

# 결정트리 학습

## Decision Tree Induction

이 건 명

충북대학교 산업인공지능학과

인공지능 : 튜링 테스트에서 딥러닝까지

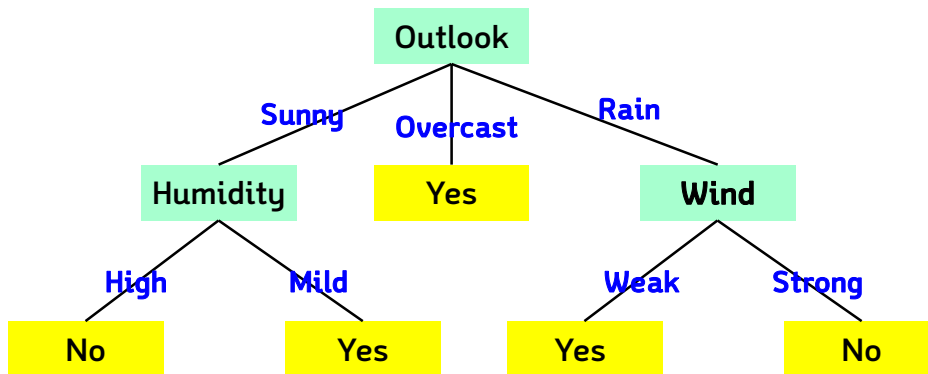
# 학습 내용

- 결정트리 생성 전략을 알아본다.
- 결정트리의 분할속성 선택 방법을 알아본다.
- 결정트리를 이용한 회귀 방법을 알아본다.

# 1. 결정트리

## ❖ 결정 트리(decision tree)

- **트리 형태로 의사결정 지식**을 표현한 것
  - 내부 노드(internal node) : 비교 속성
  - 간선(edge) : 속성 값
  - 단말 노드(terminal node) : 부류(class), 대표값



Day 날짜	Outlook 조망	Temperature 기온	Humidity 습도	Wind 바람	PlayTennis 테니스 여부
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

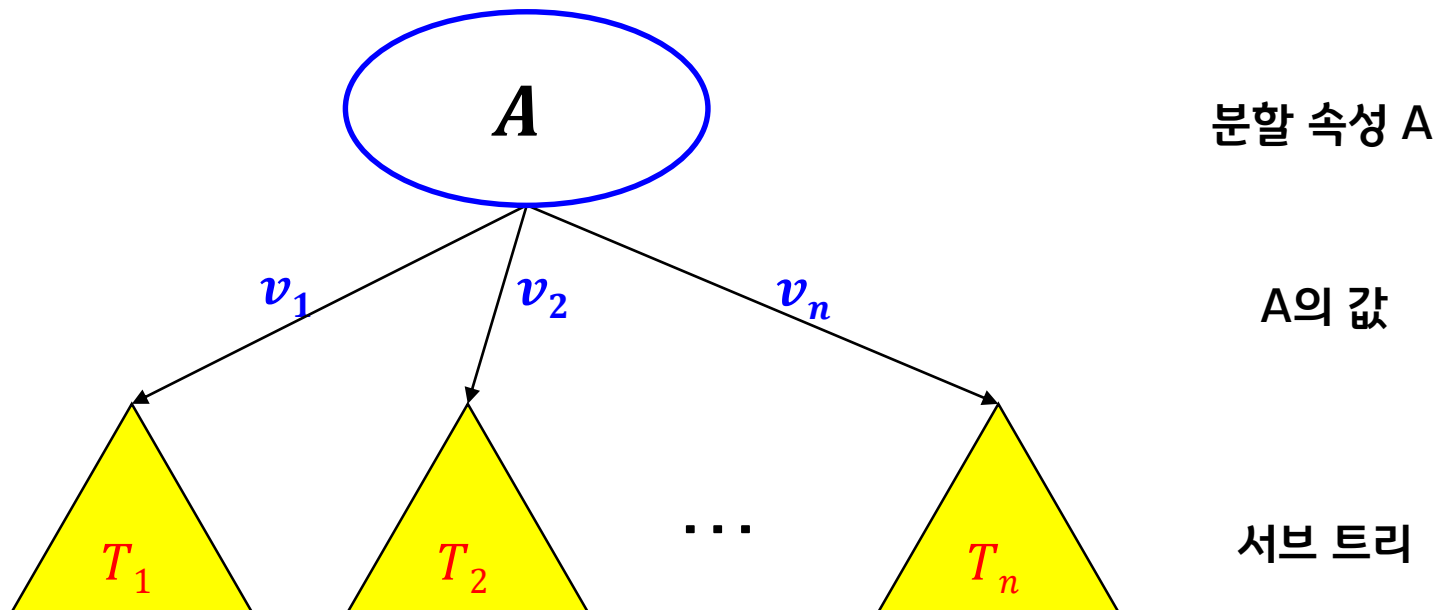
IF Outlook = Sunny AND Humidity = High THEN Answer = No

Outlook 조망	Temperature 기온	Humidity 습도	Wind 바람	PlayTennis 테니스 여부
Sunny	Hot	Mild	Weak	?
Rain	Hot	High	Weak	?

# 결정트리 학습 알고리즘

## ❖ 결정 트리(decision tree) 알고리즘

- 모든 데이터를 포함한 **하나의 노드**로 구성된 **트리**에서 **시작**
- **반복적인 노드 분할** 과정
  1. **분할 속성**(splitting attribute) **선택**
  2. 속성값에 따라 **서브트리**(subtree) **생성**
  3. **데이터**의 속성값에 따른 **분배**

