

의사 결정 트리

I. 의사 결정 트리(Decision trees)

II. 예제 기반 의사 결정 트리 학습

III. 엔트로피와 속성 선택

식당에서 자리 대기 문제

Problem 식당에서 빈 좌석이 나올 때 까지 기다릴지 결정하기

Output: 부울 변수 *WillWait* (빈 좌석 기다린다면 true).

Input: 이산적인 값을 갖는 열 개의 속성 벡터

1. *Alternate*: 주변에 적합한 다른 식당이 있는지
2. *Bar*: 식당에 편안한 바 공간이 있는지
3. *Fri/Sat*: 금요일이거나 토요일이면 true
4. *Hungry*: 지금 당장 배가 몹시 고프지
5. *Patrons*: 식당에 손님이 많은지 (values: *None*, *Some*, *Full*)
6. *Price*: 식당의 음식 가격대가 비싼지 (\$, \$\$, \$\$\$).
7. *Raining*: 밖에 비가 오는지
8. *Reservation*: 예약을 해 두었는지
9. *Type*: 식당 종류 (*French*, *Italian*, *Thai*, *burger*).
10. *WaitEstimate*: 호스트 예상 대기시간: 0-10분, 10-30분, 30-60분, 60분 이상

$2^6 \times 3^2 \times 4^2 = 9,216$ 가지의 속성 값 조합이 존재

학습 예제

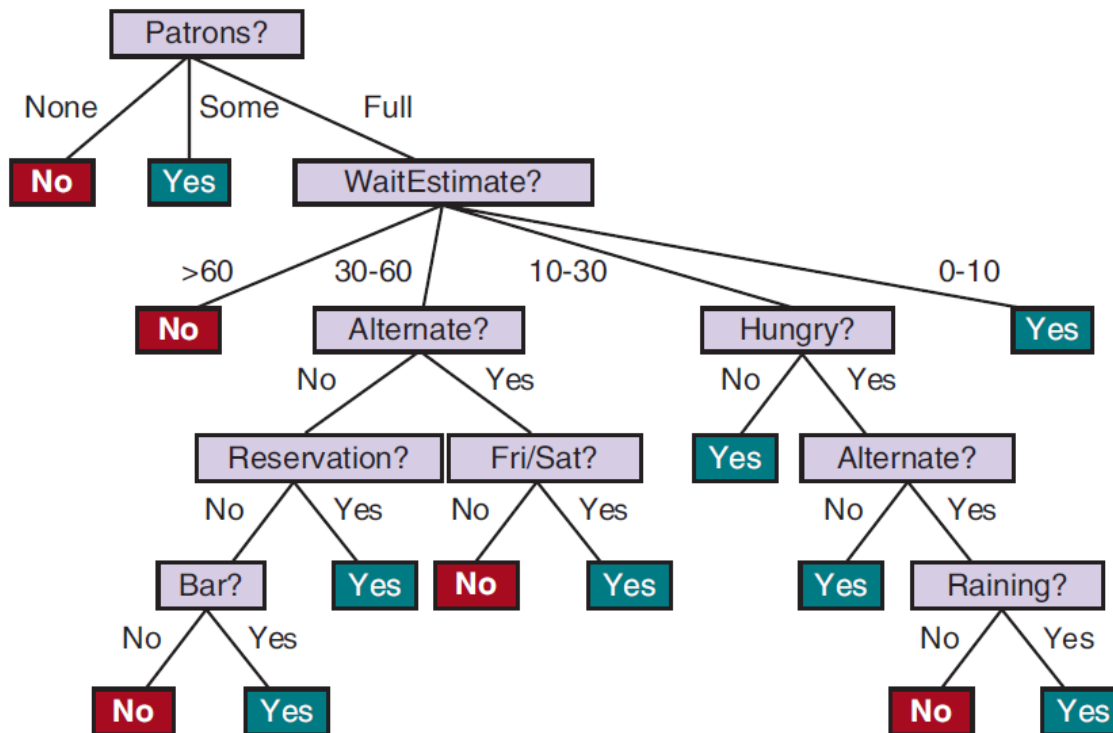
Example	Input Attributes										Output
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	<i>WillWait</i>
x₁	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>Yes</i>	<i>Some</i>	<i>\$\$\$</i>	<i>No</i>	<i>Yes</i>	<i>French</i>	<i>0-10</i>	<i>y₁ = Yes</i>
x₂	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>Yes</i>	<i>Full</i>	<i>\$</i>	<i>No</i>	<i>No</i>	<i>Thai</i>	<i>30-60</i>	<i>y₂ = No</i>
x₃	<i>No</i>	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>Some</i>	<i>\$</i>	<i>No</i>	<i>No</i>	<i>Burger</i>	<i>0-10</i>	<i>y₃ = Yes</i>
x₄	<i>Yes</i>	<i>No</i>	<i>Yes</i>	<i>Yes</i>	<i>Full</i>	<i>\$</i>	<i>Yes</i>	<i>No</i>	<i>Thai</i>	<i>10-30</i>	<i>y₄ = Yes</i>
x₅	<i>Yes</i>	<i>No</i>	<i>Yes</i>	<i>No</i>	<i>Full</i>	<i>\$\$\$</i>	<i>No</i>	<i>Yes</i>	<i>French</i>	<i>>60</i>	<i>y₅ = No</i>
x₆	<i>No</i>	<i>Yes</i>	<i>No</i>	<i>Yes</i>	<i>Some</i>	<i>\$\$</i>	<i>Yes</i>	<i>Yes</i>	<i>Italian</i>	<i>0-10</i>	<i>y₆ = Yes</i>
x₇	<i>No</i>	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>None</i>	<i>\$</i>	<i>Yes</i>	<i>No</i>	<i>Burger</i>	<i>0-10</i>	<i>y₇ = No</i>
x₈	<i>No</i>	<i>No</i>	<i>No</i>	<i>Yes</i>	<i>Some</i>	<i>\$\$</i>	<i>Yes</i>	<i>Yes</i>	<i>Thai</i>	<i>0-10</i>	<i>y₈ = Yes</i>
x₉	<i>No</i>	<i>Yes</i>	<i>Yes</i>	<i>No</i>	<i>Full</i>	<i>\$</i>	<i>Yes</i>	<i>No</i>	<i>Burger</i>	<i>>60</i>	<i>y₉ = No</i>
x₁₀	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Full</i>	<i>\$\$\$</i>	<i>No</i>	<i>Yes</i>	<i>Italian</i>	<i>10-30</i>	<i>y₁₀ = No</i>
x₁₁	<i>No</i>	<i>No</i>	<i>No</i>	<i>No</i>	<i>None</i>	<i>\$</i>	<i>No</i>	<i>No</i>	<i>Thai</i>	<i>0-10</i>	<i>y₁₁ = No</i>
x₁₂	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Full</i>	<i>\$</i>	<i>No</i>	<i>No</i>	<i>Burger</i>	<i>30-60</i>	<i>y₁₂ = Yes</i>

