

인공지능을 위한 머신러닝 알고리즘

4. 결정 트리

CONTENTS

1

의사 결정 트리와 분류 문제

2

결정 트리 구축 원칙

3

Bagging

학습 목표

- 의사 결정 트리의 분류 원리를 이해할 수 있다.
- Occam의 면도날 원칙이 결정 트리 구축 과정에 미치는 영향을 이해할 수 있다.
- Bagging 기법을 이해할 수 있다.



1. 의사 결정 트리와 분류 문제

■ 분류 (classification)

- 훈련 데이터의 특성 값 / 클래스 정보를 학습한 뒤, 새롭게 주어지는 테스트 데이터부터는 특성 값만으로 클래스를 예측하는 것

예> 날씨 정보를 바탕으로 상대방이 네이트를 어택해줄지 말지 예측

● 필요 사항

- 특성 값 (**attribute value**) : 고정된 집합의 특성들로 표현될 수 있어야 함

예> 더움, 따뜻함, 추움

- 클래스 (**class, target values**) : 이산 출력 값을 가져야 함

예> 예/아니오, 다중클래스

- 충분한 데이터 : 결정 트리에 주어질 충분한 훈련 예제

■ 분류 문제의 간단한 예제

다음 주에 경기하는 국내 축구팀의 **A**팀과 **B**팀의 경기에서 **A**팀이 이길지 혹은 질지 예측하는 모델을 만들어보자

● 가능한 특성 값들

- 경기 장소 : 홈 / 어웨이
- 경기 시간 : 오후 5시 / 7시 / 9시
- **S**선수의 포지션 : 공격수 / 미드필더
- **B**팀 **K**선수의 출전 여부 : 예 / 아니오
- ...

■ 데이터

장소	시간	K선수 출전	S선수 포지션	R선수 출전	날씨	결과
홈	7시	예	미드필더	예	맑음	승리
홈	7시	예	공격수	아니오	비	승리
어웨이	7시	예	공격수	예	맑음	승리
홈	5시	아니오	공격수	아니오	맑음	패배
어웨이	9시	예	공격수	예	비	패배
어웨이	7시	아니오	미드필더	예	맑음	승리
홈	7시	아니오	공격수	아니오	맑음	패배
홈	7시	예	미드필더	아니오	맑음	승리
어웨이	7시	예	미드필더	아니오	비	승리
홈	9시	아니오	공격수	아니오	비	패배

■ 데이터

◉ 다음 주 경기는 어웨이, **9**시 시작, **S**선수가 미드필더로 출전...

장소	시간	K선수 출전	S선수 포지션	R선수 출전	날씨	결과
어웨이	9 시	아니오	미드필더	예	맑음	??

■ 결정 트리

◉ 결정 트리는 트리 구조의 분류기

- 결정 노드 : 단일 특성에 대해 데이터를 테스트
- 말단 노드 : 클래스를 나타냄
- 엣지 : 하나의 특성 값을 분류
- 경로(**Path**) : 최종 분류 결정을 하기 위한 룰(**rule**)들의 논리합(**disjunction**)

■ 결정 트리

- 결정 트리는 뿌리(root) 노드부터 시작해서 특성 값에 따라 적절한 말단 노드에 도착함으로써 새로운 데이터의 클래스를 분류

(1) 어느 특성부터 시작하지? (뿌리 노드)

K선수 출전

예

아니오

(2) 중간 노드들의 순서는?