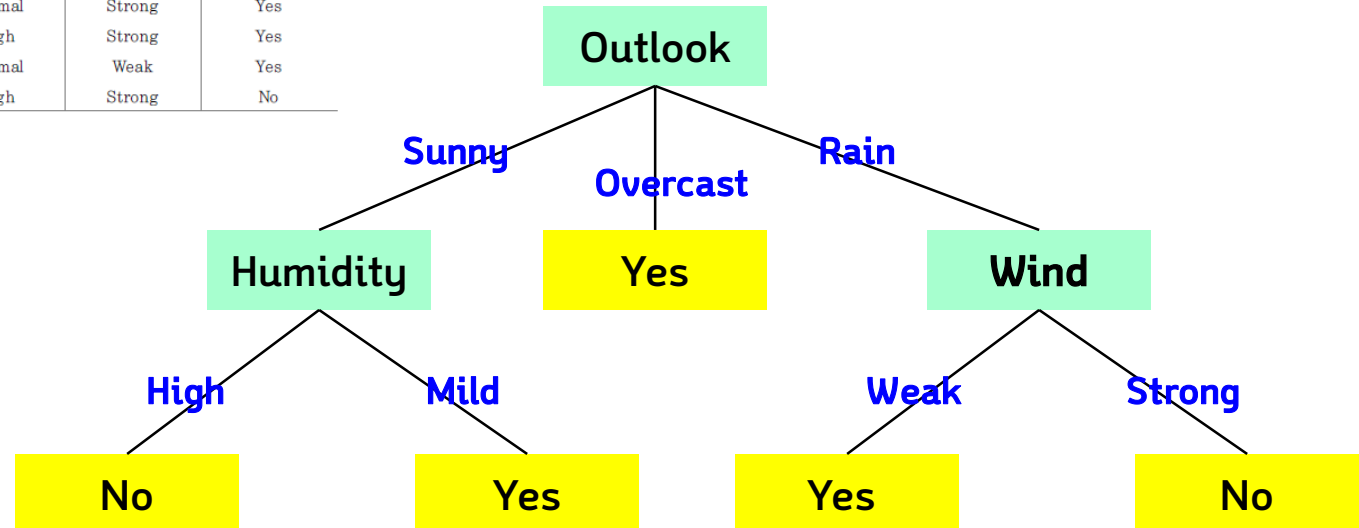


# 결정트리 학습 알고리즘

## ❖ 결정 트리 (decision tree)

### ▪ 간단한 트리

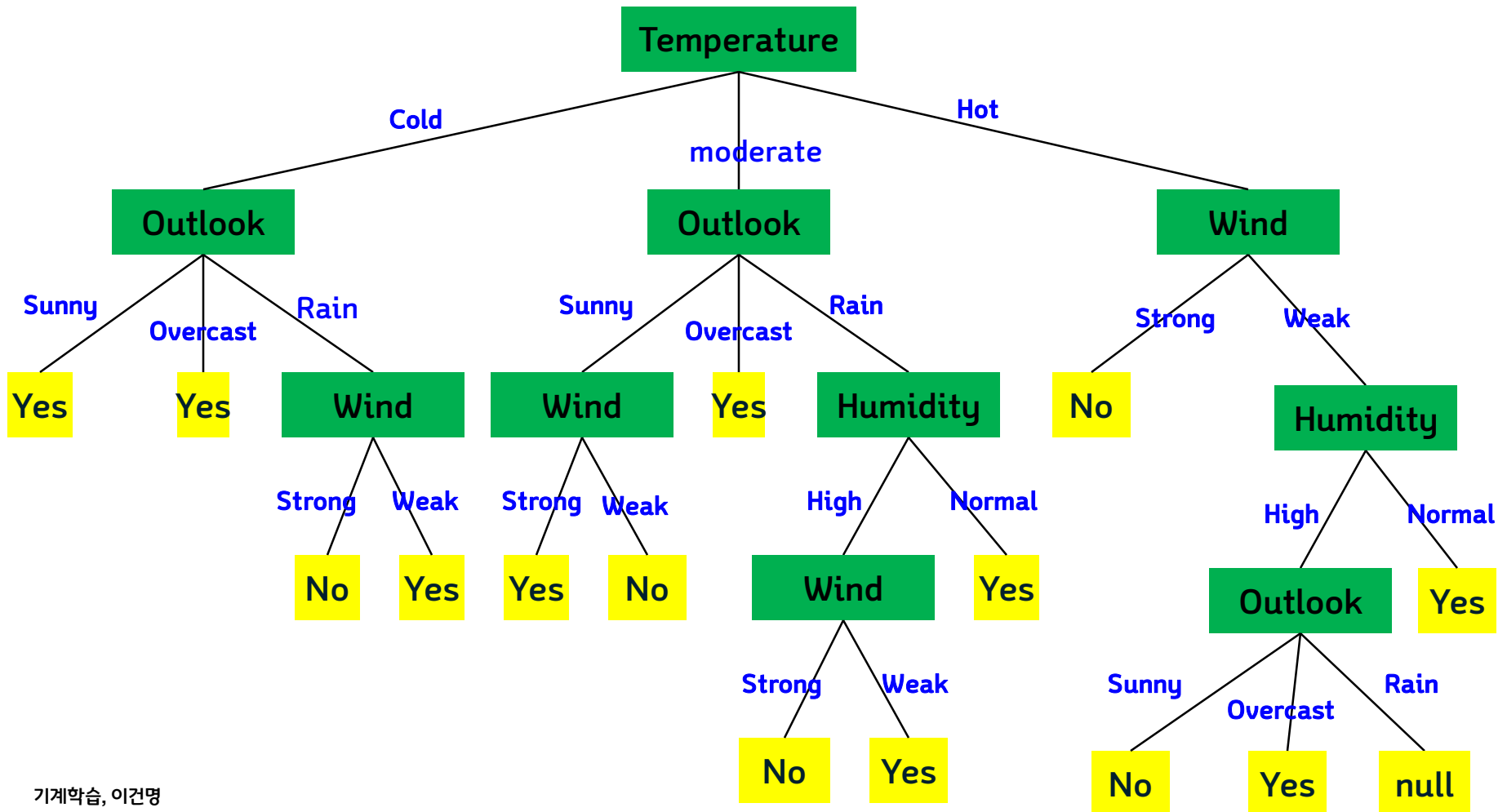
Day 날짜	Outlook 조망	Temperature 기온	Humidity 습도	Wind 바람	PlayTennis 테니스 여부
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No



# 결정트리 학습 알고리즘

## ❖ 결정 트리 (decision tree)

- 복잡한 트리



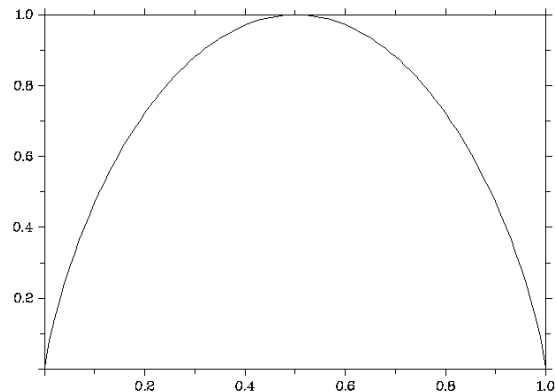
# 결정트리 학습 알고리즘

## ❖ 분할 속성(splitting attribute) 결정

- 어떤 속성을 선택하는 것이 효율적인가
  - 분할 결과가 가능하면 동질적인(pure) 것으로 만드는 속성 선택
- 엔트로피(Entropy)
  - 동질적인 정도의 측정 가능 척도
  - 원래 정보량(amount of information) 측정 목적

$$I = - \sum_c p(c) \log_2 p(c)$$

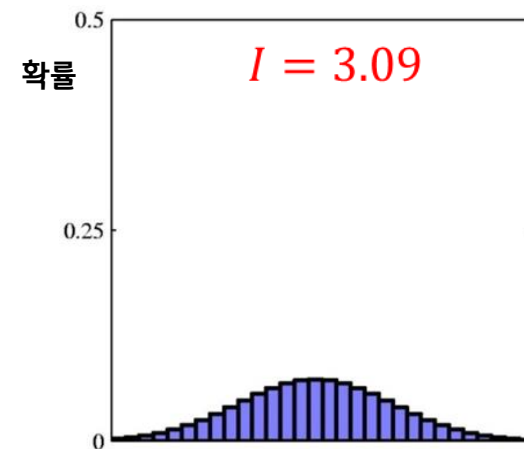
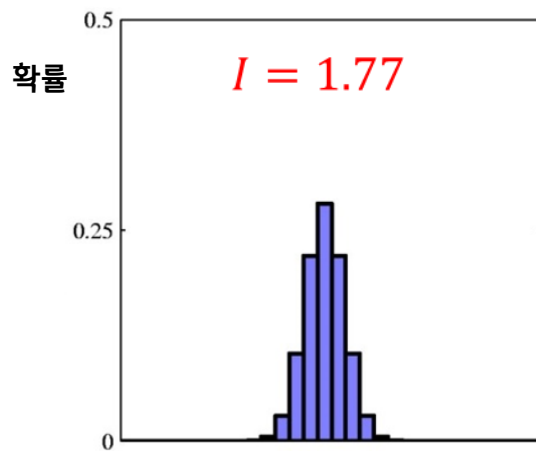
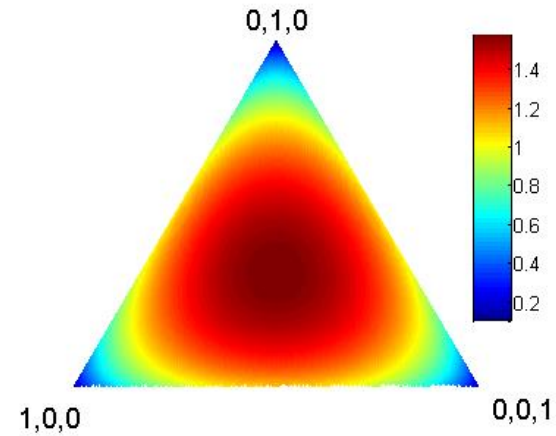
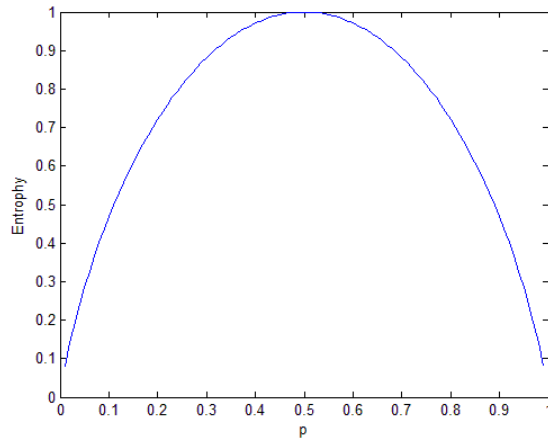
- $p(c)$  : 부류 c에 속하는 것의 비율
- 2개 부류가 있는 경우 엔트로피 분포



# 결정트리 학습 알고리즘

## ❖ 엔트로피의 특성

- 섞인 정도가 클 수록 큰 값





# 결정트리 학습 알고리즘

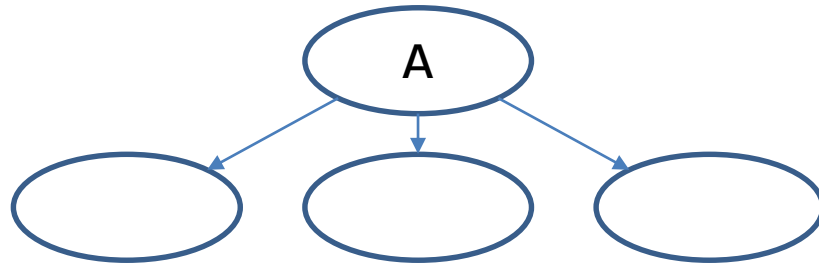
## ❖ 정보 이득(information gain)

- $IG = I - I_{res}$

- $I_{res}$  : 특정 속성으로 분할한 후의 각 부분집합의 정보량의 가중 평균

$$I_{res} = - \sum_v p(v) \sum_c p(c|v) \log_2 p(c|v)$$

$$IG = I - I_{res}(A) = - \sum_c p(c) \log_2 p(c) + \sum_v p(v) \sum_c p(c|v) \log_2 p(c|v)$$



