결정 트리

지니 계수

$$G.I(A) = \sum_{i=1}^{d} \left(R_i \left(1 - \sum_{k=1}^{m} p_{ik}^2 \right) \right)$$

- $P_{i,k}$ = 분할 후 I 영역에서 k 범주(클래스) 에 속하는 데이터의 비율, R_i = 분할 후 I 영역의 전체에 대한 비율
- 0~1 사이의 값인 지니 계수가 낮을 수록 데이터 균일도가 높다 (분리 전보다 분리 후에 데이터가 클래스별로 잘 모여있다)

학습 과정

- 모든 피쳐에 대해서 지니 계수가 가장 낮은 분기점(규칙 노드)으로 노드를 분리
- Ex. 하나의 피쳐에 대해서 오름차순으로 정렬 후 위에서부터 나눠서 (1, n - 1) (2, n – 2) … (n – 1, 1) 로 나누는 경우 중 지니 계수가 가장 낮은 분기점을 선택

가지 치기

- 사이킷런의 결정 트리 모델에서 하이퍼 파라미터를 기본으로 주었을 때 종단 노드의 클래스 종류를 한가지로 만드려고 하는데 이 때 학습데이터에 대한 과적합이 일어날 수 있다.
- 사이킷런의 결정 트리 모델에서는 하이퍼 파라미터의 값을 미리 설정함으로써 트리 구조를 제어 할 수 있다.

예측

- 학습 과정에서 만든 규칙 노드를 따라가며 종단 노드에 도착 했을 때 그 노드의 클래스 최빈값으로(회귀는 평균값) 데이터의 클래스 예측

참고

- https://ratsgo.github.io/machine%20learning/2017/03/26/tree/

Random Forest

부트 스트래핑 분할 방식

- 한 데이터 셋에서 무작위 복원 추출로 여러 개의 데이터 서브셋을 만드는 방식

학습

- 부트 스트랩을 통해 만든 데이터 셋의 개수만큼의 결정 트리를 각 데이터 셋에 대해 훈련 시킨다.

예측

- 예측 할 데이터를 학습에서 만든 결정 트리 모델들의 예측 값들에 대해 보팅을 통해 최종 예측 값을 결정한다.

장점

- 일부 결정 트리가 과적합 되어 있어도 여러 결정 트리로 보팅을 통해 예측을 하기 때문에 단일 결정 트리 모델의 과적합 문제가 해결될 수 있다.

참고

- https://ko.wikipedia.org/wiki/%EB%9E%9C%EB%8D%A4_%ED%8F%AC%EB%A0%88%EC%8A%A4%ED%8A%B8#%EB%9E%9C%EB%8D%A4_%ED%8F%AC%EB%A0%88%EC%8A%A4%ED%8A%B8
- https://tensorflow.blog/%EB%A8%B8%EC%8B%A0-%EB%9F%AC%EB%8B%9D%EC%9D%98-%EB%AA%A8%EB%8D%B8-%ED%8F%89%EA%B0%80%EC%99%80-%EB%AA%A8%EB%8D%B8-%EC%84%A0%ED%83%9D-%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98-%EC%84%A0%ED%83%9D-2/
- https://tensorflow.blog/%ED%8C%8C%EC%9D%B4%EC%8D%AC-%EB%A8%B8%EC%8B%A0%EB%9F%AC%EB%8B%9D/2-3-6-%EA%B2%B0%EC%A0%95-%ED%8A%B8%EB%A6%AC%EC%9D%98-%EC%95%99%EC%83%81%EB%B8%94/