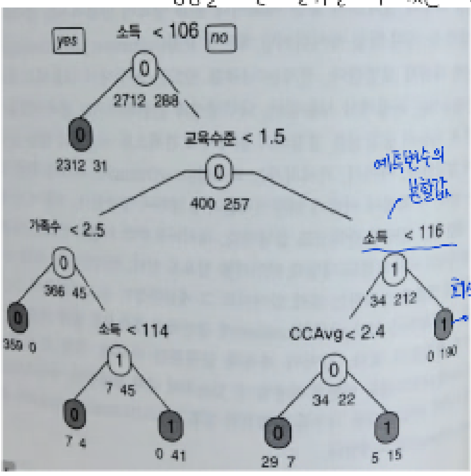
9장 분류회귀나무

1. 서론

분류회귀나무 ~ 분석가의 많은 노력 없이도 사용자가 분석정보를 이해하기 쉽고, 다양한 상황에서도 성능을 잘 발휘할 수 있는 범용적인 분류기법



모델이 제공하는 분류기준이 이해하기 쉽다는 장점.

IF (소득 ≥ 106) AND (교육수준 < 1.5) AND (가족수 < 2.5) Then 클래스 = 0 (비수락)

2. 분류나무

두 가지 핵심 아이디어 { 예측변수의 공간을 재귀적으로 분할 / 검증세트를 이용한 가지치기

· 재귀적 분할

종속변수 Y, 예측변수 $X_1, X_2, X_3, \cdots, X_p$ → 분류문제에서 모델의 결과변수는 범주형이어야 함

재귀적 분할은 X 예측변수들의 P차원 공간을 서로 겹침이 없는 다차원의 직사각형들로 나눔.

↳ 연속형, 이진형, 서열형 등이 될 수 있다.

이러한 방식의 분할은 이전 분할의 결과에 영향을 받으며 재귀적으로 이루어짐.

먼저 하나의 변수 X_i가 선택되고, P차원 공간을 두 부분으로 분할하게 하는 변수값 S_i 선택.

↳ 이 중 한 부분은 $X_i < S_i$인 모든 점 포함, 나머지 부분은 $X_i \geq S_i$인 모든 점 포함

그 후, 나뉘진 두 부분들 중 한 부분은 마찬가지로 동일한 과정에 의해 두 부분으로 분할.

⇒ 전체 X공간을 분할한 모든 직사각형들이 가능한 동질적으로 순수하게 되도록 만드는 것.

그럼 어떻게 이러한 특정분할을 선정할 것인가? → 최적의 분할을 찾기 위해서 변수와 각 변수에 대한 모든 분할 가능값을 검토. → 변수가 갖는 연속된 두 값의 중간값.

⇒ 이러한 분할값들은 분할의 결과로 생기는 직사각형에서 얼마나 많이 불순도를 감소시켰는지를 통해 순위가 매겨짐.

↳ 직사각형 안에 같은 클래스에 속한 점들이 얼마나 많이 들어있는가?

↳ 불순도로 감소시킨다는 것은 [illegible]

- 범주형 예측변수

범주형 예측변수에서는 데이터를 2개의 하위집합으로 나눌 수 있는 모든 범주들이 분할 가능값으로 선택 가능

범주의 수가 많을 경우에는 분할 가능한 방법의 수가 매우 많아지게 된다.

◦ 불순도 측정

대표적인 불순도 측정방법: 지니 지수 (Gini index), 엔트로피 측도 (entropy measure) 등

반응변수가 m개의 클래스를 갖고, $k=1, 2, \cdots, m$으로 표기.

직사각형 A에 대한 지니 불순도 지수 = $I(A) = 1 - \sum_{k=1}^{m} P_k^2$ ← P_k는 직사각형 A에서 k클래스에 속하는 관측치들의 비율. $P_k = 0.5$ 일 때 최대값

↳ 측정치는 0 (모든 관측치들이 같은 클래스) ~ (m-1)/m (관측치들이 m개의 클래스에 동일하게 분포) 사이의 값.

