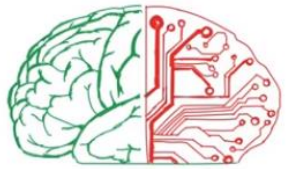


결정트리 학습
Decision Tree Induction

이건명

충북대학교 소프트웨어학과

인공지능 : 튜링 테스트에서 딥러닝까지

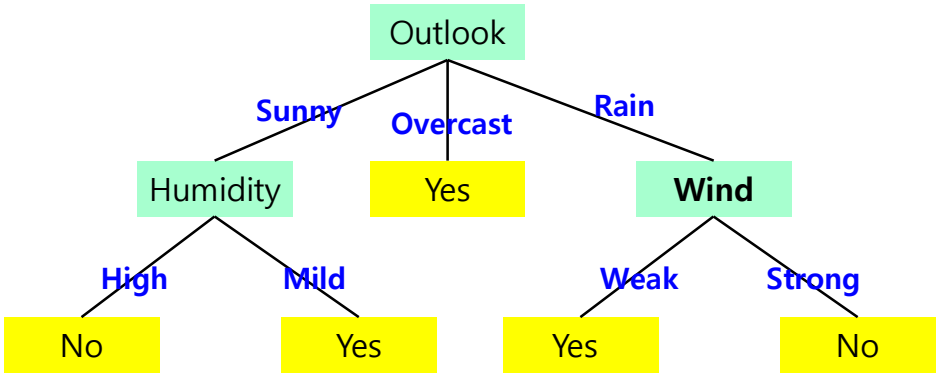
학습 내용

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

1. 결정트리

❖ **결정트리**(decision tree)

- **트리 형태**로 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

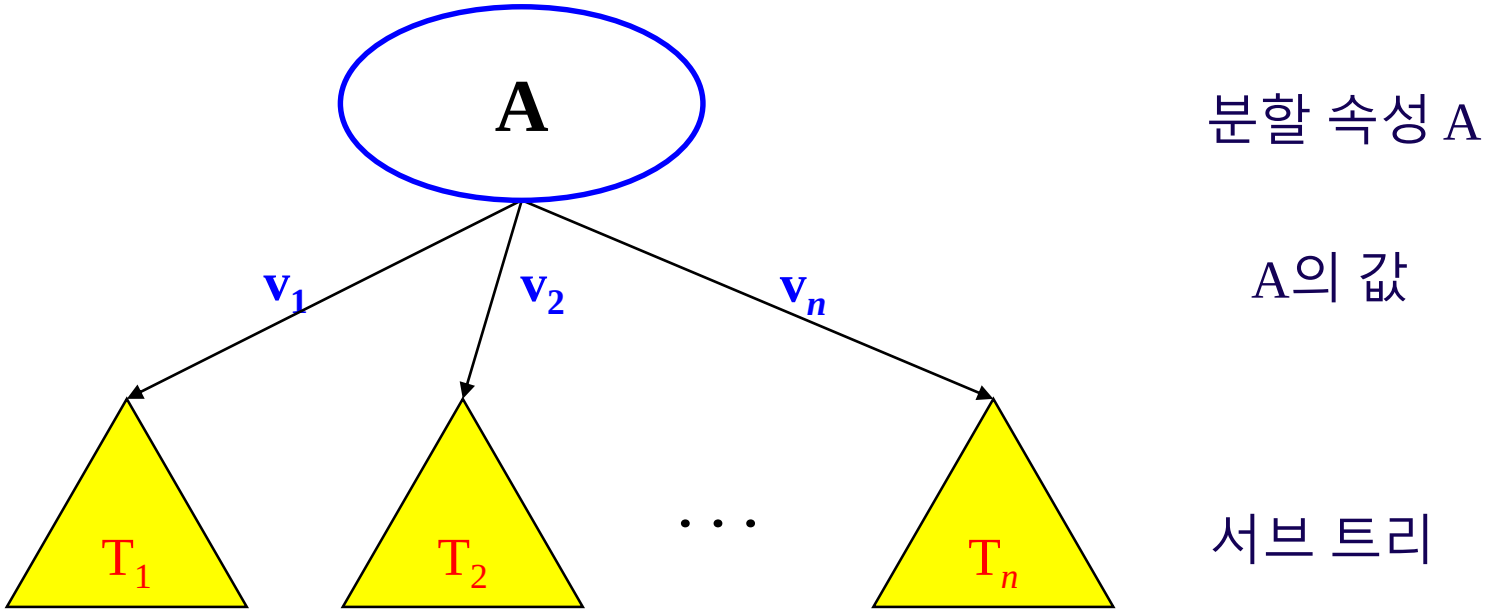
IF Outlook = Sunny **AND** Humidity = High **THEN** Answer = No