- 정보 이득이 클 수록 우수한 분할 속성

# 결정트리 학습 알고리즘

## ❖ 학습 데이터의 예

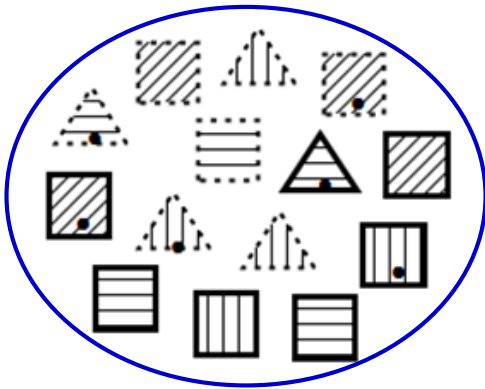
- 부류(class) 정보가 있는 데이터

	속성			부류
	<i>Pattern</i>	<i>Outline</i>	<i>Dot</i>	<i>Shape</i>
1	수직	점선	무	삼각형
2	수직	점선	유	삼각형
3	대각선	점선	무	사각형
4	수평	점선	무	사각형
5	수평	실선	무	사각형
6	수평	실선	유	삼각형
7	수직	실선	무	사각형
8	수직	점선	무	삼각형
9	대각선	실선	유	사각형
10	수평	실선	무	사각형
11	수직	실선	유	사각형
12	대각선	점선	유	사각형
13	대각선	실선	무	사각형
14	수평	점선	유	삼각형



# 결정트리 학습 알고리즘

## ❖ 엔트로피 계산



- 9 □ (사각형)

- 5 △ (삼각형)

- 부류별 확률(class probability)

$$p(\square) = \frac{9}{14}$$

$$p(\triangle) = \frac{5}{14}$$

- 엔트로피(entropy)

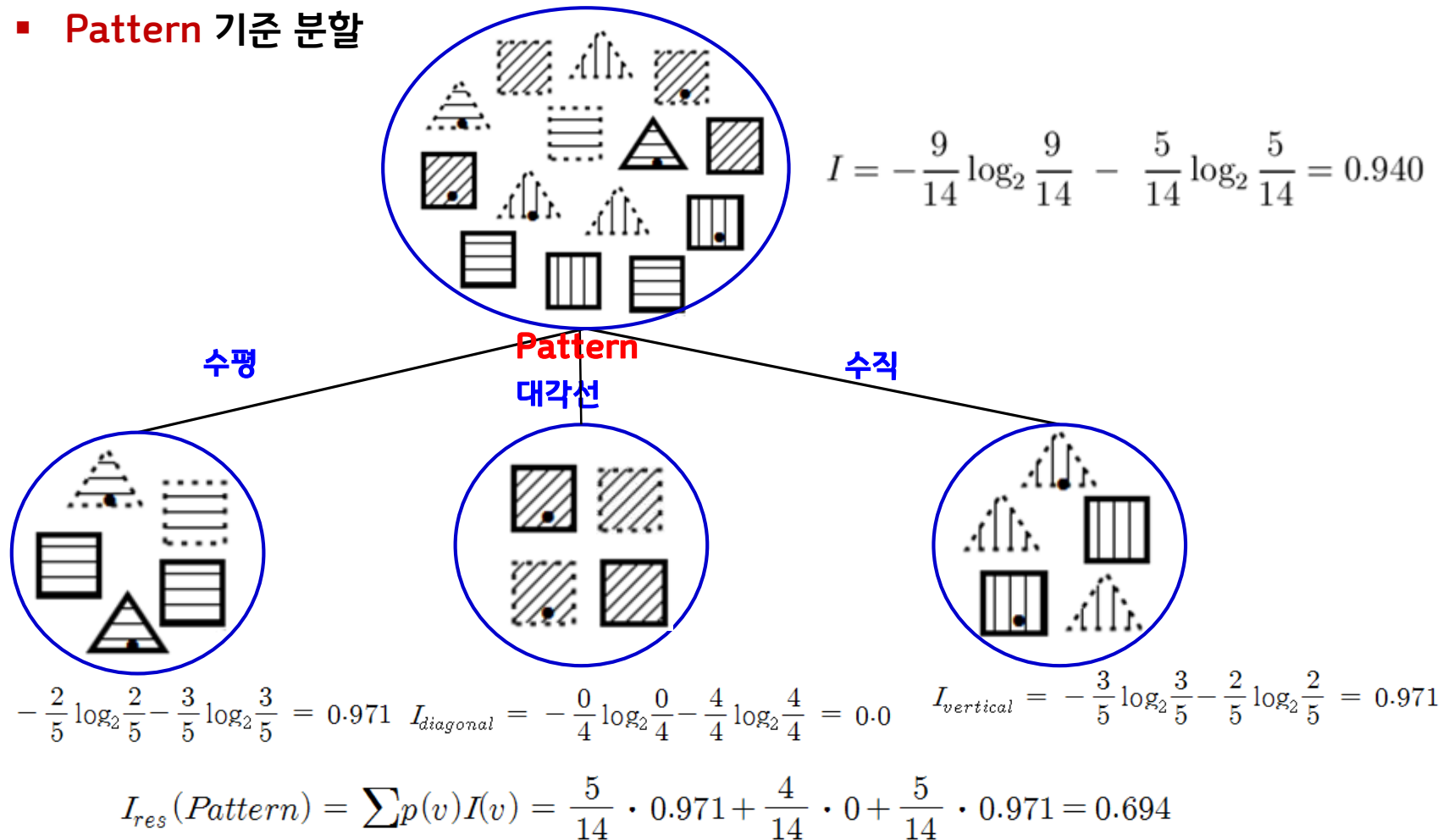
$$I = - \sum_c p(c) \log_2 p(c)$$

$$I = -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} = 0.940 \text{ bits}$$