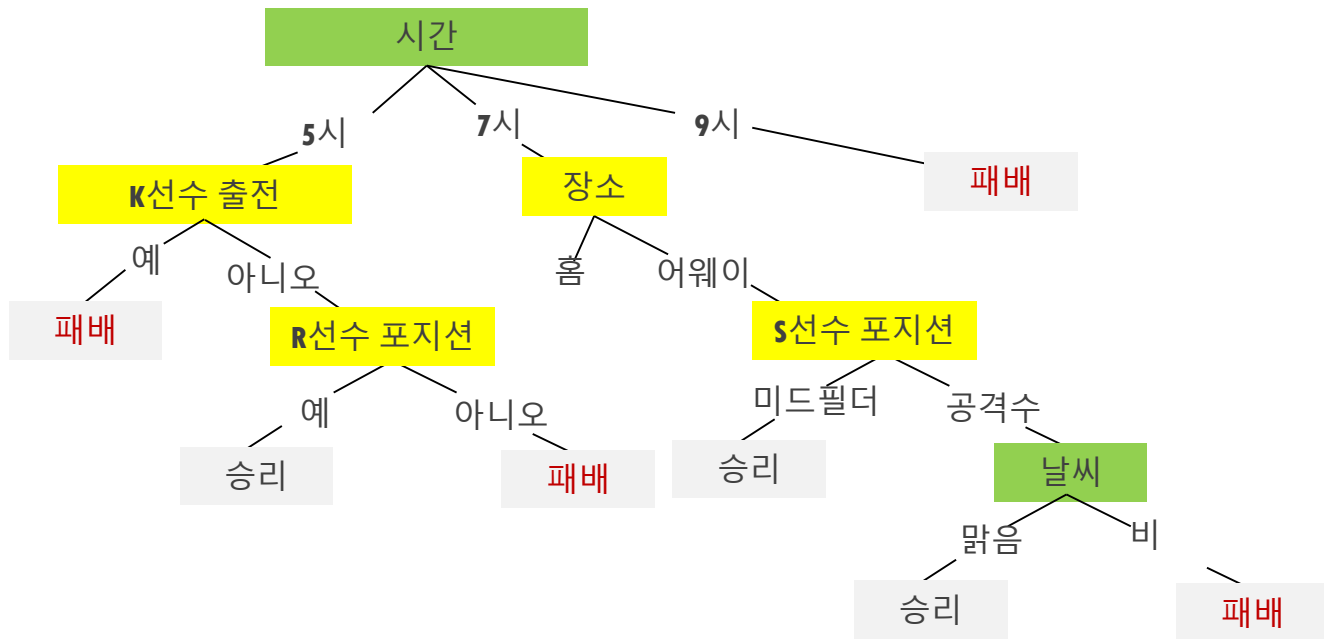
(3) 노드를 얼마나 만들어야 할까?

패배

S선수 포지션

■ 랜덤 결정 트리

- ◉ 트리의 크기가 매우 커질 수 있음
- ◉ 분류의 룰을 이해하기 어려울 수 있음
- ◉ 크기가 큰 트리는 보통 크기가 작은 트리보다 정확도가 떨어짐





2. 결정 트리 구축 원칙

■ 결정 트리 구축 원칙

❖ 각 노드에서 테스트할 특성 선택

- ◉ 분류할 때 가장 유용한 특성 순서대로 선택

❖ 정보 획득량

- ◉ 각 특성들이 훈련 예제들을 얼마나 잘 분류할 수 있는가를 측정
- ◉ 정보 획득량은 트리 구축 과정에서 테스트할 후보 특성의 순서를 결정할 때 사용