- ♣ 9,216가지 가능한 예제 중 단 12개의 실 사례만 주어짐
- ♣ 나머지 9,204개의 사례는 최선의 추측으로 채워야 함

I. 의사 결정 트리

의사 결정 트리: 속성 값 집합을 "결정"으로 매핑

- ◆ 루트에서 시작하여 리프까지 내려가는 일련의 테스트를 수행 (리프에서 최종 결과 출력)
- ◆ 리프가 아닌 내부 노드는 속성 값에 대한 테스트에 해당



식당 대기 문제에
대한 의사 결정 트리

T_1

이진 분류

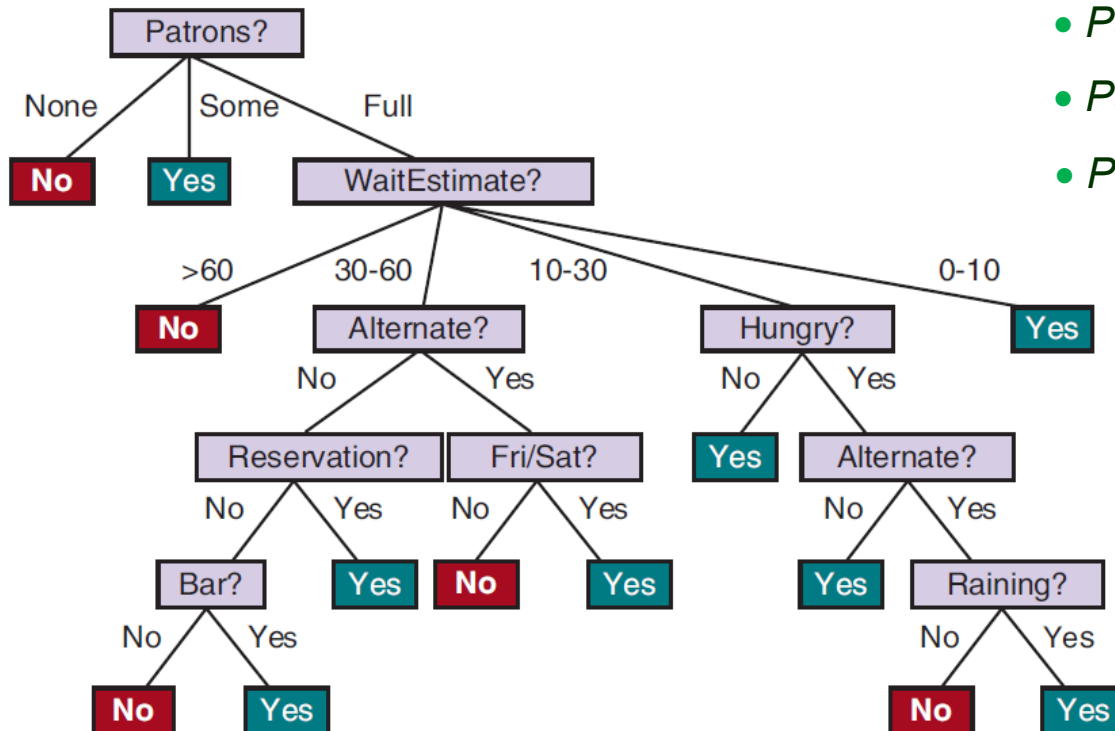
- 입력값은 이산값
- 출력값은 *true* (a positive example)이거나 *false* (a negative example)