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

1.1 결정트리 학습 알고리즘

❖ **결정 트리 (decision tree) 알고리즘**

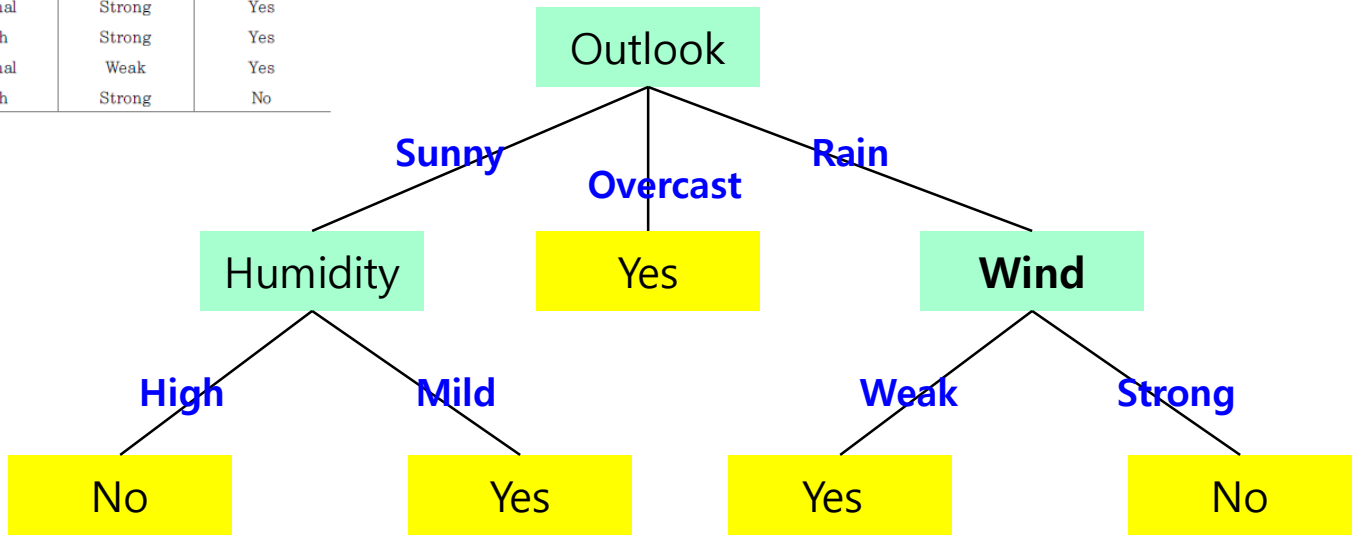
- 모든 데이터를 포함한 **하나의 노드**로 구성된 **트리**에서 시작
- **반복적인 노드 분할** 과정
  1. **분할 속성**(spliting attribute)을 **선택**
  2. 속성값에 따라 **서브트리**(subtree)를 **생성**
  3. **데이터**를 속성값에 따라 **분배**



결정트리 학습 알고리즘

❖ **결정 트리** (decision tree)
- 간단한 트리

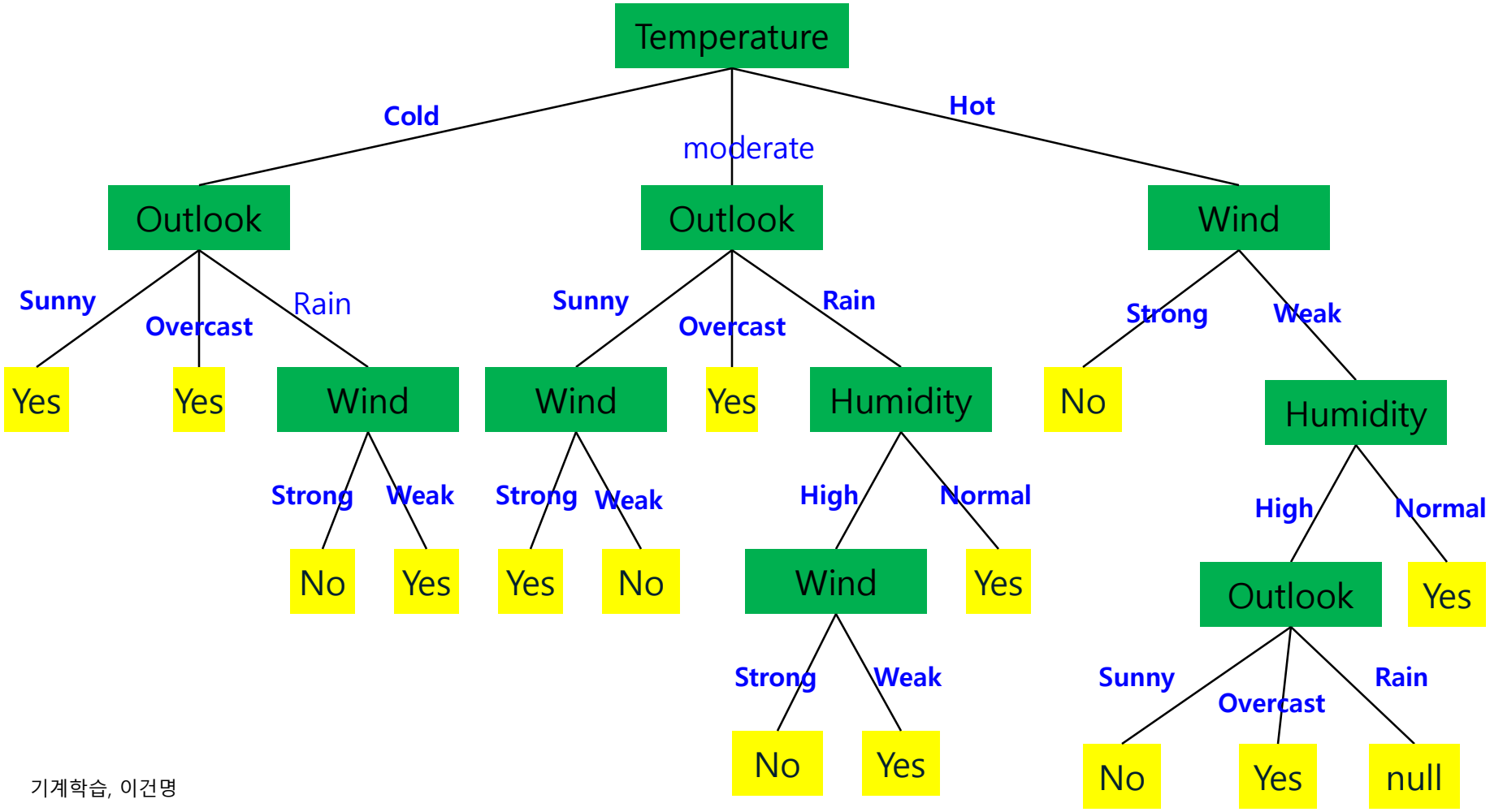
Day 날짜	Outlook 조망	Temperature 기온	Humidity 습도	Wind 바람	PlayTennis 테니스 여부
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny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