■ 결정 트리 구축 원칙

❖ 엔트로피

- ◉ 확률 변수의 불확실성을 수치로 나타냄
- ◉ 출력이 두 개(앞/뒤)인 확률 변수 **S**의 엔트로피 **E(S)**

- $E(S) = -p(F)\log_2 p(F) - p(B)\log_2 p(B)$

- 예1> 동전을 던질 때 앞면과 뒷면이 나올 확률이 같을 때 엔트로피

$$E(S) = -(1/2)\log(1/2) - (1/2)\log(1/2) = 1/2 + 1/2 = 1$$

- 예2> 동전을 던질 때 앞면이 나올 확률이 1/4 뒷면이 나올 확률이 3/4 일 때 엔트로피

$$E(S) = -(1/4)\log(1/4) - (3/4)\log(3/4) = 2/4 + 3/4 \times 0.42 = 0.31$$

■ 결정 트리 구축 원칙

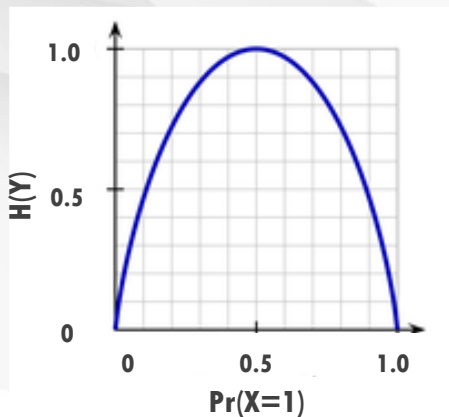
❖ 결정 트리 구축 원칙 예제

- ◎ **S**가 **25**개 예제를 가지고 있고 그 중 **15**개가 **positive** 예제,
10개가 **negative** 예제인 경우 [**15+**, **10-**], 분류에 대한 **S**의 엔트로피는 다음과 같음

- $E(S) = - p(P)\log_2 p(P) - p(N)\log_2 p(N)$
- $E(S) = - (15/25) \log_2(15/25) - (10/25) \log_2 (10/25)$

■ 엔트로피의 직관적인 이해

- 엔트로피가 0일 때, 출력은 매우 확실한 상태
- 엔트로피는 출력에 대해서 아무런 정보를 갖고 있지 않을 때 (어떠한 출력값도 동등한 확률로 나옴) 최고 값을 가짐
 - 예> 동전 던지기의 앞/뒤 확률이 같은 경우



클래스가 두 개인 문제에서
하나의 클래스의 발생 확률에 대한
엔트로피의 변화

■ 정보 획득량

❖ 정보 획득량 = 엔트로피 (불확실성)의 감소량

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

- ◉ **Values(A)**는 특성 **A**(예> 날씨)의 모든 가능한 값(예>맑음, 비)들의 집합이고,
 S_v 는 데이터 집합 **S**에서 특성 **A**의 값이 **v**인 **S**의 부분집합 $S_v = \{s \text{ in } S \mid A(s) = v\}$
- ◉ 정보 획득량 식의 첫 번째 항은 데이터 집합 **S**의 엔트로피 값

■ 정보 획득량

❖ 정보 획득량 = 엔트로피 (불확실성)의 감소량

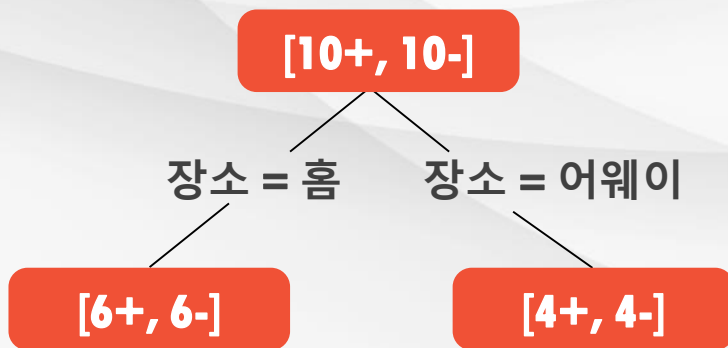
$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

- ◉ 두 번째 항은 특성 **A**를 사용하여 **S**를 분리했을 때, 예상되는 **S**의 엔트로피의 값
- ◉ 정보 획득량 : 특성 **A**를 사용하여 **S**를 분리했을 때, 예상되는 엔트로피의 감소량
- ◉ 또는, 특성 **A**를 알 때, 데이터 집합 **S**의 원소를 인코딩 할 때 감소되는 비트의 크기