# 결정트리 학습 알고리즘

## ❖ 데이터 집합 분할과 정보이득

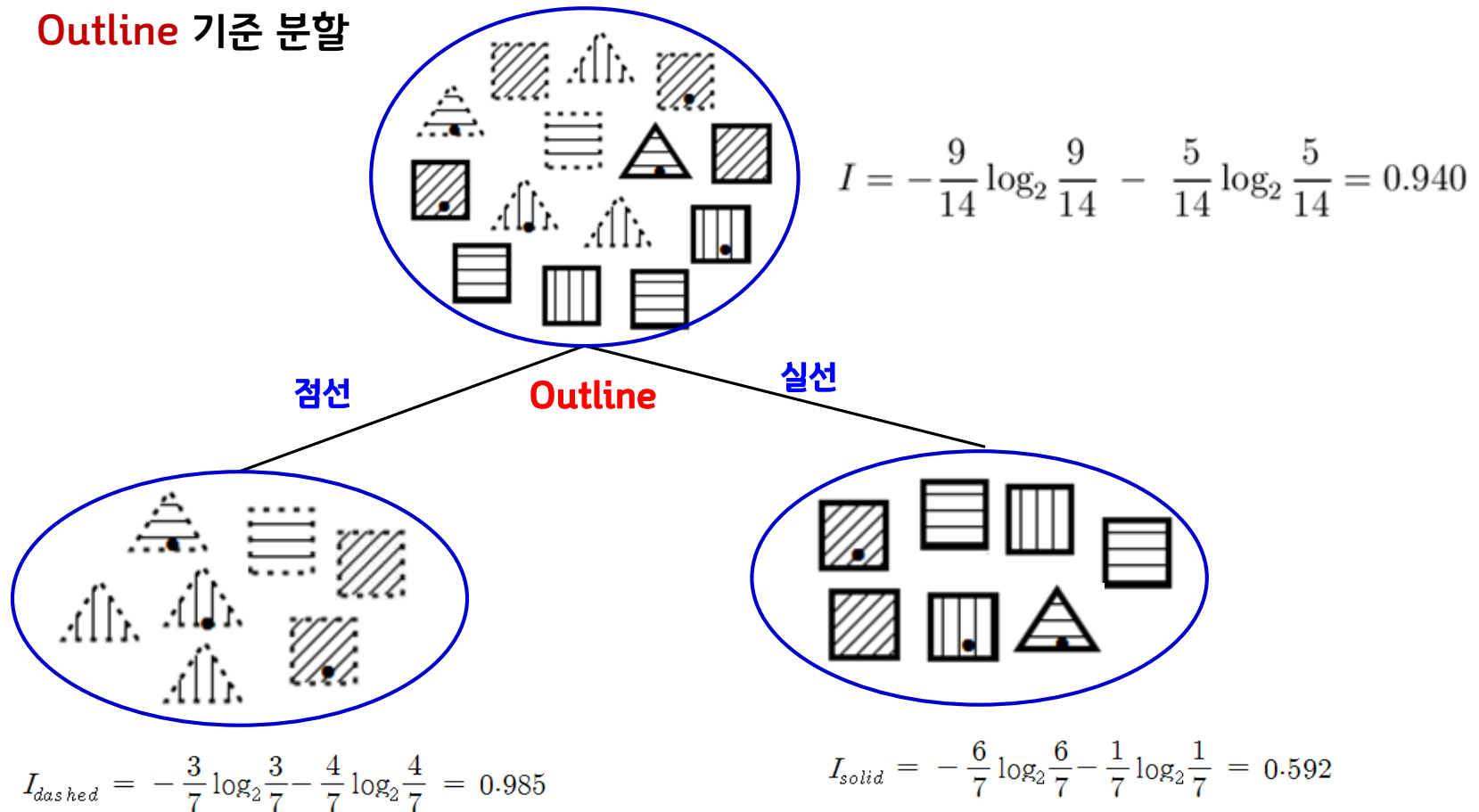
### ■ Pattern 기준 분할



# 결정트리 학습 알고리즘

## ❖ 데이터 집합 분할과 정보이득

### ▪ Outline 기준 분할



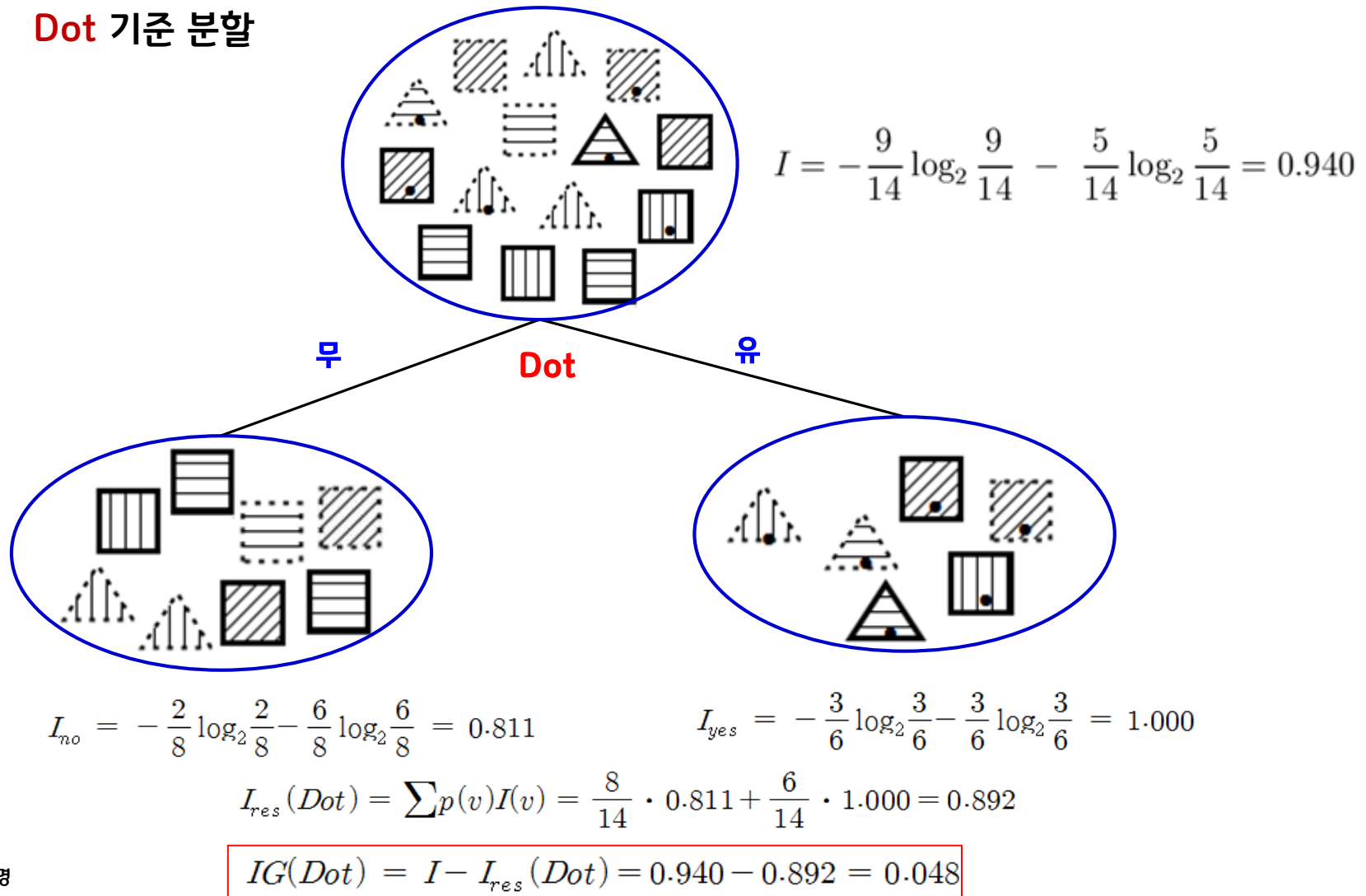
$$I_{res}(Outline) = \sum p(v) I(v) = \frac{7}{14} \cdot 0.985 + \frac{7}{14} \cdot 0.592 = 0.789$$