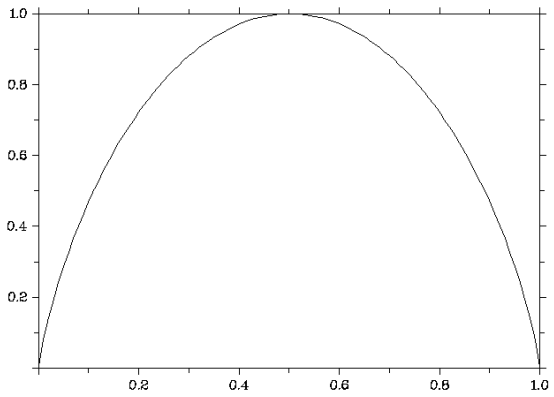
- 어떤 속성을 선택하는 것이 효율적인가
  - **분할한 결과**가 가능하면 동질적인(pure) 것으로 만드는 속성 선택

- **엔트로피**(Entropy)
  - 동질적인 정도 측정 가능 척도
  - 원래 정보량(amount of information) 측정 목적의 척도

$$I = -\sum_{c} p(c) \log_2 p(c)$$

  - $p(c)$: 부류 c에 속하는 것의 비율

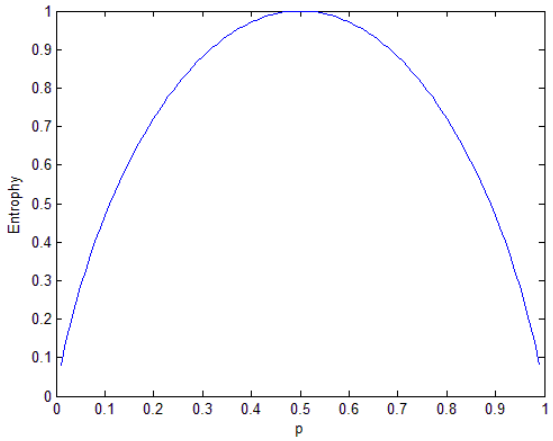
  - 2개 부류가 있는 경우 엔트로피

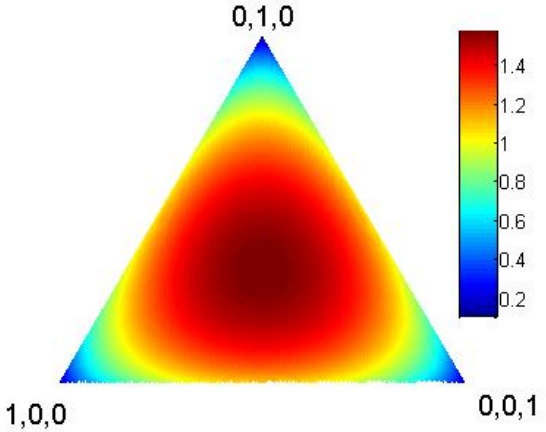


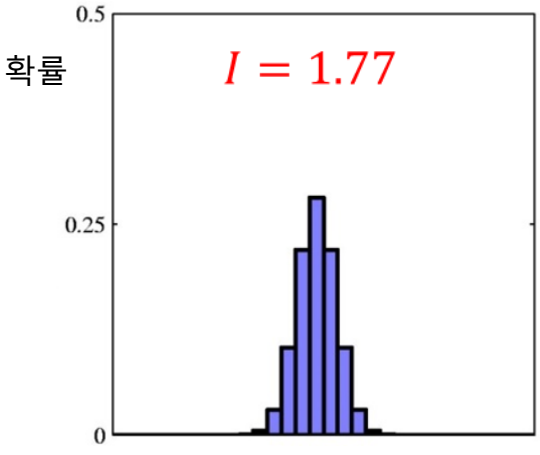
결정트리 학습 알고리즘

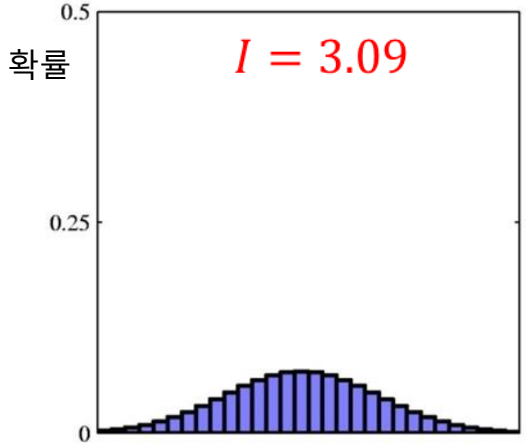
❖ 엔트로피의 특성

- 섞인 정도가 클 수록 큰 값









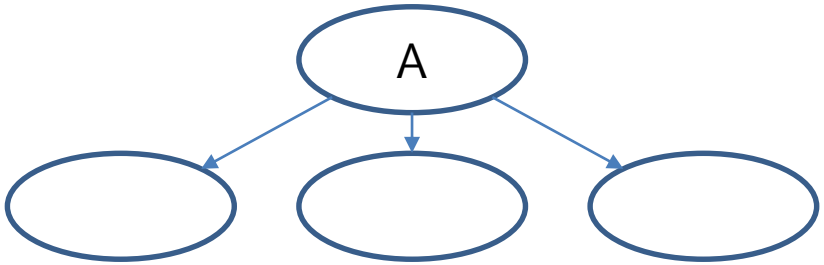
결정트리 학습 알고리즘

❖ **정보 이득** (information gain)

- $IG = I - I_{res}$
  - I_{res} : 특정 속성으로 **분할한 후**의 각 **부분집합**의 **정보량**의 **가중평균**

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

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



- **정보이득이 클 수록** 우수한 분할 속성

결정트리 학습 알고리즘

❖ 학습 데이터의 예

- 부류(class) 정보가 있는 데이터

	속성			부류
	Pattern	*Outline*	*Dot*	*Shape*
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 Δ (삼각형)