Positive examples:

- *Patrons = Some*
- *Patrons = Full* \wedge *WaitEstimate* = 0-10
- *Patrons = Full*
 \wedge *WaitEstimate* = 10-30
 \wedge *Hungry* = Yes
 \wedge *Alternate* = No

Negative examples:

- *Patrons = No*
- *Patrons = Full*
 \wedge *WaitEstimate* > 60



의사 결정 트리의 장점

- ◆ 깔끔하고 간결하며 이해하기 쉬운 결과를 산출
- ◆ 이산 속성 값으로 묘사된 인스턴스의 분류 과정을 간단한 규칙으로 표현
- ◆ 의사 결정 트리 학습 알고리즘은 비교적 효율적임
 - 의사 결정 트리의 크기는 데이터 세트의 크기에 **선형**으로 비례
- ◆ 의사 결정 트리는 종종 새로운 데이터 세트에 처음으로 시도됨
- ♠ 실수값 속성에는 적합하지 않음

의사 결정 트리의 표현력

의사 결정 트리와 동치인 논리적 문장

- ✓ 논리곱 정규형(conjunctive normal form, CNF)
논리합 절들이 논리곱으로 연결되어 있는 논리식 (합의 곱)
- ✓ 논리합 정규형(disjunctive normal form, DNF)
논리곱 절들이 논리합으로 연결되어 있는 논리식 (곱의 합)

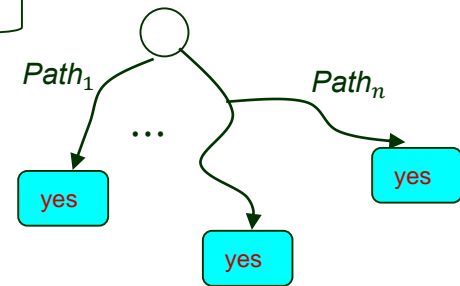
$$Output \Leftrightarrow Path_1 \vee Path_2 \vee \dots$$

여기서

$$Path_i = (A_{i1} = v_{i1} \wedge A_{i2} = v_{i2} \wedge \dots)$$

attribute value

논리합 정규형
(Disjunctive NF)



명제 논리에 속하는 어떤 문장이든 논리합 정규형(DNF)으로 변환 가능



명제 논리에서의 어떤 문장이든 의사 결정 트리로 표현 가능

II. 예제 기반 의사 결정 트리 학습

Problem a) 학습용 예제와 일치하면서(예를 들어, 식당 대기),
b) 가능한 작은 트리 찾기

- 일반적으로 일치하면서 가장 작은 트리를 찾는 것은 계산적으로 *어려움*
- 일부 간단한 휴리스틱을 사용하여 *적당한 수준의 적합한 트리* 구성 가능

탐욕적 분할 정보 전략:

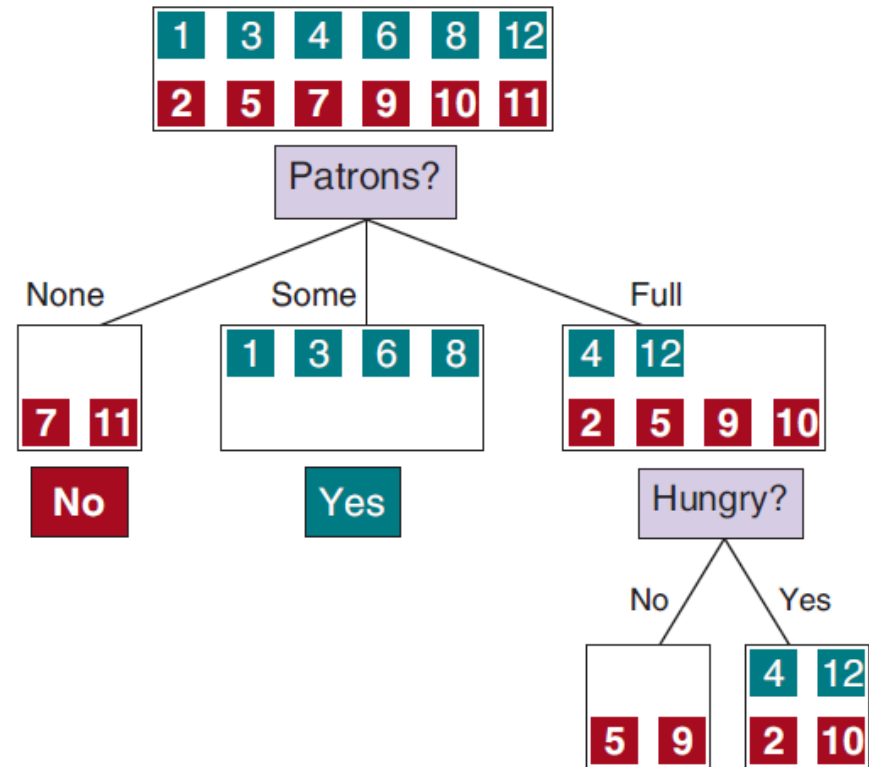
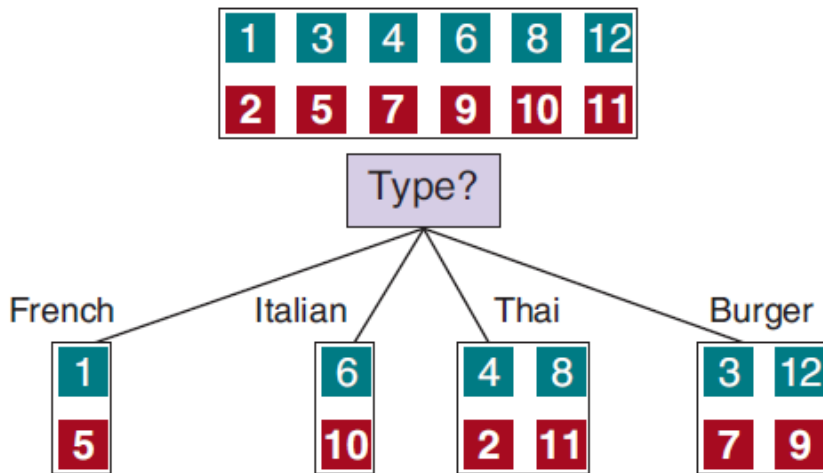
- **가장 중요한 속성**을 먼저 테스트

분류에 가장 큰 영향을
미치는 속성

- 다음으로 더 작은 하위 문제를
재귀적으로 해결

Example	Input Attributes										Output
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
x ₁	Yes	No	No	Yes	Some	\$\$\$	No	Yes	French	0-10	y ₁ = Yes
x ₂	Yes	No	No	Yes	Full	\$	No	No	Thai	30-60	y ₂ = No
x ₃	No	Yes	No	No	Some	\$	No	No	Burger	0-10	y ₃ = Yes
x ₄	Yes	No	Yes	Yes	Full	\$	Yes	No	Thai	10-30	y ₄ = Yes
x ₅	Yes	No	Yes	No	Full	\$\$\$	No	Yes	French	>60	y ₅ = No
x ₆	No	Yes	No	Yes	Some	\$\$	Yes	Yes	Italian	0-10	y ₆ = Yes
x ₇	No	Yes	No	No	None	\$	Yes	No	Burger	0-10	y ₇ = No
x ₈	No	No	No	Yes	Some	\$\$	Yes	Yes	Thai	0-10	y ₈ = Yes
x ₉	No	Yes	Yes	No	Full	\$	Yes	No	Burger	>60	y ₉ = No
x ₁₀	Yes	Yes	Yes	Yes	Full	\$\$\$	No	Yes	Italian	10-30	y ₁₀ = No
x ₁₁	No	No	No	No	None	\$	No	No	Thai	0-10	y ₁₁ = No
x ₁₂	Yes	Yes	Yes	Yes	Full	\$	No	No	Burger	30-60	y ₁₂ = Yes

나쁜 속성과 좋은 속성



나쁜 속성: *Type*:

- ♠ 네 가지 결과
- ♠ 각각 긍정적 예와 부정적 예의 수가 동일

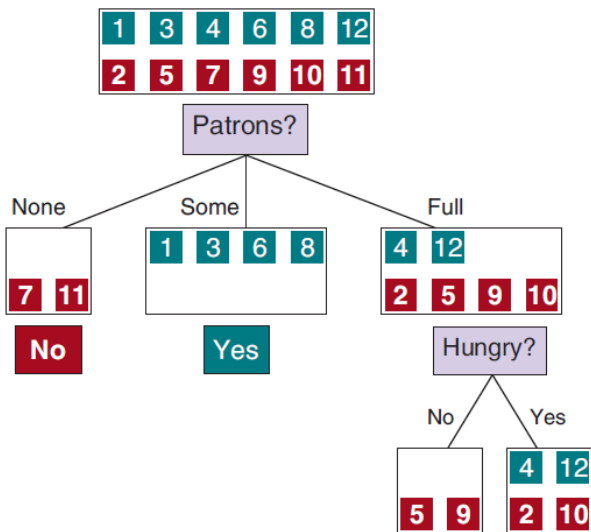
좋은 속성: *Patrons* 과 *Hungry*:

- ♦ 긍정적인 예와 부정적 예를 효과적으로 분리

4가지 사례

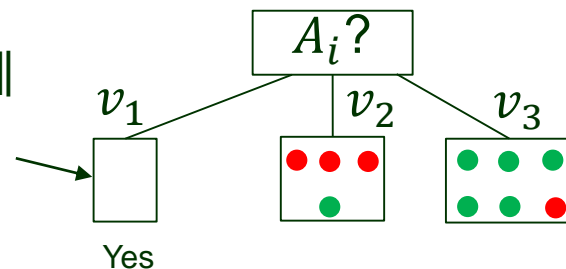
사례 1. 모두가 긍정적이거나 모두가 부정적인 예제
추가 고려사항 없음.