■ 정보 획득량 예제

◉ 분리하기 전, **s**의 엔트로피

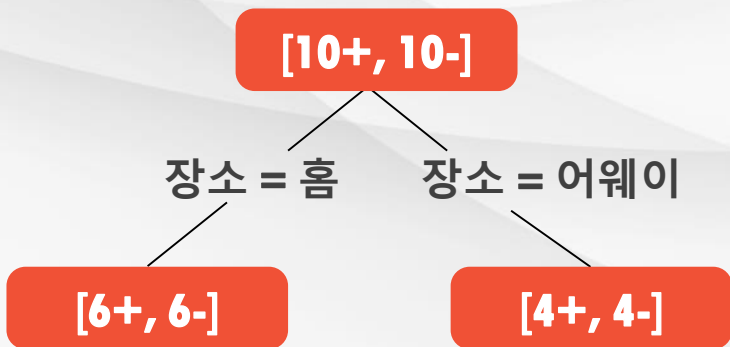
- $H(10/20, 10/20)$
 $= - 10/20 \log(10/20) - 10/20 \log(10/20) = 1$



■ 정보 획득량 예제

◉ 장소 특성을 사용하여 두 개의 부분집합으로 분리

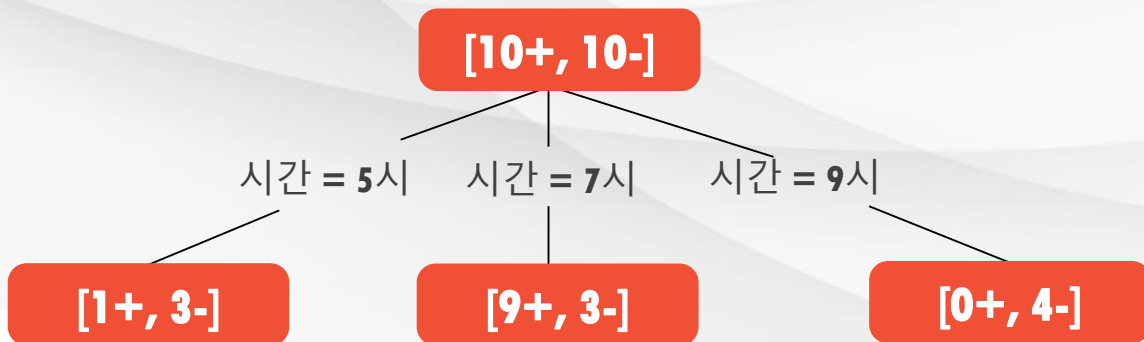
- 첫 번째 집합의 엔트로피 : $H(\text{홈}) = - 6/12 \log(6/12) - 6/12 \log(6/12) = 1$
- 두 번째 집합의 엔트로피 : $H(\text{어웨이}) = - 4/8 \log(6/8) - 4/8 \log(4/8) = 1$
- 분리한 뒤 S의 엔트로피 : $12/20 * H(\text{홈}) + 8/20 * H(\text{어웨이}) = 1$



■ 정보 획득량 예제

◉ 시간 특성을 사용하여 세 개의 부분집합으로 분리

- 첫 번째 집합의 엔트로피 : $H(5\text{시}) = -1/4 \log(1/4) - 3/4 \log(3/4) = 0.31$
- 두 번째 집합의 엔트로피 : $H(7\text{시}) = -9/12 \log(9/12) - 3/12 \log(3/12) = 0.34$
- 세 번째 집합의 엔트로피 : $H(9\text{시}) = -0/4 \log(0/4) - 4/4 \log(4/4) = 0$