직사각형 A에 대한 엔트로피 지수 = $Entropy(A) = -\sum_{k=1}^{m} P_k \log_2(P_k)$ ← $P_k = 0.5$ 일 때 최대값

↳ 0 (모든 관측치들이 동일 클래스) ~ $\log_2(m)$ (관측치들이 m개의 클래스에 동일한 분포) 사이의 값.

분할을 통해 생성된 두 직사각형의 불순도를 결합한 값 → 두 직사각형에 대한 불순도 측정치의 가중평균

↳ 가중치는 직사각형별 관측치의 수에 따라 달라짐

모든 가능한 예측변수 별로 모든 가능한 분할에 대해 불순도 감소를 비교해 봄으로서 다음 분할 선택.

↳ 재귀적 분할 과정은 알고리즘이 선행되는 동안 만들어 낸 직사각형이 보다 순수하게 정제되어감.

↳ 마지막 단계에서는 하나의 클래스 점들만 가지고 있음 (불순도 = 0)

◦ 나무구조

두 가지 종류의 노드 : 결정노드 (Decision node), 단말노드 (Terminal node)

결정노드 - 자식을 가지고 있는 노드. 단말노드 - 예측변수에 의한 데이터 분할

◦ 새로운 관측치의 분류

새로운 관측치를 분류하기 위해서는 관측치를 나무 아래로 떨어뜨리는, 나무모델을 위에서 아래로 할당하는 과정을 거침

클래스 할당에는 '투표' 방식이 일반적이지만, 관심있는 클래스가 있는 경우, 확률을 추정하여 컷오프값 적용 가능.

3. 분류나무의 성능평가

모델을 평가하고 조정하기 위해 학습데이터 이외의 샘플데이터가 필요

나무구조는 선택된 샘플에 따라 다소 불안정하고 크게 변할 수 있다.
완전 적합된 나무모델은 대개 과적합(overfitting)을 초래한다. } → 두 가지 측면에서 학습데이터 이외의 데이터 더욱 필요

데이터를 A, B 두 샘플로 나누고, 두 샘플로 각각 나무모델을 만들었을 때,

A, B 양 측에서 비슷한 예측력을 가지는 예측변수들이 있으면 두 모델의 산출결과는 완전히 다를 수 있음.

예측된 클래스는 혼동행렬표 (confusion matrix)를 통해 실제 클래스와 비교 가능

특정 클래스에 관심이 있는 경우 향상차트 (lift-chart)를 통해 성능 평가 가능

일반적으로 학습데이터에 대해서는 학습세트로 만들어진 모델이 좋은 성능을 보이지만

검증데이터나 새로운 데이터에 대해서는 그렇지 않음 → 과적합을 방지할 방법필요.

4. 과적합 방지하기

과적합 → 새로운 데이터에 대해 좋지 않은 성능을 가져옴

↳ 나무모델에서는 과적합이 일어나는 점에 이르기까지 분할 수준의 수는 증가하고 이에 따라 총 오류수는 감소.

↳ 학습데이터에 대해서는 분할이 계속되어 총 오류가 0이 될 때까지 계속 분할

↳ 새로운 데이터에 대해서는 나무모델이 클래스와 예측변수 사이의 관계를 모델링하는 시점까지만 오류감소.

↳ 그 시점이 지나면 학습데이터의 노이즈까지 학습해서 검증이나 새로운 데이터에 대한 오류증가

나무모델의 분할수준이 높을 때는 노드들이 매우 적은 수의 관측치를 사용하여 분할.

↳ 클래스 간 차이가 예측변수가 주는 정보를 기반으로 발생했다기보다 잡음에 의해 발생했을 가능성이 높다.