사례 2. 긍정적인 예와 부정적인 예가 혼합된 경우
둘 사이를 구분할 수 있는 최상의 속성을 선택

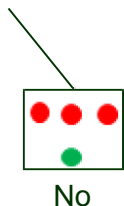


사례 3. 남은 예시가 없는 경우

부모($A_i?$)를 구성하는 예제
세트에서 가장 많이 나온
출력 값(예: ●)을 반환



사례 4. 남은 속성이 없지만 긍정적인 예와
부정적인 예가 모두 있는 경우



해당 사례에서 가장 많이
나온 출력 값(다수결)

의사 결정 트리 학습 알고리즘

function LEARN-DECISION-TREE(*examples*, *attributes*, *parent_examples*) **returns** a tree

if *examples* is empty **then return** PLURALITY-VALUE(*parent_examples*) // 사례 3

else if all *examples* have the same classification **then return** the classification // 사례 1

else if *attributes* is empty **then return** PLURALITY-VALUE(*examples*) // 사례 4

else // 사례 2

$A \leftarrow \operatorname{argmax}_{a \in \text{attributes}} \text{IMPORTANCE}(a, \text{examples})$

tree \leftarrow a new decision tree with root test *A*

for each value *v* of *A* **do**

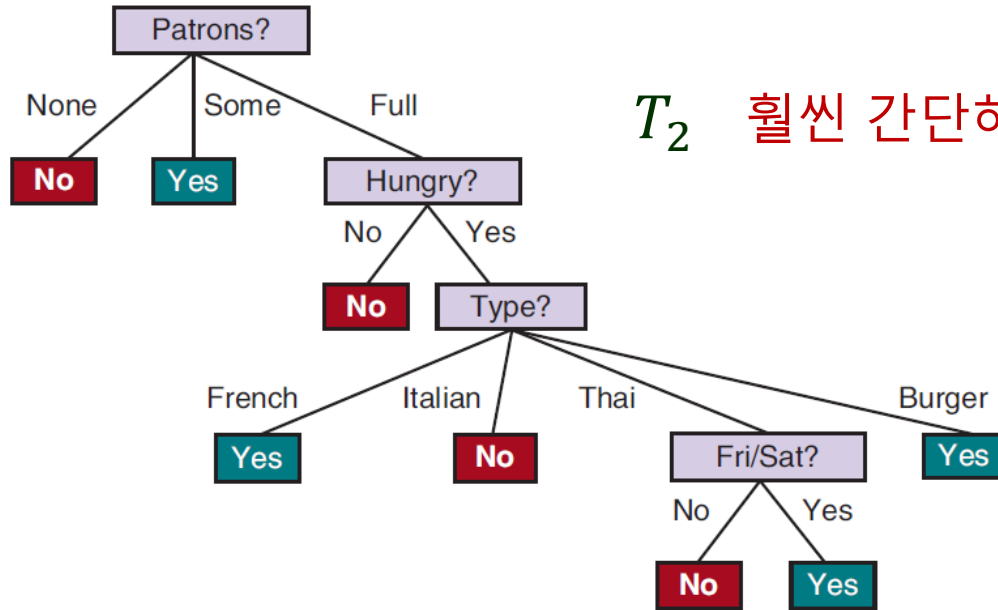
$\text{exs} \leftarrow \{e : e \in \text{examples} \text{ and } e.A = v\}$

subtree \leftarrow LEARN-DECISION-TREE(*exs*, *attributes* − *A*, *examples*)

