 이산형(범주형) 목표변수의 경우 목표변수의 각 범주에 속하는 빈도에 기초하여 분리가 일어남.

회귀나무(regression tree) : 연속형(구간형) 목표변수의 경우 목표변수의 평균과 표준편차에 기초하여 분리가 일어남



➔ 위와 같은 데이터가 있다고 가정 했을 때 각 분류기준(split rule)에 따라 어떻게 분류되는지 살펴보자

1) Gini index

$$G_i = 1 - \sum_{k=1}^{n} {p_{i,k}}^2$$

위의 예시에서는 3가지의 split rule에 따라 데이터가 분류되었다. 3가지 중 어떤 split rule을 따를지 결정하기 위해서는 각 split rule에 따른 gini index와 ∆(worth)를 계산해 보아야 한다. ∆(worth)란 각 split이 얼마나 쓸모 있는가를 계산한 것으로, 만약에 ∆가 크면 child node의 불순도가 작아진 것이니까 그 split이 의미있다라고 할 수 있다.

계산 공식은 다음과 같다 → ∆(t) = ı(t) − [pLı(tL) + pRı(tR)].

여기서 ı(t)는 parent node의 불순도이고, pLı(tL)은 (왼쪽 childe node로 갈 확률) x (왼쪽 child node의 불순도), 마찬가지로 pRı(tR)은 (오른쪽 childe node로 갈 확률) x (오른쪽 child node의 불순도) 이라고 할 수 있다.

위의 예시에서 Root node의 gini index는 $1 - \left[\left(\frac{100}{200}\right)^2 + \left(\frac{100}{200}\right)^2\right] = 0.5$이다. 여기서 split rule1에 따라 데이터를 분류했을 때 왼쪽 데이터의 gini index는 $1 - \left[\left(\frac{100}{100}\right)^2 + \left(\frac{0}{100}\right)^2\right] = 0$이고, 오른쪽 데이터의 gini index는 $1 - \left[\left(\frac{100}{100}\right)^2 + \left(\frac{0}{100}\right)^2\right] = 0$이다. 따라서 split rule1의 ∆(t)를 계산해보면 $0.5 - \left[\left(\frac{100}{200}\right) * 0 + \left(\frac{100}{200}\right) * 0\right] = 0.5$이다.

같은 방식으로 split rule 2,3의 ∆(t)를 계산해보면 각각 0, 0.02가 나온다. 결과적으로 split rule1의 ∆(t)가 가장 크므로 split rule1에 의해 분리가 된다. Decision tree는 같은 node끼리는 동질성이 크고 다른 node끼리는 이질적이도록 분류하기 때문에 자연스러운 결과라고 할 수 있다.

2) Entropy index

$$H_i = -\sum_{k=1}^{n} p_{i,k} \log_2(p_{i,k})$$

엔트로피 방식의 경우, 열역학에서 사용되는 '무질서도'에 대한 척도로써, Entropy가 낮을수록 질서 정연한 상태라는 개념이 이 알고리즘에서도 동일하게 적용된다. 식에서 $p_{i,k}$는 i번째 node에 있는 데이터 중 k범주에 속한 데이터의 비율을 의미한다.

위 예시에서 root node의 entropy index는 $-[\left(\frac{100}{200}\right)log_2\left(\frac{100}{200}\right)+\left(\frac{100}{200}\right)log_2\left(\frac{100}{200}\right)]=1$이다. Split rule1일 때 왼쪽 데이터의 entropy index는 $-[\left(\frac{100}{100}\right)log_2\left(\frac{100}{100}\right)+\left(\frac{0}{100}\right)log_2\left(\frac{0}{100}\right)]=0$이고 오른쪽 데이터의 entropy index는 $-[\left(\frac{0}{100}\right)log_2\left(\frac{0}{100}\right)+\left(\frac{100}{100}\right)log_2\left(\frac{100}{100}\right)]=0$이다. 따라서 split rule1의 Δ(t)를 계산해보면 $1-\left[\left(\frac{100}{200}\right)*0+\left(\frac{100}{200}\right)*0\right]=1$이다.

같은 방식으로 split rule 2,3의 Δ(t)를 계산해보면 각각 0, 0.03이 나온다. 결과적으로 split rule1의 Δ(t)가 가장 크므로 split rule1에 의해 분리가 된다.