- 나무모델 성장 멈추기: 조건부 추론 나무

과적합되기 전에 성장을 중단시키기 위한 여러 기준 → 나무모델의 깊이(분할회수), 노드 내의 최소 관측치의 수, 불순도 감소량 등

↳ 이러한 기준들로도 성장을 멈출 적절한 시점 결정은 어려움.

기존 방법들은 재귀적 분할 아이디어에 기초 → 과도하게 성장하여 과적합하게 되는 것을 방지하기 위한 규칙 사용

↳ 보편적인 방법은 CHAID (chi-squared automatic interaction detection)

노드를 더 분할할 지 결정하기위해 통계검정기법 (독립변수에 대한 카이제곱 검정) 사용

↳ 노드의 분할 결과로 얻어진 순도의 증가량이 통계적으로 유의할 때 분할.

↳ 각 노드에서는 반응변수와 가장 큰 연관성을 보이는 예측변수로 분할

↳ 연관성의 강도는 카이제곱 독립성 검정에 의한 P값으로 측정

통계적 유의성이 확보되지 않으면 분할은 종료. → 범주형 예측변수에 더 적합하지만 연속형 변수도 범주화 통해 적용가능

- 나무모델의 가지치기

완전히 성장한 나무모델을 가지치기 → 성장을 멈추는 것보다 더 성공적이라고 알려짐

기본아이디어 - 매우 큰 나무모델은 과적합되기가 쉽고 오분류율을 줄일 수 없는 약한 가지들은 제거

↳ 결정노드를 고르고 그 노드를 잎노드로 재설정하는 일련의 과정을 반복하면서 진행

└ 학습데이터의 잡음이 아닌 패턴을 찾아내는 나무모델을 얻기 위해 두 대립요소, 검증데이터의 오차율과 나무모델의 결정노드 수가 타협하는 점을 찾는다.

- 교차검정

검증데이터를 통한 가지치기 → 과적합 문제 해결. 그러나 모델의 불안정성 문제는 해결될까.

└ 최상위 레벨에서 어떤 변수를 사용해 분할할지가 안정적이지 않음 → 성장해가면서 끝까지 영향을 줌.

⇒ 데이터를 학습과 검증으로 한 번만 나누지 않고 반복적으로 여러 번 교차검정하여 결론을 이끌어 내는 것.

└ 교차검정으로 나무모델을 단순히 측적하는 것으로 많은 알을 못하지만 최종모델이 얼마나 깊게 자라는지는 알 수 있다.

└ 나무모델이 얼마나 깊게 자라는지 측정하고 조정할 수 있는 파라미터 → 교차검정 시 최소 오류로 적합된 나무모델들의 파라미터 평균값을 구함.

└ 이 값은 새로운 데이터가 주어졌을 때, 최적의 나무 깊이로 활용되어 나무의 성장을 제한하는데 활용

비용복잡성 기준 (Cost complexity; CC) $CC(T) = Err(T) + \alpha L(T)$

- $Err(T)$: 오분류된 학습 세트 관측치들의 비율
- α: 크기에 대한 벌점요인
- $L(T)$: 잎노드의 개수

└ 나무모델의 오분류율 (학습세트에 대한)과 나무모델의 크기에 대한 벌점요인을 합해놓은 것.

└ $\alpha = 0$ 일 때는 아무리 많은 노드를 가져도 벌점 X. ⇒ 완전히 성장한 (오분류율이 0인) 나무모델 선택

└ α가 큰 값일 때는 가장 적은 수의 잎노드를 가진 나무모델 선택. ↳ 모델의 크기가 α에 의해 결정

· 교차검정의 과정

① 학습과 검증으로 데이터 분할

②학습데이터를 이용해 나무 성장

③ 각 단계마다 CP를 따라 가지치기.

④ 검증데이터에 대해 가장 작은 오류가 나는 CP를 기록

⑤ 데이터를 다시 학습과 검증으로 나누고 위의 과정 반복.