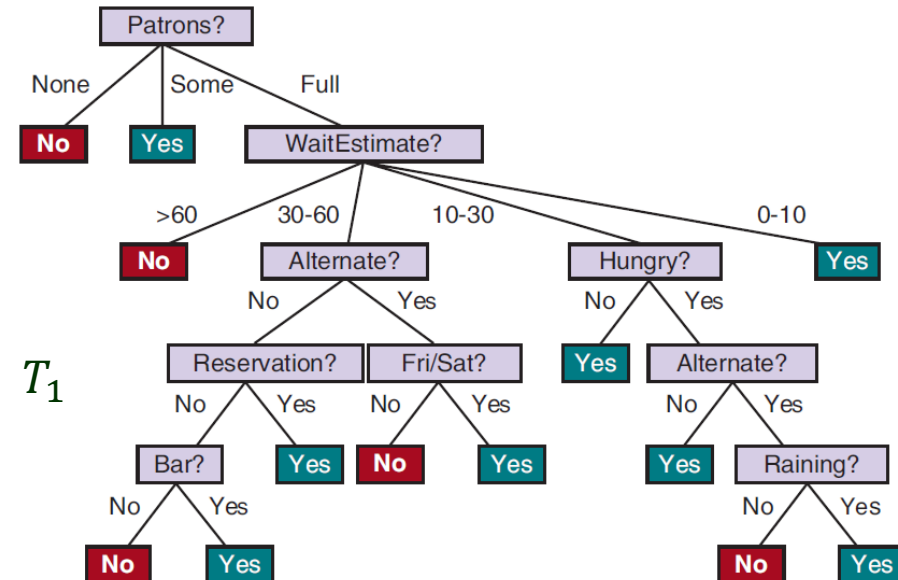
add a branch to *tree* with label (*A* = *v*) and subtree *subtree*

return *tree*

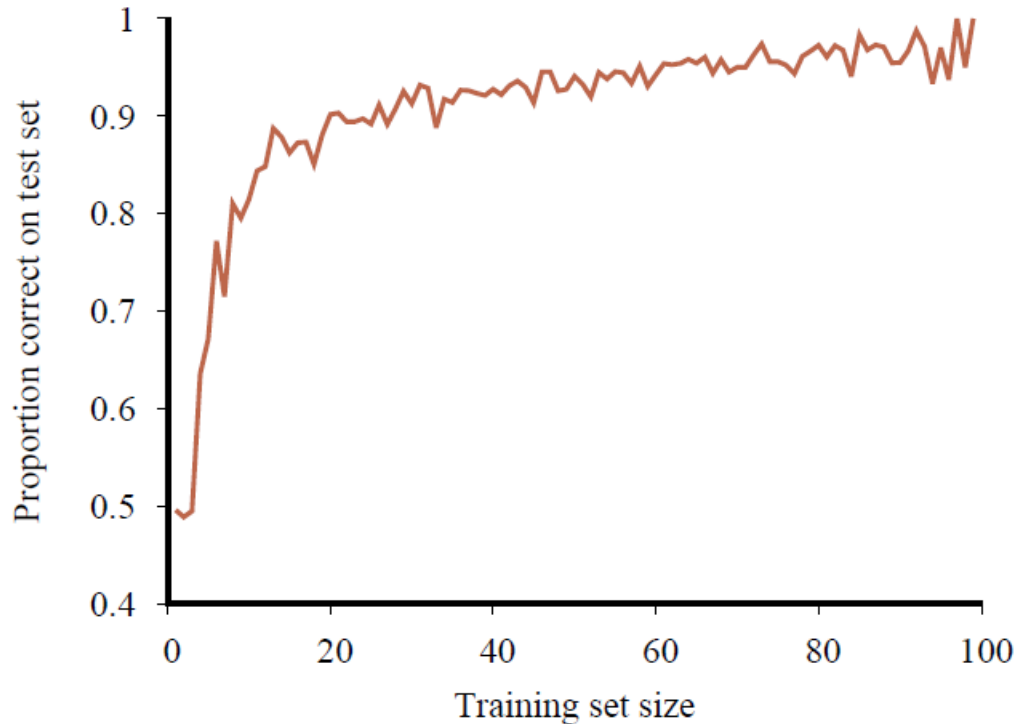
학습 결과 트리



원본 의사 결정 트리 T_1



학습 커브



“행복 그래프”

- 식당 문제에 관해 무작위로 뽑은 100개의 사례
- 무작위로 섞은 후 학습 세트와 테스트 세트로 분리
 - ◆ 학습 세트 크기 1에서 99까지 확장. 각 크기마다 20번씩 재분할
 - ◆ 각 크기 별 20번의 학습 결과를 평균하여 성능 측정

III. 엔트로피와 속성 선택

속성의 중요성은 **엔트로피** 기반으로 정의되는 정보 이득의 개념을 사용하여 측정

확률 변수의 불확실성 측정 지표:
정보가 많을수록 엔트로피가 낮아짐

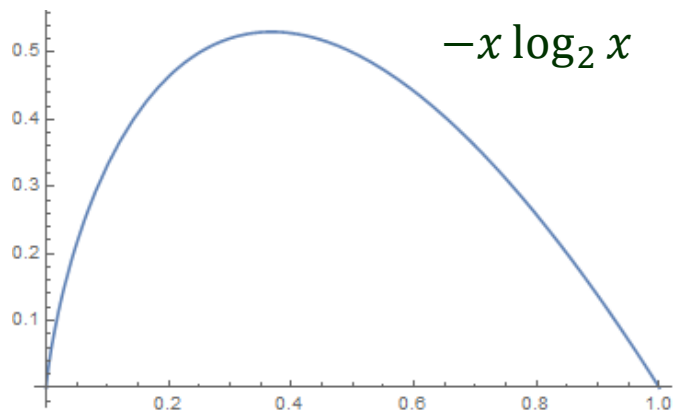
- 공평한 동전은 던질 때 앞면과 뒷면이 나올 확률이 동일하므로 1 비트의 엔트로피를 가짐
- 4면체 주사위는 선택할 수 있는 동등한 선택지가 2^2 개 있기 때문에 엔트로피가 2 비트
- 99%의 확률로 앞면이 나오는 불공정한 동전의 엔트로피 측정값은 거의 0에 가까움