$$IG(Outline) = I - I_{res}(Outline) = 0.940 - 0.789 = 0.151$$

# 결정트리 학습 알고리즘

## ❖ 데이터 집합 분할과 정보이득

### ▪ Dot 기준 분할



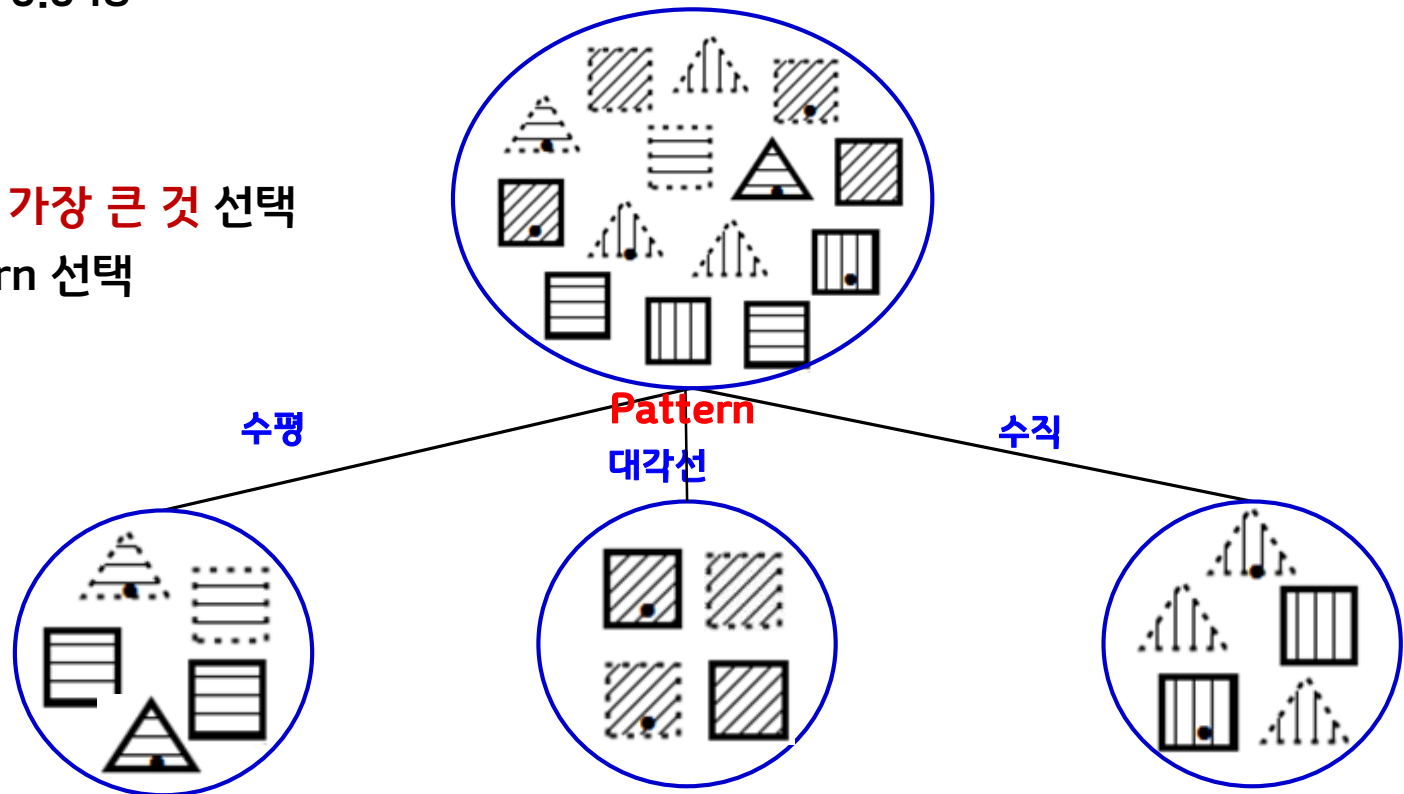
# 결정트리 학습 알고리즘

## ❖ 속성별 정보이득

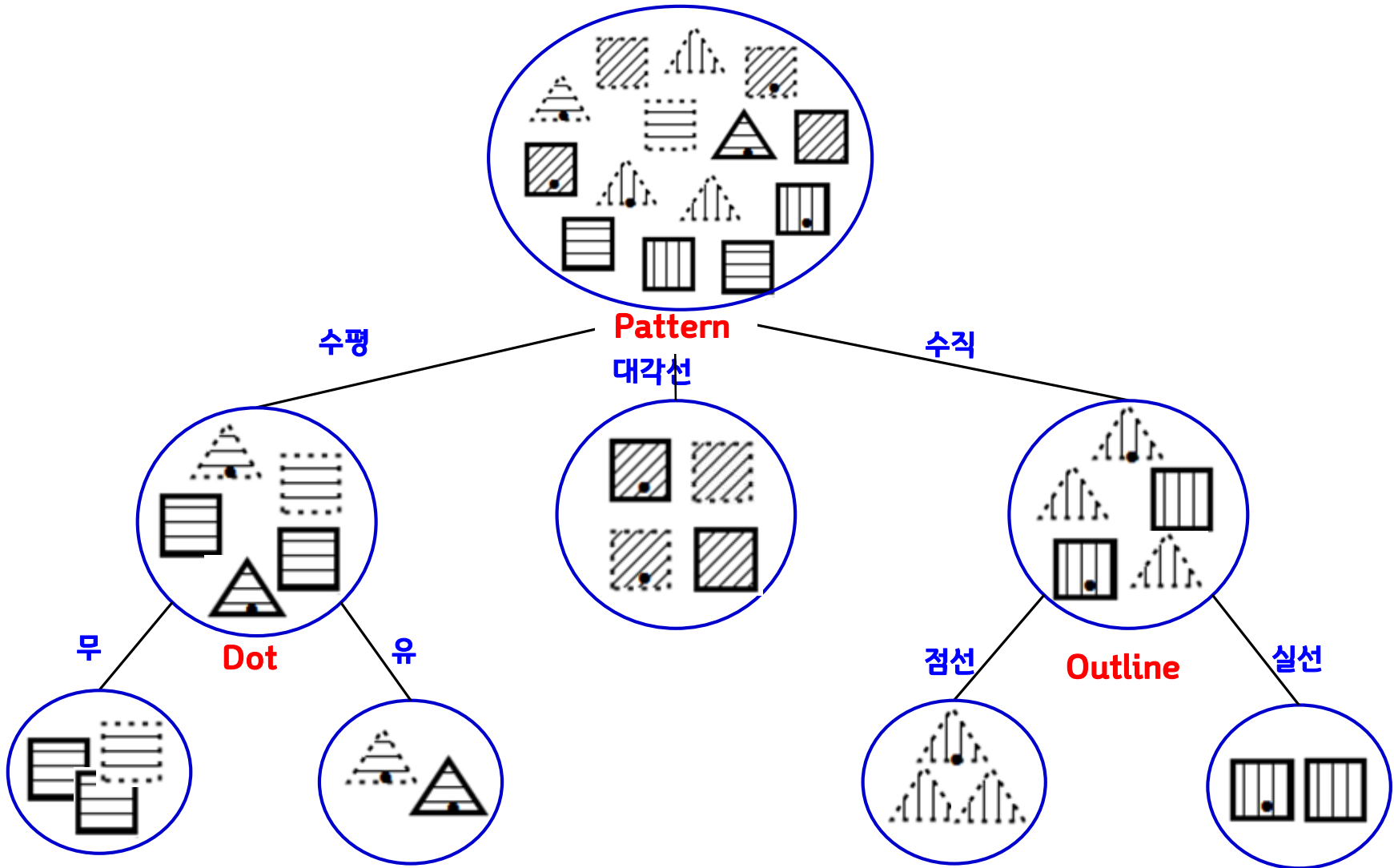
- $IG(\text{Pattern}) = 0.246$
- $IG(\text{Outline}) = 0.151$
- $IG(\text{Dot}) = 0.048$

## ❖ 분할속성 선택

- 정보이득이 가장 큰 것 선택
  - Pattern 선택



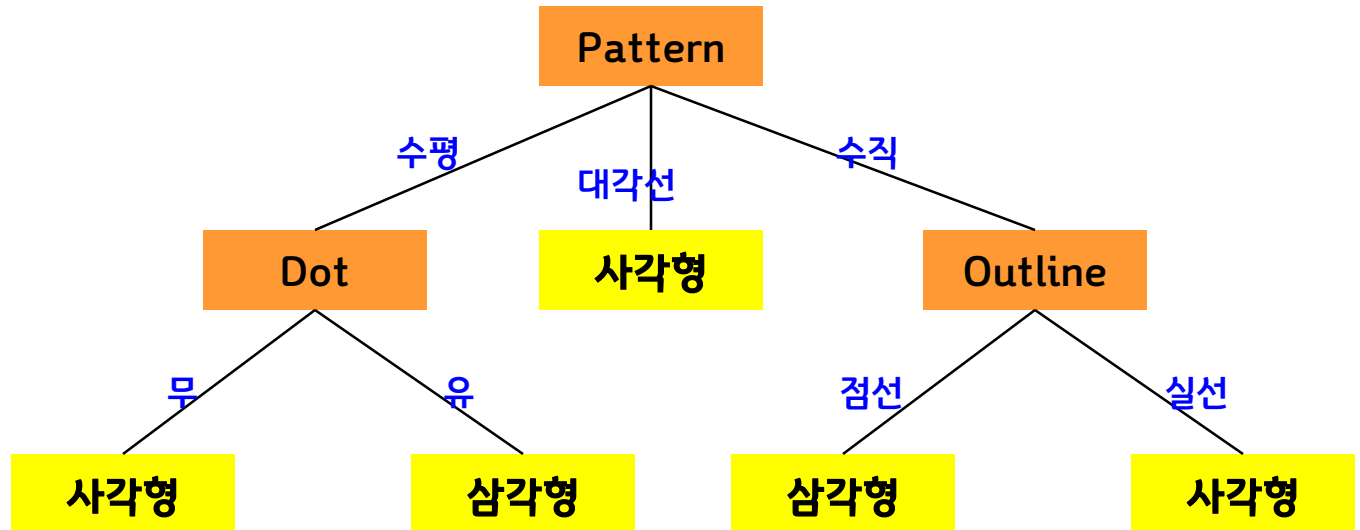
# 결정트리 학습 알고리즘





# 결정트리 학습 알고리즘

## ❖ 최종 결정트리



# 결정트리 학습 알고리즘

## ❖ 정보이득(information gain) 척도의 단점

$$IG = I - I_{res}(A) = - \sum_c p(c) \log_2 p(c) + \sum_v p(v) \sum_c p(c|v) \log_2 p(c|v)$$

- 속성값이 많은 것 선호
  - 예. 학번, 이름 등
- 속성값이 많으면 데이터집합을 많은 부분집합으로 분할
  - 작은 부분집합은 동질적인 경향

## ❖ 개선 척도

- 정보이득비(information gain ratio)
- 지니 지수(Gini index)

# 결정트리 학습 알고리즘

## ❖ 정보이득 비(information gain ratio) 척도

- 정보이득(information gain) 척도의 개선
  - 속성값의 가짓수가 많은 속성에 대해 불이익 부여

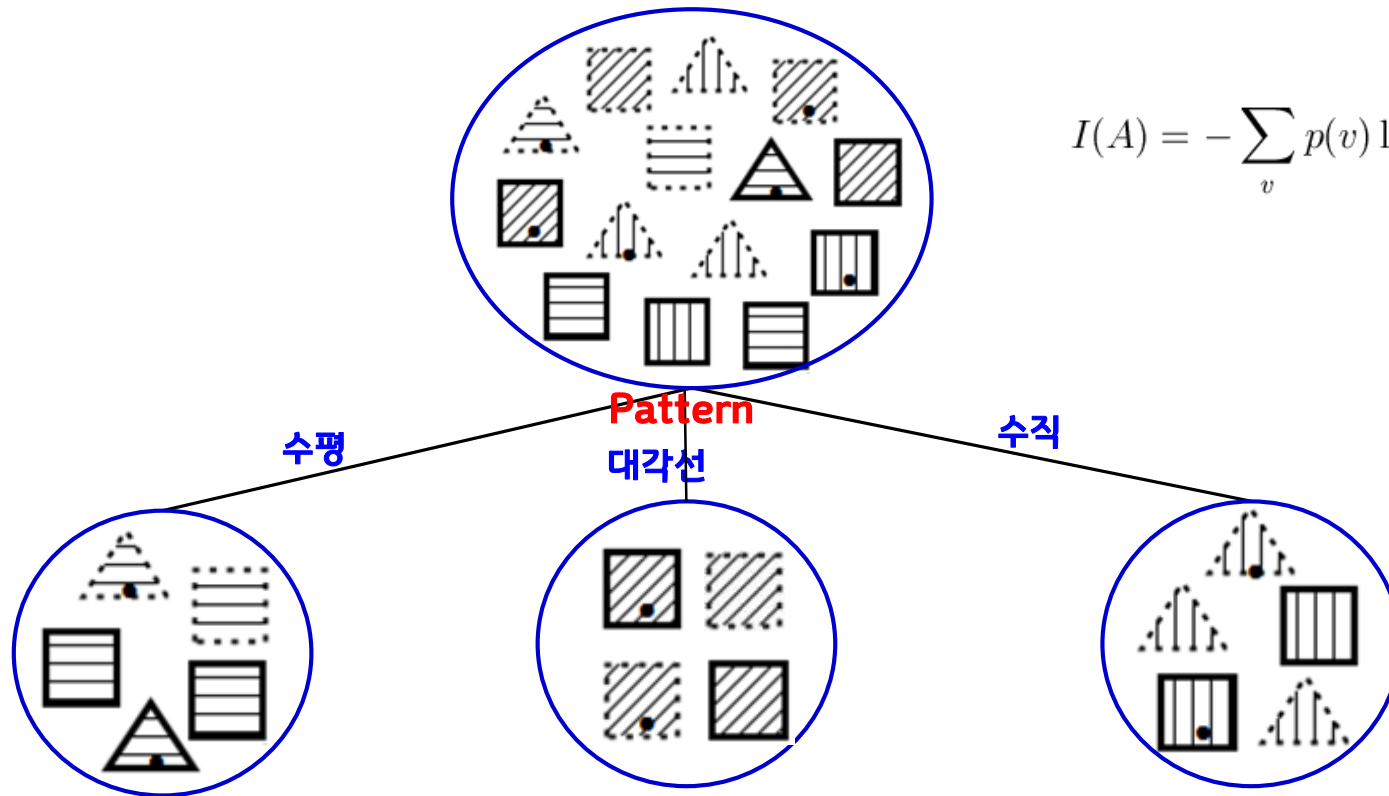
$$GainRatio(A) = \frac{IG(A)}{I(A)} = \frac{I - I_{res}(A)}{I(A)}$$