⑥ 계속 반복하여 얻은 CP값의 평균을 구함. 이 때, CP값은 각 나무모델의 최소 에러를 완성한 값.

⑦ 원본데이터 또는 앞으로 작업할 데이터로 돌아가서, 위 과정을 통해 얻은 최적의 CP값(평균)을 활용해 성장을 멈춤.

일반적으로 교차검정에서는 학습과 검증데이터가 겹치지 않도록 분할.

- 최적으로 가지치기된 나무모델.

모델을 되도록이면 복잡하지 않게 만든다는 측면에서 샘플링 오차를 함께 고려

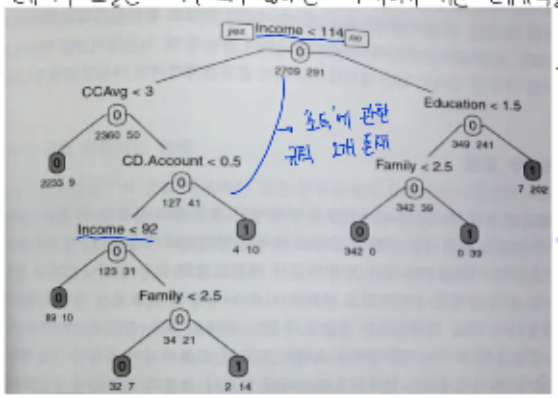
└ 다른 샘플을 취했을 때 최소에러값이 바뀌게 될지도 모르는 불안을 개선 → 나무를 더 가지치기하기 위해 교차검정 에러에 대한 추정 표준오차(xstd) 사용

→ 최소 교차검정 오류율(xerror)에 대해 1 수준의 추정 표준오차를 추가.

↳ '최적으로 가지치기된 나무모델'은 이 범위에서 최대한 많이 가지치기된 모델.

5. 분류나무 모델에서의 분류규칙

분류나무 모델은 너무 크지 않다면 이해하기 쉬운 분류규칙을 제공



규칙의 수는 중복된 규칙을 제거함으로서 줄일 수 있다.

절차가 명료하고 알고리즘을 이해하기 쉽기 때문에

분류가 최종 목적이 아닌 경우에도 유용

6. 세 개 이상의 클래스를 분류하기

분류나무 모델 → 세 개 이상의 클래스를 가진 분류문제에도 유용하게 사용

불순도 측정에서 지니계수나 엔트로피 지수 → m클래스로도 쉽게 확장 가능.

7. 회귀나무 모델

나무모델을 이용한 방법론 → 반응변수가 연속형인 경우에도 적용가능 → 분류나무와 상당히 유사

- 반응변수의 값을 예측하는 방법

  관측치의 반응변수 Y값의 예측 → 분류문제와 유사

  결정노드에서의 값은 해당 노드에 속하는 학습데이터들의 평균값으로 결정

- 불순도 측정 방법

  회귀나무 모델 → 잎노드의 평균에 대한 제곱편차 합.

  잎노드의 평균값은 예측값으로 쓰이기 때문에 이 값은 제곱오차와 동일.

- 성과평가

  예측성능은 다른 예측모델과 마찬가지로 RMSE와 같은 요약측도들이나 향상차트와 같은 차트방법 사용

8. 예측력 향상 : 랜덤 포레스트와 부스트나무

규칙을 시각화하는 것이 중요하지 않은 오직 '예측'만을 위한 문제

↳ 다중 나무의 결과를 조합하는 몇 가지 확장된 나무모델이 더 나은 성능을 제공

- 랜덤 포레스트

  기본 아이디어

① 데이터로 부터 복원추출 방식으로 여러 랜덤 샘플들을 생성 (이러한 추출을 부스트랩 (bootstrap) 이라고 함.)