엔트로피

값이 v_k 인 확률변수 V 는 확률 $P(v_k)$ 를 가짐

V 의 엔트로피:

$$\begin{aligned} H(V) &= \sum_k P(v_k) \log_2 \frac{1}{P(v_k)} \\ &= - \sum_k P(v_k) \log_2 P(v_k) \end{aligned}$$



엔트로피는 무작위 시행의 결과를 식별함으로써 전달되는 **정보의 기대량**을 측정

엔트로피 사례

- 공평한 동전 던지기

$$H(\text{Fair}) = -(0.5 \log_2 0.5 + 0.5 \log_2 0.5) = 1 \text{ (bit)}$$

*최대치*의 의외성이 존재하는 사례

- A coin with known outcome 시행이 끝나 결과가 알려진 경우(e.g., head)

$$H(\text{Head}) = -1 \cdot \log_2 1 = 0 \text{ (bit)}$$

불확실성이 전혀 존재하지 않음 – *정보가 없음*

- 99%의 확률로 앞면이 나오도록 조작된 동전

$$H(\text{Loaded}) = -(0.99 \log_2 0.99 + 0.01 \log_2 0.01) \approx 0.08 \text{ (bits)}$$

- 공정한 4면이 있는 주사위

$$H(\text{Die4}) = -(4 \cdot 0.25 \log_2 0.25) = 2 \text{ (bits)}$$

부울 변수의 엔트로피

q 의 확률로 참(True)인 부울 확률 변수

$$B(q) = -(q \log_2 q + (1 - q) \log_2 (1 - q)) \implies B(q) = B(1 - q)$$

$$H(\text{Loaded}) = B(0.99) \approx 0.08$$

- 학습 세트는 p 개의 긍정 샘플과 n 개의 부정 샘플로 구성

$$H(\text{Output}) = B\left(\frac{p}{p + n}\right)$$

$$p = n = 6$$



$$B(0.5) = 1$$

Example	Input Attributes										Output
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
x ₁	Yes	No	No	Yes	Some	\$\$\$	No	Yes	French	0-10	y ₁ = Yes
x ₂	Yes	No	No	Yes	Full	\$	No	No	Thai	30-60	y ₂ = No
x ₃	No	Yes	No	No	Some	\$	No	No	Burger	0-10	y ₃ = Yes
x ₄	Yes	No	Yes	Yes	Full	\$	Yes	No	Thai	10-30	y ₄ = Yes
x ₅	Yes	No	Yes	No	Full	\$\$\$	No	Yes	French	>60	y ₅ = No
x ₆	No	Yes	No	Yes	Some	\$	Yes	Yes	Italian	0-10	y ₆ = Yes
x ₇	No	Yes	No	No	None	\$	Yes	No	Burger	0-10	y ₇ = No
x ₈	No	No	No	Yes	Some	\$	Yes	Yes	Thai	0-10	y ₈ = Yes
x ₉	No	Yes	Yes	No	Full	\$	Yes	No	Burger	>60	y ₉ = No
x ₁₀	Yes	Yes	Yes	Yes	Full	\$\$\$	No	Yes	Italian	10-30	y ₁₀ = No
x ₁₁	No	No	No	No	None	\$	No	No	Thai	0-10	y ₁₁ = No
x ₁₂	Yes	Yes	Yes	Yes	Full	\$	No	No	Burger	30-60	y ₁₂ = Yes

속성 선택

- ♣ 임의의 속성 A 는 d 개의 서로 다른 값을 가짐
- ♣ 속성 A 는 학습 세트 E 를 부분 집합 E_1, \dots, E_d 로 나눔
 - 각 부분 집합 E_k 는 p_k 개의 긍정 예제와 n_k 개의 부정 예제를 가짐

$$p = \sum_{k=1}^d p_k \quad \text{and} \quad n = \sum_{k=1}^d n_k$$

- $B\left(\frac{p_k}{p_k+n_k}\right)$ 비트의 추가 정보를 의미합니다.
- 학습 세트로부터 무작위로 선택된 예제를 고려
 - ◆ 속성 A 값은 확률 $(p_k + n_k)/(p + n)$ 이며 E_k 에 포함
- ♣ 속성 A 를 테스트한 후 남아 있는 기대 엔트로피

$$Remainder(A) = \sum_{k=1}^d \frac{p_k+n_k}{p+n} B\left(\frac{p_k}{p_k+n_k}\right)$$