- $I(A)$ 
  - 속성  $A$ 의 속성값을 부류(class)로 간주하여 계산한 엔트로피
  - 속성값의 가짓수가 많을 수록 커지는 경향

$$I(A) = - \sum_v p(v) \log_2(p(v))$$

# 결정트리 학습 알고리즘

## ❖ 정보이득 비



$$I(A) = - \sum_v p(v) \log_2(p(v))$$

$$I(\text{Pattern}) = -\frac{5}{14} \log_2 \frac{5}{14} - \frac{5}{14} \log_2 \frac{5}{14} - \frac{4}{14} \log_2 \frac{4}{14} = 1.58$$

$$IG(\text{Pattern}) = I - I_{\text{res}}(\text{Pattern}) = 0.940 - 0.694 = 0.246$$

$$\text{GainRatio}(\text{Pattern}) = \frac{IG(\text{Pattern})}{I(\text{Pattern})} = \frac{0.246}{1.58} = 0.156$$

# 결정트리 학습 알고리즘

## ❖ 정보이득 vs 정보이득 비

속성	속성의 개수	정보 이득	정보 이득비
Pattern	3	0.247	0.156
Outline	2	0.152	0.152
Dot	2	0.048	0.049

Maiden Lane →



SPEED  
LIMIT  
30

Information  
Entropy

CANNAL



# 결정트리 학습 알고리즘

## ❖ 지니 지수(Gini index)

- 데이터 집합에 대한 지니(Gini) 값
  - $i, j$ 가 부류(class)를 나타낼 때

$$Gini = \sum_{i \neq j} p(i)p(j)$$



$$p(\square) = \frac{9}{14}$$

$$p(\triangle) = \frac{5}{14}$$

$$Gini = \frac{9}{14} \times \frac{5}{14} = 0.230$$

- 속성  $A$ 에 대한 지니 지수값의 가중평균

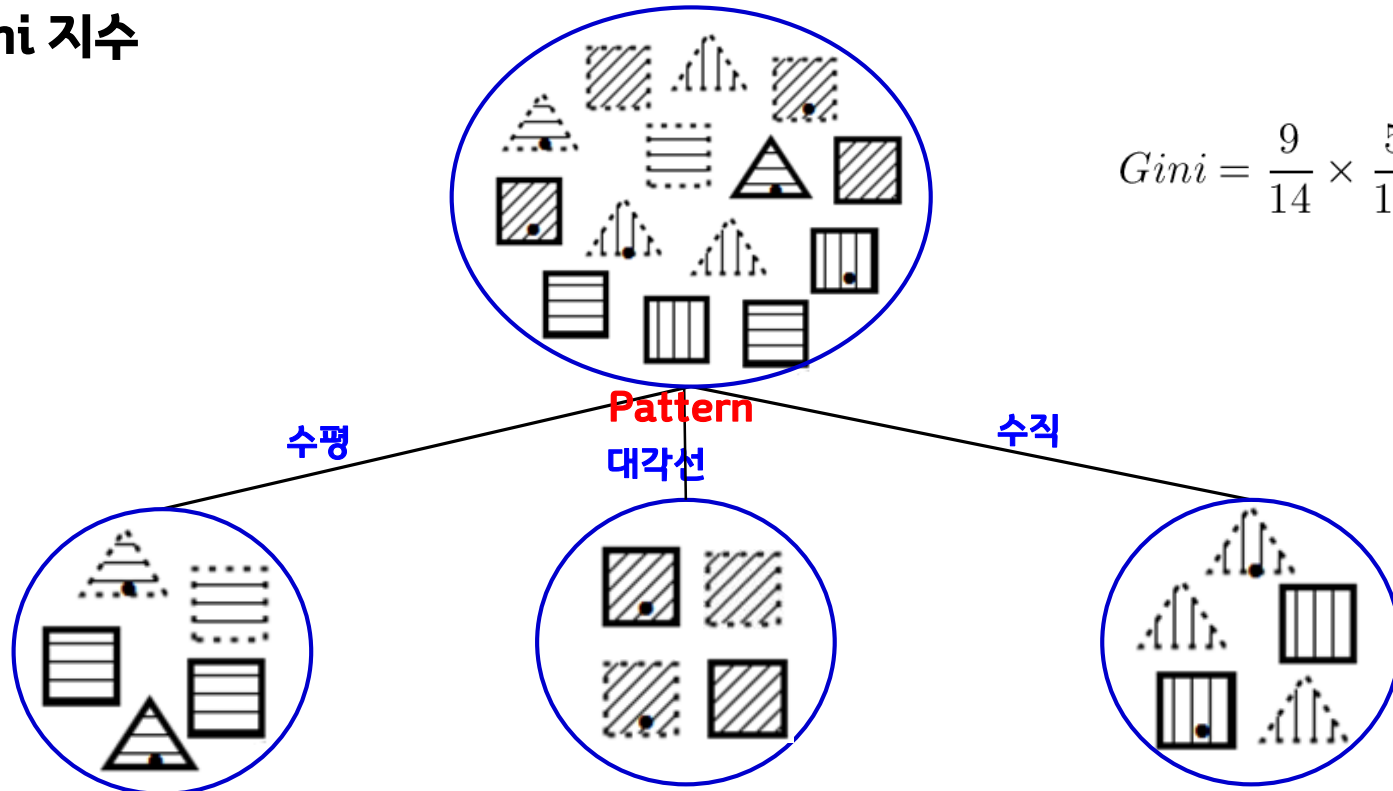
$$Gini(A) = \sum_v p(v) \sum_{i \neq j} p(i|v)p(j|v)$$

- 지니 지수 이득(Gini index gain)

$$GiniGain(A) = Gini - Gini(A)$$

# 결정트리 학습 알고리즘

## ❖ Gini 지수



$$Gini = \frac{9}{14} \times \frac{5}{14} = 0.230$$

$$Gini(A) = \sum_v p(v) \sum_{i \neq j} p(i|v) p(j|v)$$

$$Gini(Pattern) = \frac{5}{14} \times \left( \frac{3}{5} \times \frac{2}{5} \right) + \frac{5}{14} \times \left( \frac{2}{5} \times \frac{3}{5} \right) + \frac{4}{14} \times \left( \frac{4}{4} \times \frac{0}{4} \right) = 0.171$$

$$Gini\ Gain(Pattern) = 0.230 - 0.171 = 0.058$$



# 결정트리 학습 알고리즘

## ❖ 분할속성 평가 척도 비교

속성	정보 이득	정보 이득비	지니 이득
Pattern	0.247	0.156	0.058
Outline	0.152	0.152	0.046
Dot	0.048	0.049	0.015

# 결정트리 학습 알고리즘

## ❖ 결정트리 알고리즘

### ▪ ID3 알고리즘

- 범주형(categorical) 속성값을 갖는 데이터에 대한 결정트리 학습
- 예. PlayTennis, 삼각형/사각형 문제

### ▪ C4.5 알고리즘

- 범주형 속성값과 수치형 속성값을 갖는 데이터로 부터 결정트리 학습
- ID3를 개선한 알고리즘

### ▪ C5.0 알고리즘

- C4.5를 개선한 알고리즘

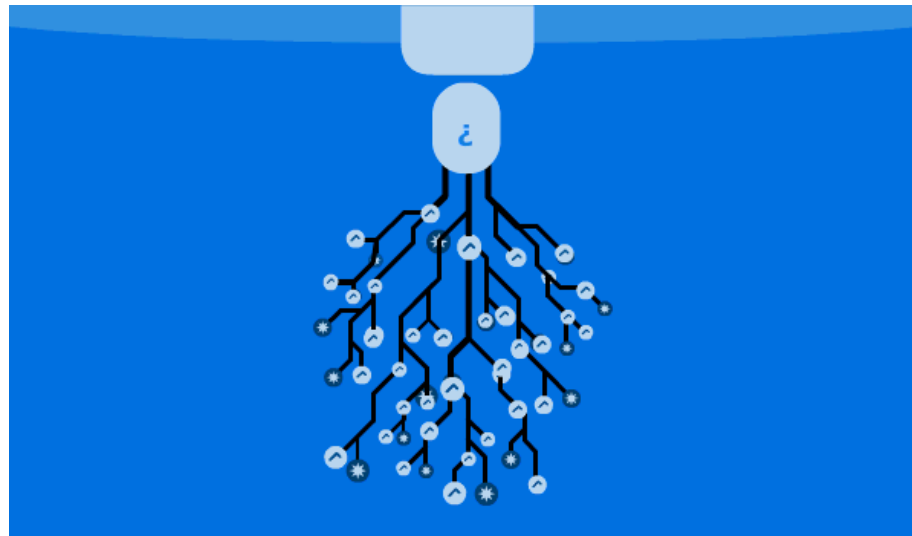
### ▪ CART 알고리즘

- 수치형 속성을 갖는 데이터에 대해 적용

# 결정트리 학습 알고리즘

## ❖ 결정트리의 과적합

- 트리의 깊이가 지나치게 길어진 경우
- 과적합 완화 방법
  - 결정트리 가지치기
    - 트리 생성 후 가지치기
  - 조기 중단(early stopping)
    - 트리 생성 중간에 트리 생성 중단
    - 결정트리 깊이 제한
    - 단말노드의 최소 데이터 개수 지정