- **부류별 확률(class probability)**

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

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

- **엔트로피(entropy)**

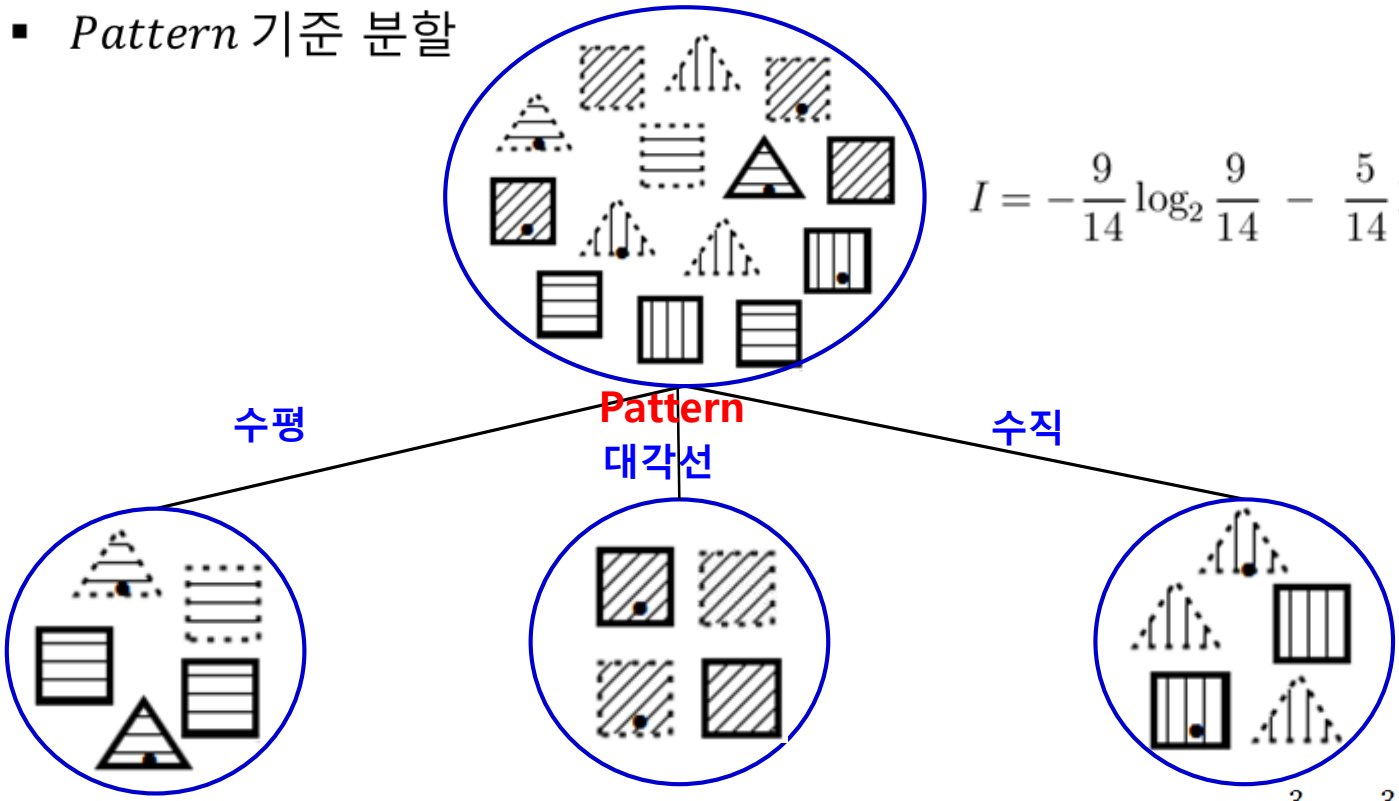
$$I = -\sum_{c} p(c) \log_2 p(c)$$

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

결정트리 학습 알고리즘

❖ 데이터 집합 분할과 정보이득

- *Pattern* 기준 분할



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

$$I_{horizontal} = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971 \quad I_{diagonal} = -\frac{0}{4}\log_2\frac{0}{4} - \frac{4}{4}\log_2\frac{4}{4} = 0.0 \quad I_{vertical} = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971$$

$$I_{res}(Pattern) = \sum p(v)I(v) = \frac{5}{14}\cdot 0.971 + \frac{4}{14}\cdot 0 + \frac{5}{14}\cdot 0.971 = 0.694$$

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

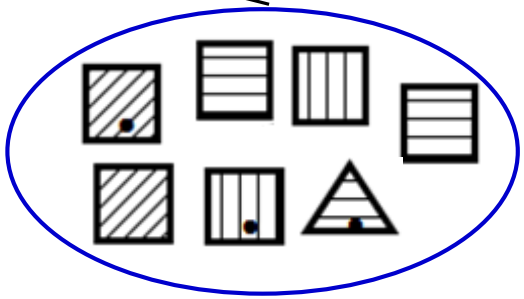
결정트리 학습 알고리즘

❖ 데이터 집합 분할과 정보이득

- *Outline* 기준 분할

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

점선 Outline 실선



$$I_{dashed} = -\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7} = 0.985$$

$$I_{solid} = -\frac{6}{7}\log_2\frac{6}{7} - \frac{1}{7}\log_2\frac{1}{7} = 0.592$$

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

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

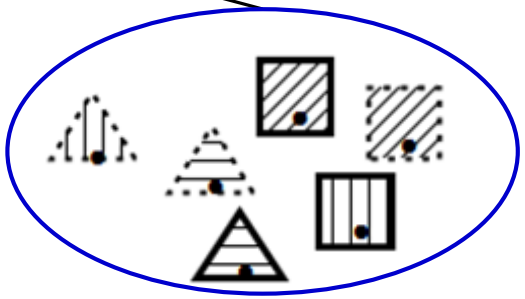
결정트리 학습 알고리즘

❖ 데이터 집합 분할과 정보이득

- *Dot* 기준 분할

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

무 Dot 유



$$I_{no} = -\frac{2}{8}\log_2\frac{2}{8} - \frac{6}{8}\log_2\frac{6}{8} = 0.811$$

$$I_{yes} = -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6} = 1.000$$

$$I_{res}(Dot) = \sum p(v)I(v) = \frac{8}{14}\cdot 0.811 + \frac{6}{14}\cdot 1.000 = 0.892$$