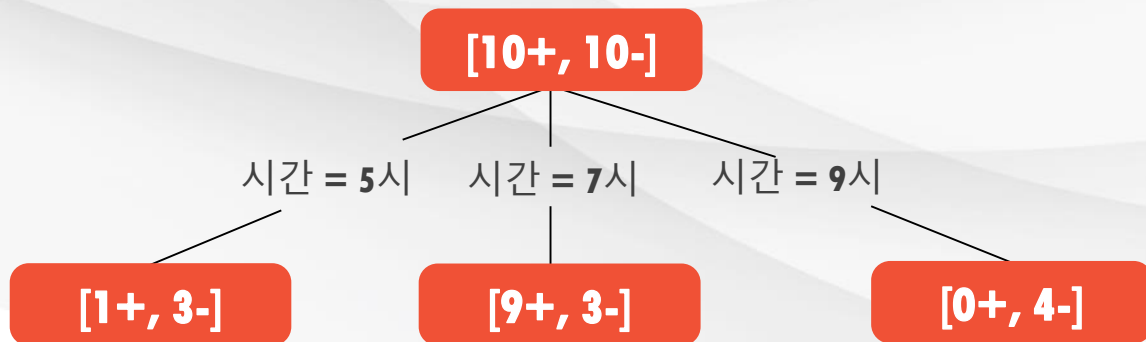
■ 정보 획득량 예제

◉ 분리한 뒤 **S**의 엔트로피

$$\blacksquare \quad 4/20 * H(1/4, 3/4) + 12/20 * H(9/12, 3/12) + 4/20 * H(0/4, 4/4) = 0.65$$

◉ 정보 획득량

$$\blacksquare \quad 1 - 0.65 = 0.35$$



■ 결정 트리 구축 원칙

❖ 어떤 노드가 먼저 선택되어야 할까?

- ◉ 시간 특성 값을 알 때가, 장소 특성 값을 알 때보다 더 많은 정보를 줌
- ◉ 시간 특성 값을 장소 특성 값보다 먼저 테스트 해야 함
- ◉ 이와 같이, 다른 특성들에 대해서 정보 획득량을 계산
- ◉ 결정 트리의 각 노드에서 가장 큰 정보 획득량을 갖는 특성을 선택

❖ 얼마나 노드를 만들어야 할까? (stopping rule)

- ◉ 모든 특성들이 트리의 경로에 모두 포함되어 있는 경우
- ◉ 말단 노드와 연관되어 있는 모든 훈련 예제들이 같은 클래스에 해당하는 경우

즉, 엔트로피가 0일 때

■ 결정 트리 구축 원칙

❖ 가장 적합한 트리는?

- ◉ 모델을 구축하기 위해 **'bias'** 가 필요

예> 크기가 가장 작은 트리를 선호, 또는 깊이가 가장 높은 트리, 노드 개수가 가장 많은 트리

- ◉ 어떠한 트리가 새로운 데이터를 가장 잘 분류할 수 있을까?

■ 결정 트리 구축 원칙

❖ Occam의 면도날

- ◉ 면도날은 필요하지 않은 가설을 잘라내 버린다는 비유
- ◉ 데이터의 특성과 부합하는 가장 간단한 가설을 선호
- ◉ 모든 것이 똑같은 조건일 때, 가장 쉬운 설명이 가장 옳은 것
- ◉ 과학자의 가설 선호도
- ◉ 많은 데이터를 설명할 수 있으면서 가장 간단한 가설을 선택

예> $F=ma$

“

결정 트리는 데이터의 특성과 부합하면서도
가능한 가장 작은 트리여야 함

”

■ 결정 트리 분류의 특징

❖ 장점

- ◉ 다른 분류 방법에 비해서 상대적으로 빠름 (분류 시 계산량이 적음)
- ◉ 간단하고 모델 구축 원리를 이해하기 쉬움
- ◉ 모델의 분류 룰(**rule**)을 사람이 직관적으로 이해하기 쉬움
 - 어느 특성이 분류에 가장 중요한지 명확히 나타냄
- ◉ 다른 모델들에 비해서 더 좋은 성능 나타낼 때가 있음