## ❖ [실습] 결정트리 생성

```
from sklearn import datasets
import numpy as np
```

```
def test_split(index, value, dataset): # 데이터 분할
    left, right = list( ), list( )
    for row in dataset:
        if row[index] < value:
            left.append(row)
        else:
            right.append(row)
    return left, right
```

```
def gini_index(groups, classes): # 지니지수 계산
    n_instances = float(sum([len(group) for group in groups]))
    gini = 0.0
    for group in groups:
        size = float(len(group))
        if size == 0:
            continue
        score = 0.0
        for class_val in classes:
            p = [row[-1] for row in group].count(class_val) / size
            score += p * p
        gini += (1.0 - score) * (size / n_instances)
    return gini
```

```

def get_split(dataset):
    class_values = list(set(row[-1] for row in dataset))
    b_index, b_value, b_score, b_groups = 999, 999, 999, None
    for index in range(len(dataset[0])-1):
        for row in dataset:
            groups = test_split(index, row[index], dataset)
            gini = gini_index(groups, class_values)
            if gini < b_score:
                b_index, b_value, b_score, b_groups = index, row[index], gini, groups
    return {'index':b_index, 'value':b_value, 'groups':b_groups}

```

```

def split(node, max_depth, min_size, depth): # 분할
    left, right = node['groups']
    del(node['groups'])
    if not left or not right:
        node['left'] = node['right'] = to_terminal(left + right)
        return
    if depth >= max_depth:
        node['left'], node['right'] = to_terminal(left), to_terminal(right)
        return
    if len(left) <= min_size:
        node['left'] = to_terminal(left)
    else:
        node['left'] = get_split(left)
        split(node['left'], max_depth, min_size, depth+1)
    if len(right) <= min_size:
        node['right'] = to_terminal(right)
    else:
        node['right'] = get_split(right)
        split(node['right'], max_depth, min_size, depth+1)

```

```
def to_terminal(group):
    outcomes = [row[-1] for row in group]
    return max(set(outcomes), key=outcomes.count)
```

```
def build_tree(train, max_depth, min_size): # 결정트리 생성
    root = get_split(train)
    split(root, max_depth, min_size, 1)
    return root
```

```
def print_tree(node, depth=0):
    if isinstance(node, dict):
        print('%s[X%d < %.3f]' % ((depth*' ', (node['index']+1), node['value'])))
        print_tree(node['left'], depth+1)
        print_tree(node['right'], depth+1)
    else:
        print('%s[%s]' % ((depth*' ', node)))
```

```
iris = datasets.load_iris()
dataset = np.c_[iris.data,iris.target]
```

```
tree = build_tree(dataset, 3, 1)
print_tree(tree)
```

```
[X3 < 3.000]
  [X1 < 5.100]
    [X1 < 4.900]
      [0.0]
      [0.0]
      [X1 < 5.100]
        [0.0]
        [0.0]
        [X4 < 1.800]
          [X3 < 5.000]
            [1.0]
            [2.0]
            [X3 < 4.900]
              [2.0]
              [2.0]
```

## 2. 결정트리를 이용한 회귀

### ❖ 회귀(regression)를 위한 결정트리

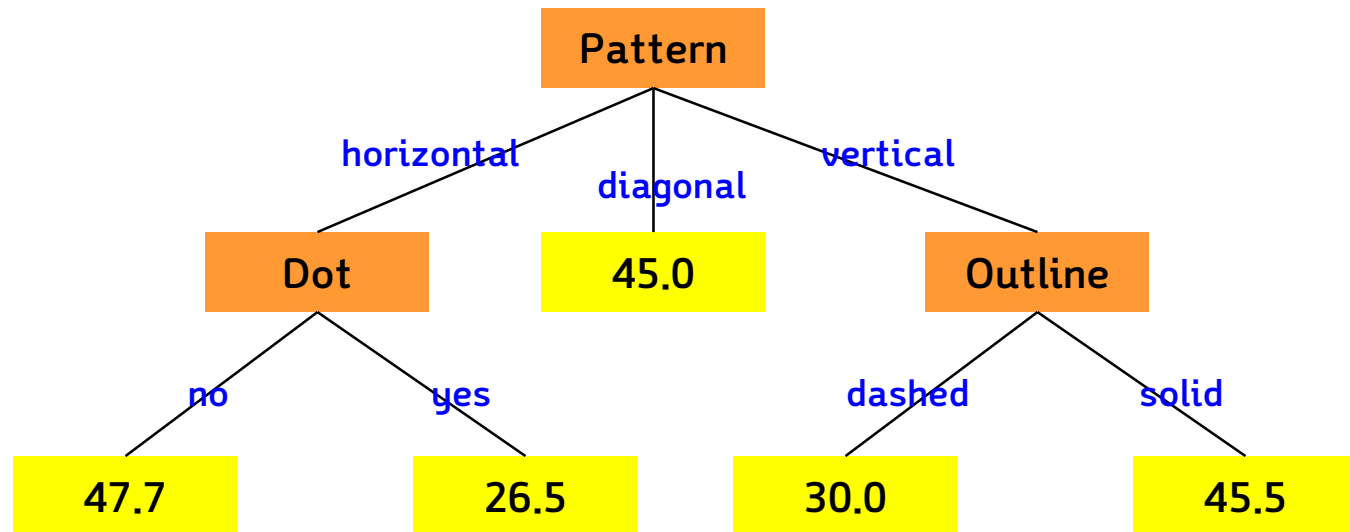
- 출력값이 수치값

표 4.5 도형 면적에 대한 데이터

	속성			면적
	<i>Pattern</i>	<i>Outline</i>	<i>Dot</i>	<i>Area</i>
1	수직	점선	무	25
2	수직	점선	유	30
3	대각선	점선	무	46
4	수평	점선	무	45
5	수평	실선	무	52
6	수평	실선	유	23
7	수직	실선	무	43
8	수직	점선	무	35
9	대각선	실선	유	38
10	수평	실선	무	46
11	수직	실선	유	48
12	대각선	점선	유	52
13	대각선	실선	무	44
14	수평	점선	유	30

# 결정트리를 이용한 회귀

## ❖ 회귀(regression)를 위한 결정트리





# 결정트리를 이용한 회귀

## ❖ 회귀 (regression)를 위한 결정트리

- 분류를 위한 결정트리와 차이점
  - 단말노드가 부류(class)가 아닌 수치값(numerical value)임
  - 해당 조건을 만족하는 것들이 가지는 대표값
- 분할 속성 선택
  - 표준편차 축소(reduction of standard deviation)  $SDR$ 를 최대화 하는 속성 선택

$$SDR(A) = SD - SD(A)$$

– 표준편차  $SD = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - m)^2}$   $m$  : 평균

–  $SD(A)$

» 속성 A를 기준으로 분할 후의 부분 집합별 표준편차의 가중평균

# 결정트리를 이용한 회귀

## ❖ 회귀(regression)를 위한 결정트리

Area
26
30
48
46
62
23
43
36
38
48
48
62
44
30

←  $SD = 9.67$

		Area의 표준편차	개수
<i>Pattern</i>	수평	12.15	5
	수직	9.36	5
	대각선	5.77	4
			14

$$SD(Pattern) = \frac{5}{14} \times 12.15 + \frac{5}{14} \times 9.36 + \frac{4}{14} \times 5.77 = 9.05$$

$$SDR(Pattern) = SD - SD(Pattern) = 9.67 - 9.05 = 0.61$$

## ❖ [실습] 결정트리 회귀

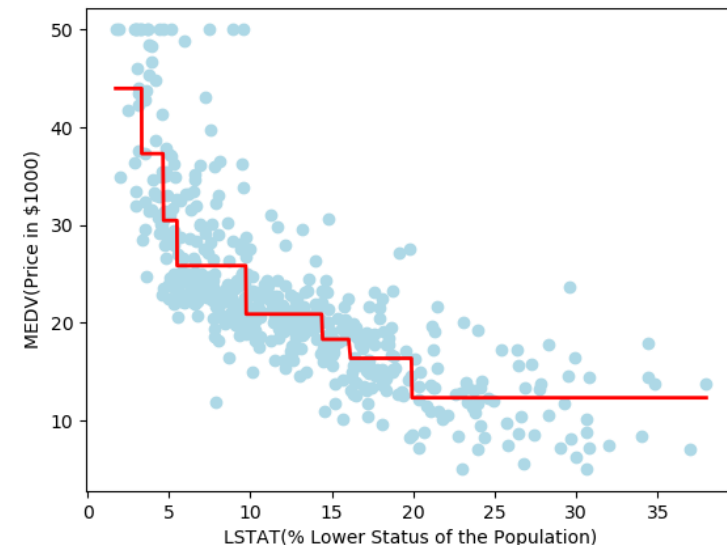
```
from sklearn.tree import DecisionTreeRegressor
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('housing.data', header=None, sep='\\s+')
df.columns = ['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD',
              'TAX', 'PTRATIO', 'B', 'LSTAT', 'MEDV']
print(df.head())
X = df[['LSTAT']].values
y = df['MEDV'].values

tree = DecisionTreeRegressor(max_depth=3)
tree.fit(X,y)

sort_idx = X.flatten().argsort()
plt.scatter(X[sort_idx], y[sort_idx], c='lightblue')
plt.plot(X[sort_idx], tree.predict(X[sort_idx]), color='red', linewidth=2)
plt.xlabel('LSTAT(% Lower Status of the Population)')
plt.ylabel('MEDV(Price in $1000)')
plt.show( )
```

	CRIM	ZN	INDUS	CHAS	NOX	...	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0	0.538	...	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	...	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	...	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	...	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	...	222.0	18.7	396.90	5.33	36.2



# Quiz

## ❖ 결정 트리에 대한 설명으로 옳지 않은 것은?

- ① 결정 트리의 내부 노드에는 비교할 속성이 위치한다.
- ② 단말 노드에는 출력값이 위치한다.
- ③ 동일한 성능이면 트리의 깊이가 낮은 것이 바람직하다.
- ④ 분할 속성은 엔트로피가 큰 것 중에서 선택한다.