$$IG(Dot) = I - I_{res}(Dot) = 0.940 - 0.892 = 0.048$$

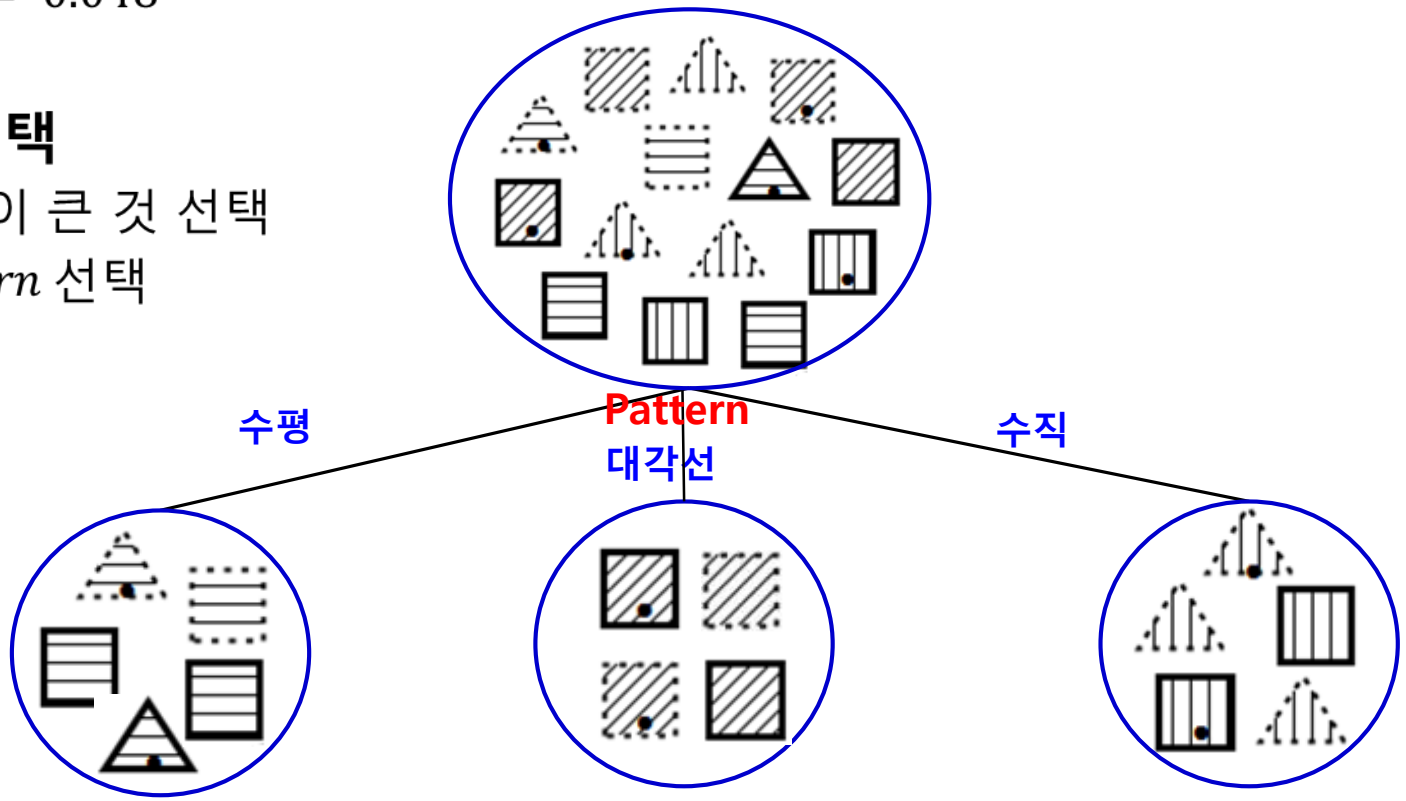
결정트리 학습 알고리즘