■ 결정 트리 분류의 특징

❖ 단점

- ◉ 연속적인 특성 값을 갖는 데이터에 적합하지 않음
- ◉ 클래스의 개수가 많고 데이터가 적을 때 성능이 좋지 않음
- ◉ 훈련과정에서 계산량이 많음
 - 트리의 노드 선택 순서를 정하기 위해
모든 특성의 정보 획득량을 계산한 뒤, 정렬해야 함

■ 특성이 연속된 값을 가질 때 결정 트리 구축

❖ 연속된 값을 갖는 특성

- ◉ 특성 **A**의 연속된 값을 이산적인 구간의 집합으로 나눔
- ◉ 새로운 **boolean** 특성 **A_c**의 값을 임계치 **c**에 따라 설정

$$A_c = \begin{cases} true & \text{if } A_c < c \\ false & \text{otherwise} \end{cases}$$

예> 새로운 특성: 섭씨 온도

❖ 어떻게 임계치 **c**를 정할까?

- ◉ 데이터를 특성의 값에 따라서 정렬시킴
- ◉ 클래스가 다르면서 인접한 두 개의 데이터를 참음
 - 임계치의 집합이 생성됨
- ◉ 정보 획득량이 가장 많은 임계치를 선택



3. Bagging

■ Bootstrap

- ◉ 데이터의 개수가 부족할 때 사용할 수 있음

- 예> 날씨가 맑은 날 A팀의 평균 승리 확률은?

- ◉ 원본 데이터 집합 \mathbf{X} 가 있을 때, $\mathbf{X}=(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$, 다음 과정을 m 번 반복

- 크기가 n 인 데이터 집합 \mathbf{X} 에서 복원 추출법을 사용하여 부분 집합 \mathbf{x}_k 생성
- \mathbf{x}_k 에 대해서 구하고자 하는 값 $\hat{\theta}$ * 계산



bootstrap 값 계산

$$\hat{\theta}^* = (\hat{\theta}_1^*, \dots, \hat{\theta}_B^*)$$

■ Bootstrap

- ◉ **Bootstrap** 값들을 사용하여 표준 편차 또는 확신 구간 등 설정 가능
- ◉ **1980**년대부터 컴퓨터가 통계적 용도로 사용되면서 쓰여지기 시작
- ◉ **Bootstrap** 분포는 원본 데이터에서 통계적 변화를 측정하기 위해 주로 사용됨

■ Bagging

- ◉ 1996년 Breiman에 의해 소개됨
- ◉ Bagging : “bootstrap aggregating”
- ◉ 다수의 분류기를 결합하는 앙상블 방법 중 하나
 - 주로 결정 트리 분류기에 많이 쓰임
- ◉ 원본 훈련 데이터 집합 X 가 있을 때, 다음 과정을 m 번 반복
 - X 로부터 **bootstrap** 샘플 X_k 를 얻음
 - X_k 로부터 분류기 C_k 를 훈련

■ Bagging

- ◉ m 개의 분류기를 다음의 방식으로 결합
 - 투표
 - 평균
- ◉ 데이터가 갖는 **high variance** 문제를 해결할 수 있음



학습정리

지금까지 [결정 트리]에 대해서 살펴보았습니다.

의사 결정 트리와 분류 문제

결정 트리는 트리 구조의 분류기

노드 : 단일 특성에 대해 데이터를 테스트

말단 노드 : 클래스를 나타냄

엣지: 하나의 특성 값을 분류

경로: 최종 분류 결정을 하기 위한 룰들의 논리합

결정 트리 구축 원칙

정보 획득량은 결정 트리 구축 과정에서 테스트할 후보 특성의 순서를 결정할 때 사용

Bagging

Bagging은 다수의 분류기를 결합하는 앙상블 방법 중 하나
k개의 bootstrap 샘플 $X_{1...k}$ 를 훈련시켜 얻은 k개의 분류기를
투표, 평균 등의 방법을 사용하여 결합