+ 추가로 사용되는 계산 방법

3) Chi-square statistic : 카이제곱통계량이 최대가 되는 분리 사용

$$\chi^2=\sum(\text{관측값}-\text{기댓값})^2/\text{기댓값}$$

4) 분산분석 F통계량(F-Statistics)

F통계량은 그 값이 클수록 오차의 변동에 비해 처리(treatment)의 변동이 크다는 것을 의미 -> 자식노드(처리들)간이 이질적임을 의미 -> F통계량이 커지는(p value는 작아지는) 방향으로 가지분할을 수행.

5) 분산의 감소량(Variance reduction)

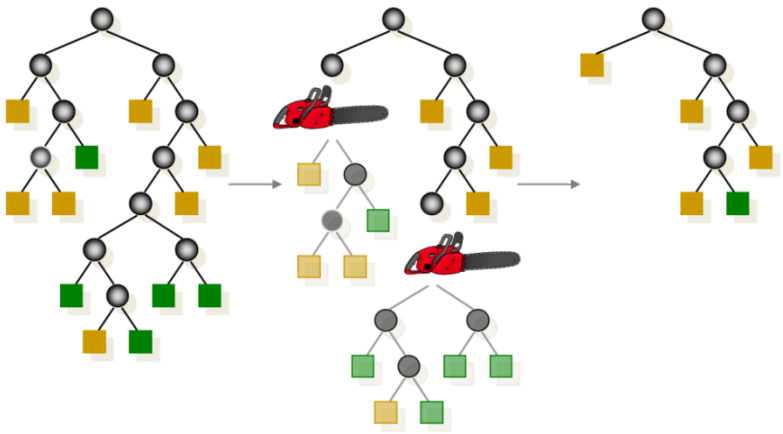
4. 가지치기(Pruning)

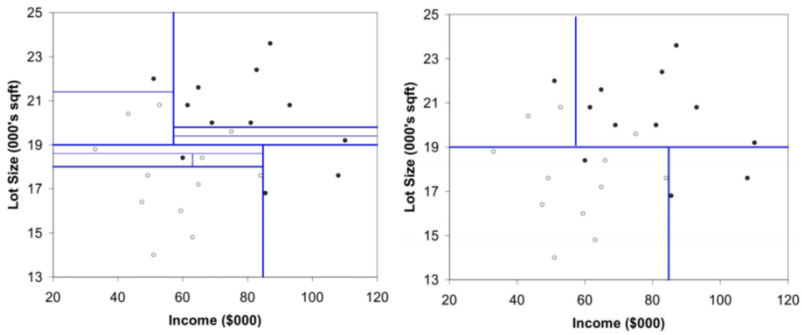
가지치기는 의사결정나무 모델에서 적합성에 큰 영향을 주는 과적합(overfitting)을 방지하고 적절한 적합성을 가진 모델을 만들기 위해 실시한다. Overfitting은 train date가 과도하게 fitting된 상태로, 모든 terminal node의 순도가 100%인 상태이지만 조건이 상세하고 복잡해 분류모델 해석이 매우 어려우며, 새로운 데이터에 대한 분류정확도가 매우 떨어진다.

가지치기는 크게 '사전 가지치기'와 '사후 가지치기'로 나눌 수 있다.

'사전 가지치기'는 Decision tree의 분류정지 조건을 사전에 설정하여, 분할을 멈추도록 하는 방식이다.

'사후 가지치기'는 full tree를 먼저 생성한 뒤, 모델에 대한 해석과 평가가 완화되는 방향으로 tree의 branch를 쳐내는 방식이다.





5. 장단점

장점

- 구조가 단순하여 모델의 결과해석이 쉽고, 데이터 전처리를 간단하게 할 수 있다
- 범주형 데이터에도 적용 가능하다
- 선형성, 정규성, 등분산성 등의 수학적 가정을 만족하는지 확인할 필요 없이 간편하고 강력하게 사용 가능한 비모수적 방법이다.

단점

- 기준값의 경계선 근방의 자료 값에 대해서는 오차가 클 수 있다.
- Overfitting의 위험성이 크다
- 선형성이 미흡해 모델의 안정성이 낮다

출처

https://blog.naver.com/rlacksdid93/221482013474