❖ **속성별 정보 이득**

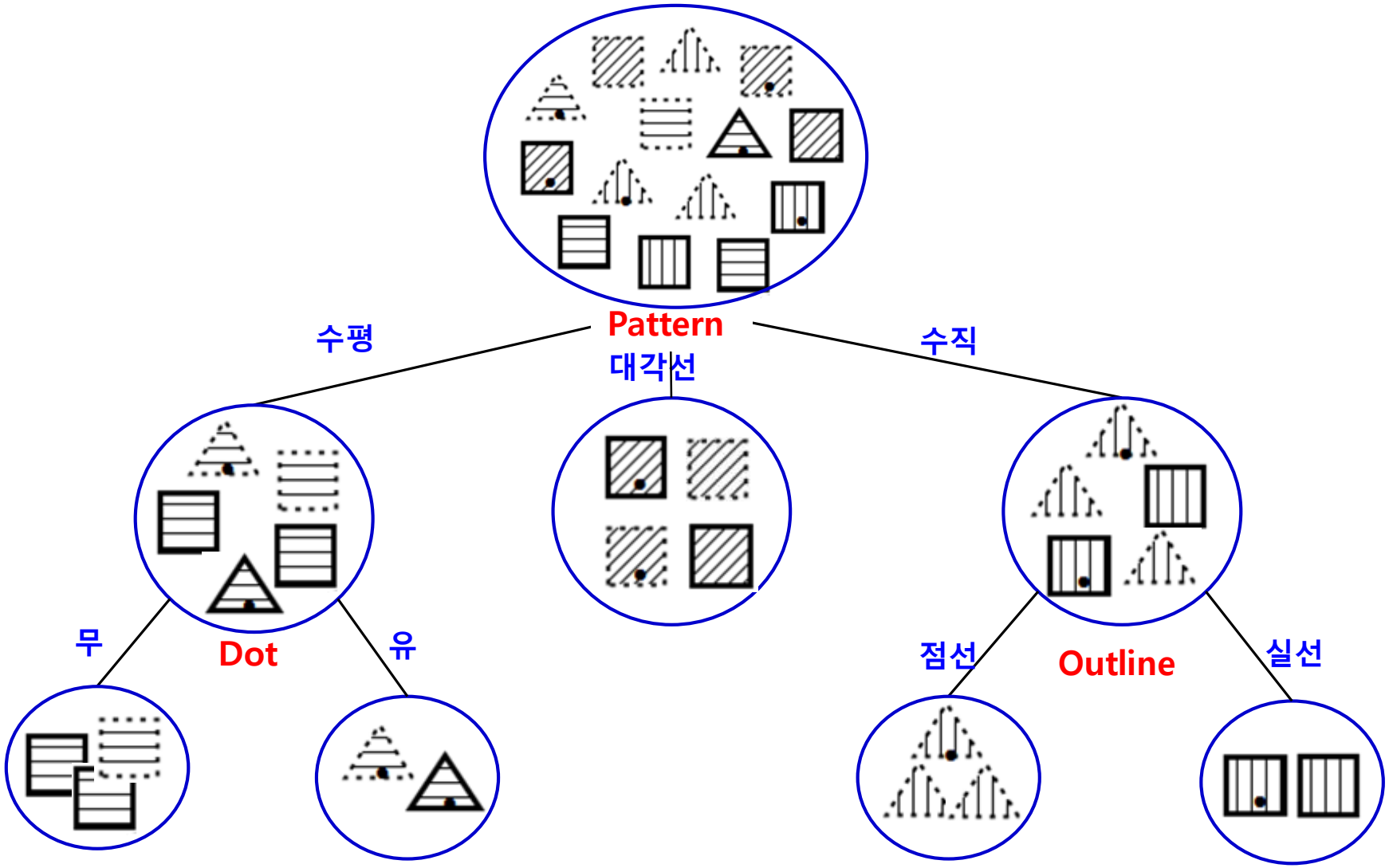
- $IG(Pattern) = 0.246$
- $IG(Outline) = 0.151$
- $IG(Dot) = 0.048$