정보 이득

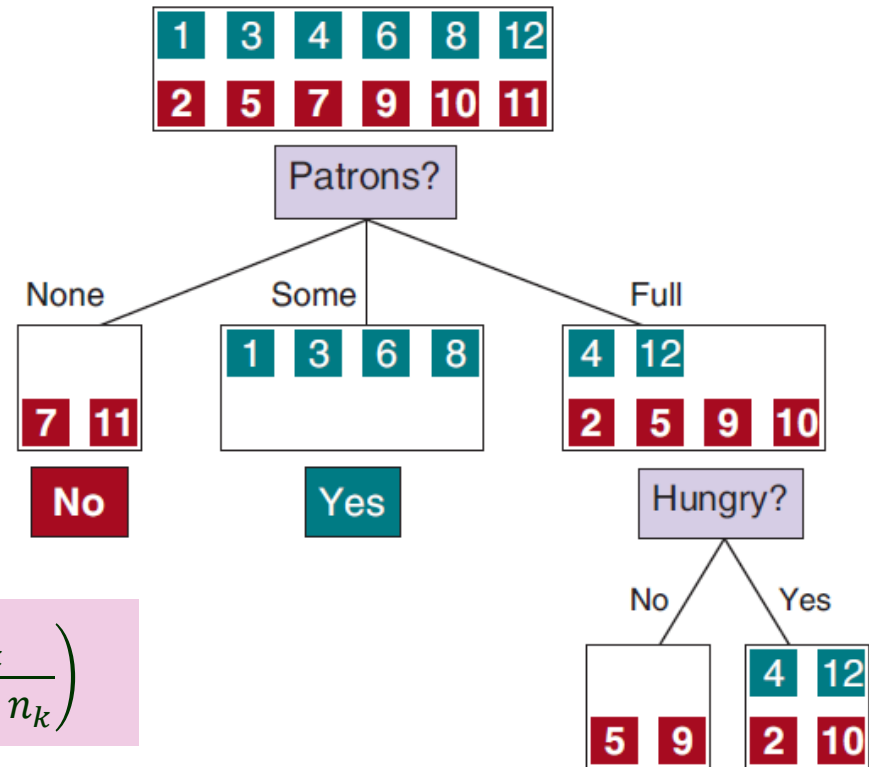
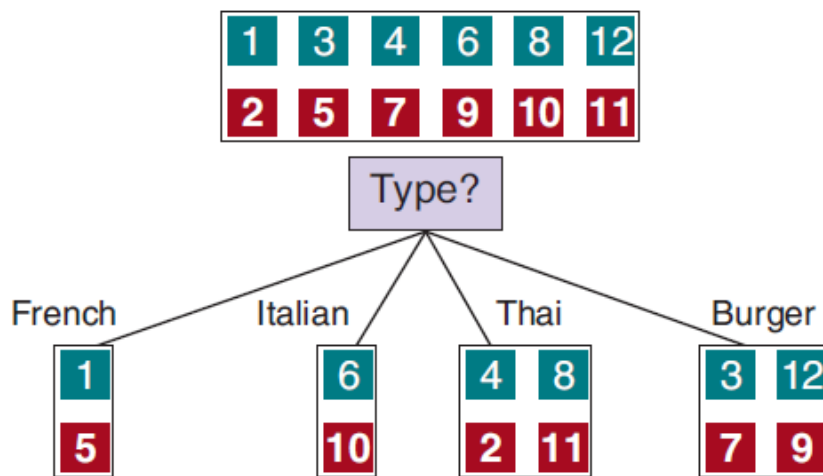
속성 A 에 대한 테스트로 인한 엔트로피 감소의 기대값

$$Gain(A) = B\left(\frac{p}{p+n}\right) - Remainder(A)$$

$$= B\left(\frac{p}{p+n}\right) - \sum_{k=1}^d \frac{p_k + n_k}{p+n} B\left(\frac{p_k}{p_k + n_k}\right)$$

나머지 속성들 중에서 $Gain(A)$ 를 **최대화** 시켜 주는 속성 A 를 선택

정보 이득



$$Gain(A) = B\left(\frac{p}{p+n}\right) - \sum_{k=1}^d \frac{p_k + n_k}{p+n} B\left(\frac{p_k}{p_k + n_k}\right)$$

$$Gain(Type) = B\left(\frac{1}{2}\right) - \left(\frac{2}{12} B\left(\frac{1}{2}\right) + \frac{2}{12} B\left(\frac{1}{2}\right) + \frac{4}{12} B\left(\frac{2}{4}\right) + \frac{4}{12} B\left(\frac{2}{4}\right)\right) = 0 \text{ bits}$$

$$Gain(Patrons) = B\left(\frac{1}{2}\right) - \left(\frac{2}{12} B\left(\frac{0}{2}\right) + \frac{4}{12} B\left(\frac{4}{4}\right) + \frac{6}{12} B\left(\frac{2}{6}\right)\right) \approx 0.541 \text{ bits}$$