## ❖ 다음 척도에 대한 설명을 옳지 않은 것은?

- ① 엔트로피가 클수록 해당 집단의 동질성이 크다.
- ② 정보 이득이 큰 속성이 일반적으로 분할 속성으로 바람직하다.
- ③ 정보이득비는 속성값의 개수가 많은 속성에 대해서 불이익을 준다.
- ④ 지니 지수 이득이 큰 속성은 일반적으로 분할 속성으로 바람직하다.

## ❖ 결정 트리의 노드에 사용되는 기준으로 옳지 않은 것은?

- ① 엔트로피    ② 지니 불순도    ③ 정보 이득    ④ 경사하강법

# Quiz

- ❖ 결정트리에서 과적합(overfitting)을 방지하기 위한 전략이 아닌 것은?
  - ① 가지치기(pruning)
  - ② 최대 깊이 제한
  - ③ 노드의 최소 데이터 수 설정
  - ④ 더 많은 데이터 수집
  
- ❖ 결정 트리의 깊이(depth)가 너무 깊을 때 발생할 수 있는 가장 큰 문제점은?
  - ① 편향이 증가한다.
  - ② 분산이 감소한다.
  - ③ 과적합(overfitting)이 발생할 확률이 높아진다.
  - ④ 계산 속도가 빨라진다.
  
- ❖ 결정 트리에서 노드가 순수하다는 것의 의미는?
  - ① 노드에 속한 샘플들이 모두 같은 클래스에 속한다.
  - ② 노드에 속한 샘플들의 수가 일정 이상이다.
  - ③ 노드에 속한 샘플들의 평균 값이 특정 값 이상이다.
  - ④ 노드에 속한 샘플들이 모두 다른 클래스에 속한다.

# Quiz

## ❖ 결정트리에 대한 설명으로 옳지 않은 것은?

- ① 결정트리를 생성할 때 분할속성 선택에 따라 결정트리의 크기가 영향을 받는다.
- ② 결정트리를 회귀 문제에 적용할 수 있다.
- ③ C4.5 알고리즘은 ID3 알고리즘보다 개선된 알고리즘이다.
- ④ 결정트리를 적용하기 위해서는 모두 입력 속성이 범주형 값을 가져야 한다.

## ❖ 결정트리에 대한 설명으로 옳지 않은 것은?

- ① 결정트리를 생성할 때 분할속성 선택에 따라 결정트리의 크기가 영향을 받는다.
- ② 엔트로피가 큰 데이터 집합일수록 동일한 부류에 속하는 경향이 크다.
- ③ 정보이득이 큰 속성이 일반적으로 분할속성으로서 바람직하다.
- ④ 결정트리는 분류 문제뿐만 아니라 회귀 문제에도 사용된다.

## ❖ 결정 트리의 특성이 아닌 것은?

- ① 비선형 구조의 데이터도 분류 가능하다.
- ② 결정 경계가 축에 수직이다.
- ③ 대부분의 경우 데이터의 정규화나 표준화가 필요하다.
- ④ 복잡한 트리는 데이터의 노이즈까지 학습할 수 있다.

# Quiz

❖ 결정트리 회귀에서 노드를 분할할 때 주로 사용하는 불순도 측정 방법은?

- ① 엔트로피
- ② 평균 표준편차 차이
- ③ 지니 불순도
- ④ 정보이득

❖ 결정트리 회귀에서 단말 노드의 예측 값은 어떻게 결정되는가?

- ① 해당 노드의 모든 데이터 포인트의 평균 값
- ② 해당 노드의 중간 값
- ③ 해당 노드의 최빈 값
- ④ 해당 노드의 데이터 포인트 중 임의의 값

❖ 결정트리 회귀에서 노드 분할의 주요 목적은 무엇인가?

- ① 클래스 라벨 분포 최적화
- ② 최대화
- ③ 오차의 최소화
- ④ 노드 크기 최대화

# Quiz

❖ 결정트리 회귀의 단말 노드에서 예측하는 값의 특징은?

- ① 범주형 값
- ② 연속적인 값
- ③ 이진 값
- ④ 최빈 값

❖ 지니지수가 0인 경우, 해당 노드는 어떤 특성을 가지고 있는가?

- ① 노드에 데이터가 없다.
- ② 노드의 데이터가 모두 같은 클래스에 속한다.
- ③ 노드의 데이터가 여러 클래스에 균등하게 분포되어 있다.
- ④ 노드의 데이터가 두 개의 클래스에 반반씩 분포되어 있다.

❖ 정보이득을 최대화하는 방향으로 결정 트리를 분할하면 어떤 효과가 발생하는가?

- ① 노드의 순도가 감소한다.
- ② 노드의 지니지수가 증가한다.
- ③ 분할 후의 불확실성이 감소한다.
- ④ 데이터의 분포가 균일해진다.



# Quiz

❖ 불순도(Impurity)가 낮은 경우 정보 이득은 어떻게 됩니까?

- ① 높다
- ② 낮다
- ③ 항상 0이다
- ④ 변하지 않는다.

❖ 어떤 속성이 다른 속성보다 정보 이득이 높다면, 결정 트리를 생성할 때 어떻게 하는가?

- ① 해당 속성을 무시한다.
- ② 해당 속성을 먼저 사용하여 분할한다.
- ③ 해당 속성을 나중에 사용하여 분할한다.
- ④ 해당 속성을 제거한다.

❖ 엔트로피가 0인 노드의 특징은 무엇입니까?

- ① 모든 클래스의 샘플이 고르게 분포한다.
- ② 한 클래스의 샘플만 존재한다.
- ③ 두 개 이상의 클래스의 샘플이 동일한 비율로 있다.
- ④ 엔트로피를 계산할 수 없다.

# Quiz

❖ 엔트로피 공식에서  $\log_2 p_i$ 의 값이 음수일 경우, 그 이유는?

- ① 잘못된 계산 때문이다.
- ②  $p_i$ 값이 0과 1 사이에 있기 때문이다.
- ③  $p_i$ 값이 1 이상이기 때문이다.
- ④  $p_i$ 값이 0 이하이기 때문이다.

❖ 두 개의 클래스가 동일한 수의 샘플을 가질 때, 엔트로피의 값은?

- ① 0
- ② 1
- ③ 0.5
- ④ 2

❖ 지니 지수의 공식은?

- ①  $Gini = -\sum p_i(1 - p_i)$
- ②  $Gini = -\sum p_i^2$
- ③  $Gini = 1 - \sum p_i^2$
- ④  $Gini = \sum_{i \neq j} p_i p_j$

# Quiz

## ❖ 지니 지수가 0인 경우, 그 노드의 상태는?

- ① 모든 클래스의 샘플이 고르게 분포한다.
- ② 한 클래스의 샘플만 존재한다.
- ③ 두 개 이상의 클래스의 샘플이 동일한 비율로 있다.
- ④ 모든 클래스의 샘플이 없다.

## ❖ 지니 지수의 값의 범위는?

- ① 0과 1 사이
- ② 0과 무한대 사이
- ③ -1과 1 사이
- ④ 1과 2 사이

## ❖ 결정트리 회귀에서 트리의 가지치기(pruning)의 주된 목적은?

- ① 트리의 크기를 늘리기 위해
- ② 트리의 깊이를 극대화하기 위해
- ③ 과적합을 방지하기 위해
- ④ 모든 노드를 리프 노드로 만들기 위해

# Quiz

## ❖ 결정트리에서 지니 지수의 역할에 대한 잘못된 설명은?

- ① 회귀 모델 구축에서 분할속성의 적합도를 측정한다.
- ② 낮은 지니 지수는 높은 순도를 의미한다.
- ③ 지니 지수 이득을 통해 최적의 분할을 결정한다.
- ④ 지니 지수가 높을수록 데이터 분포가 고르다.

## ❖ 결정트리에 대한 설명으로 옳지 않은 것은?

- ① ID3 알고리즘은 범주형 속성값만을 갖는 데이터에 대한 결정트리를 생성한다.
- ② C4.5 알고리즘은 범주형 속성값과 수치형 속성값을 갖는 데이터에 대한 결정트리를 생성한다.
- ③ CART 알고리즘은 수치형 속성을 갖는 데이터에 대해 적용될 수 있다.
- ④ 결정트리 알고리즘은 분류 문제에 적용되고 회귀 문제에는 적용되지 않는다.