❖ **분할속성 선택**

- 정보이득이 큰 것 선택
  - $Pattern$ 선택

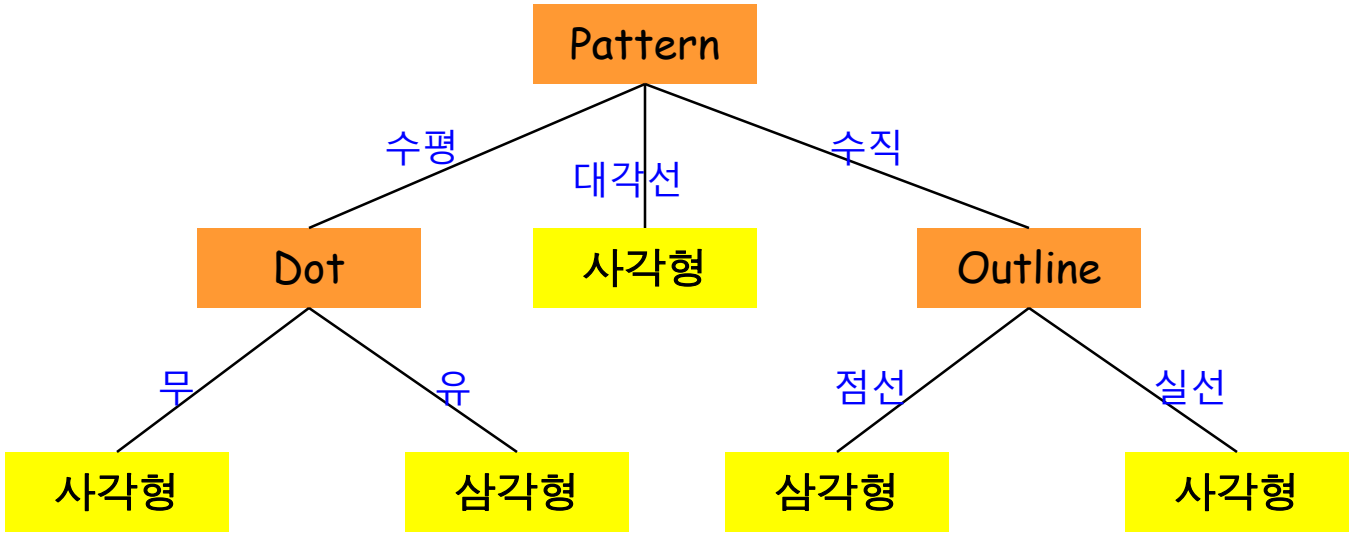


결정트리 학습 알고리즘



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) 척도를 개선한 것
  - **속성값이 많은 속성**에 대해 **불이익**