② 각 단계마다 무작위로 예측변수들을 선택하여 서브셋을 만들고, 샘플에 대해 나무를 적합 (그 결과 '포레스트' 얻음)

③ 예측을 향상시키기 위해 각 나무들로부터 얻은 예측/분류 결과를 결합. 분류에 대해선 '투표', 예측에 대해선 '평균화'

단일 나무와는 다르게 랜덤 포레스트로 부터 얻은 결과를 나무 다이어그램으로 표현불가.

⇒ 단일 나무에서 제공하는 해석력을 잃어버림.

랜덤 포레스트는 각기 다른 예측변수의 상대적인 기여도를 측정하는 "변수 중요도 (variable importance)" 척수를 제공

ㄴ특정 예측변수에 대한 중요도 점수는 포레스트에 속한 모든 나무들이 이 변수에 의해 감소한 지니지수를 합산하여 계산

- 부스트 나무

나무들이 순차적으로 구성, 각 나무는 이전 나무의 오분류 레코드 고려

① 단일 나무를 생성

② 잘못 분류된 레코드들에 가장 높은 선택확률을 제공하는 하나의 샘플을 추출.

③ 새로운 샘플을 나무에 학습

④ 단계 2, 3을 여러 번 반복.

⑤ 레코드들을 분류하기 위해서 가중투표를 사용. 이 때 가중치의 크기는 구성되는 나무들의 엑손으로 부여.

부스트 나무는 오분류된 레코드에 중점을 두기 때문에 자연스럽게 오분류율이 감소

9. 나무모델의 장단점

나무모델 → 분류와 예측을 수행하기에 편리하도록 패키지화된 좋은 도구.

ㄴ 나무의 상단 부분들에 나타나는 변수들이 일반적으로 가장 중요한 변수이므로 변수선택기법에도 유용하게 쓰임.

아래와 같은 측면에서 사용자들에게 상대적으로 적은 수고만을 요구.

① 변수를 변환시킬 필요가 없다. (변수들에 대한 어떤 단조변환에 대해서도 동일한 결과 산출)

② 변수에 대한 부분집합 선택이 자동으로 이루어진다. → 변수 선택이 나무 분할과정의 일부분

나무모델은 분할에 대한 선택이 절대값 '크기'에 의해서가 아니고 '순서'에 의해 결정

⇒ 이상치 (outlier)에 강건

그러나, 데이터의 변화에 민감 ⇒ 작은 데이터 변화가 전혀 다른 분할을 초래하기도 함.

분류회귀 나무 모델 - 비선형, 비모수 모델 - 예측변수와 반응변수가 모델에서 다항한 관계를 맺을 여지를 줌.

ㄴ 단점으로도 작용가능 - 나무모델의 분할은 예측변수들의 조합보다는 단일 예측변수에 대해 이루어지기 때문.

└ 특히, 선형회귀나 로지스틱 회귀와 같은 선형구조를 가진 예측변수 간의 관계를 놓칠 수 있다.

나무모델은 예측공간을 수학적, 수적적으로 잘 분할할 수 있을 때 좋은 성능.

└ 그렇지 않을 경우 성능이 나빠진다. → 성능향상 위해서는 기존 예측변수에서 파생된 새로운 예측변수 생성

좋은 분류모델을 위해선 큰 데이터세트가 필요.

└ 모델을 성장시키는데 많은 계산량, 비용 ↑ 모든 변수들에 대해 분할을 검토하고 Sorting을 동반 + 검증세트를 이용한 검증

변수선택을 하기위해 나무모델을 사용하는 것은 유용하지만 모델이 선호하는 분할잠재력이 높은 변수는 많이 나타날 수 있음.

└ 다수의 범주를 작은 세트로 결합하고 수치값들을 묶음으로서 해결 가능

결측치를 대체하거나 결측치를 포함한 데이터를 삭제하지 않고도 결측치 처리 가능

가장 중요하고 실질적인 장점 - 모델이 생성하는 분류규칙이 명료

└ 이런 장점은 랜덤 포레스트, 부스트 나무 등에선 사라짐.