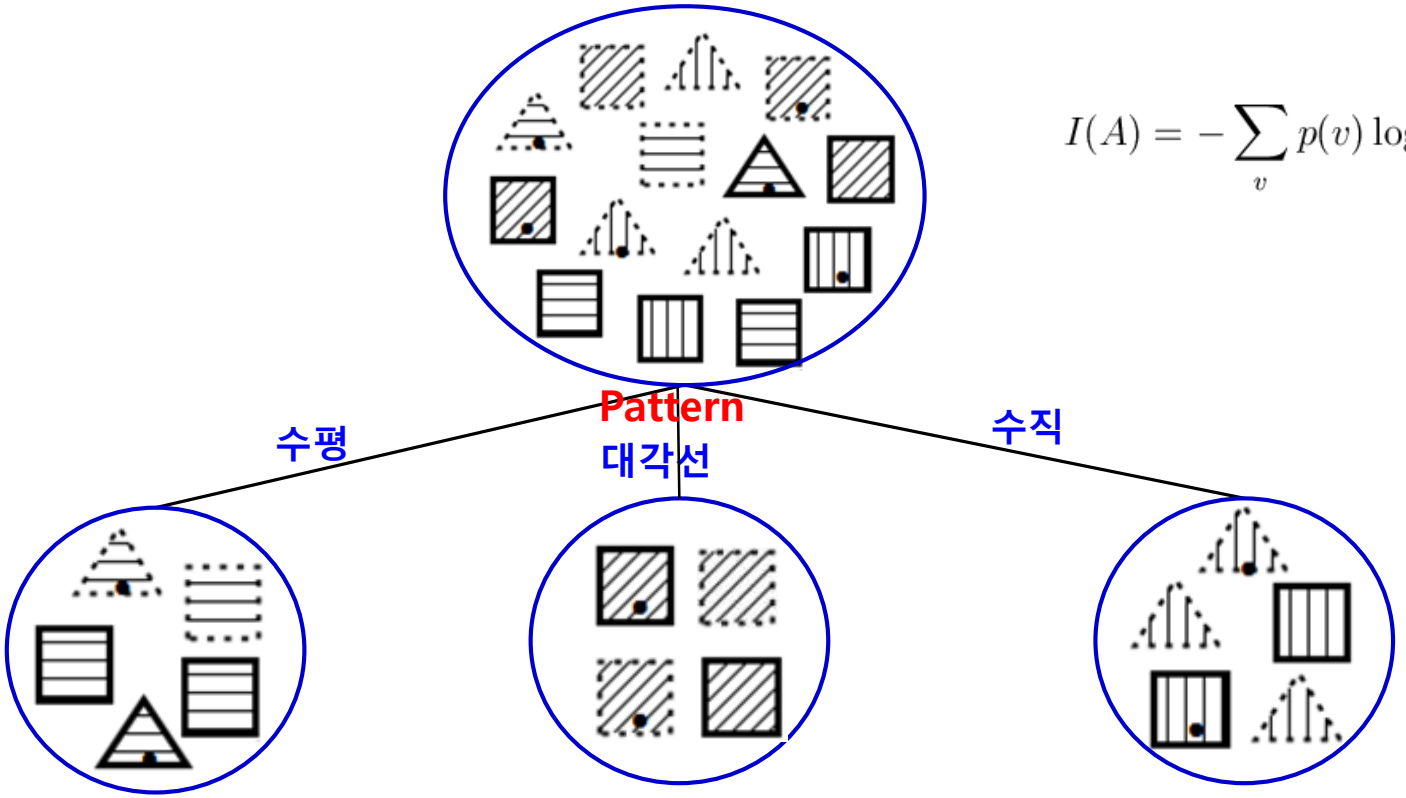
$$GainRatio(A) = \frac{IG(A)}{I(A)} = \frac{I - I_{res}(A)}{I(A)}$$

  - $I(A)$
    - **속성 A**의 **속성값**을 **부류**(class)로 **간주**하여 계산한 엔트로피
    - 속성값이 많을 수록 커지는 경향

$$I(A) = -\sum_{v} p(v)\log_2(p(v))$$

결정트리 학습 알고리즘

❖ 정보이득 비



$$I(A) = -\sum_{v} p(v)\log_2(p(v))$$

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

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

$$GainRatio(Pattern) = \frac{IG(Pattern)}{I(Pattern)} = \frac{0.940 - 0.694}{1.58} = 0.156$$

결정트리 학습 알고리즘

❖ **정보이득 vs 정보이득 비**

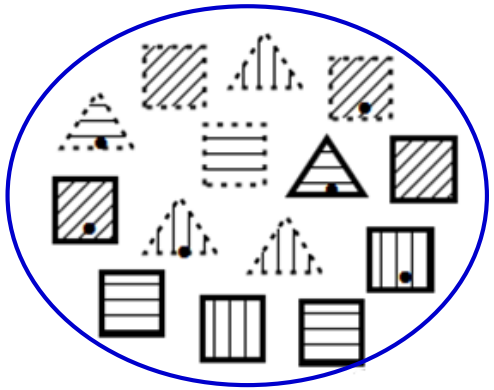
속성	속성의 개수	정보 이득	정보 이득비
Pattern	3	0.247	0.156
Outline	2	0.152	0.152
Dot	2	0.048	0.049

결정트리 학습 알고리즘

❖ 지니 지수(Gini index)

- 데이터 집합에 대한 **지니 값**
  - i, j가 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 지수값 가중평균**

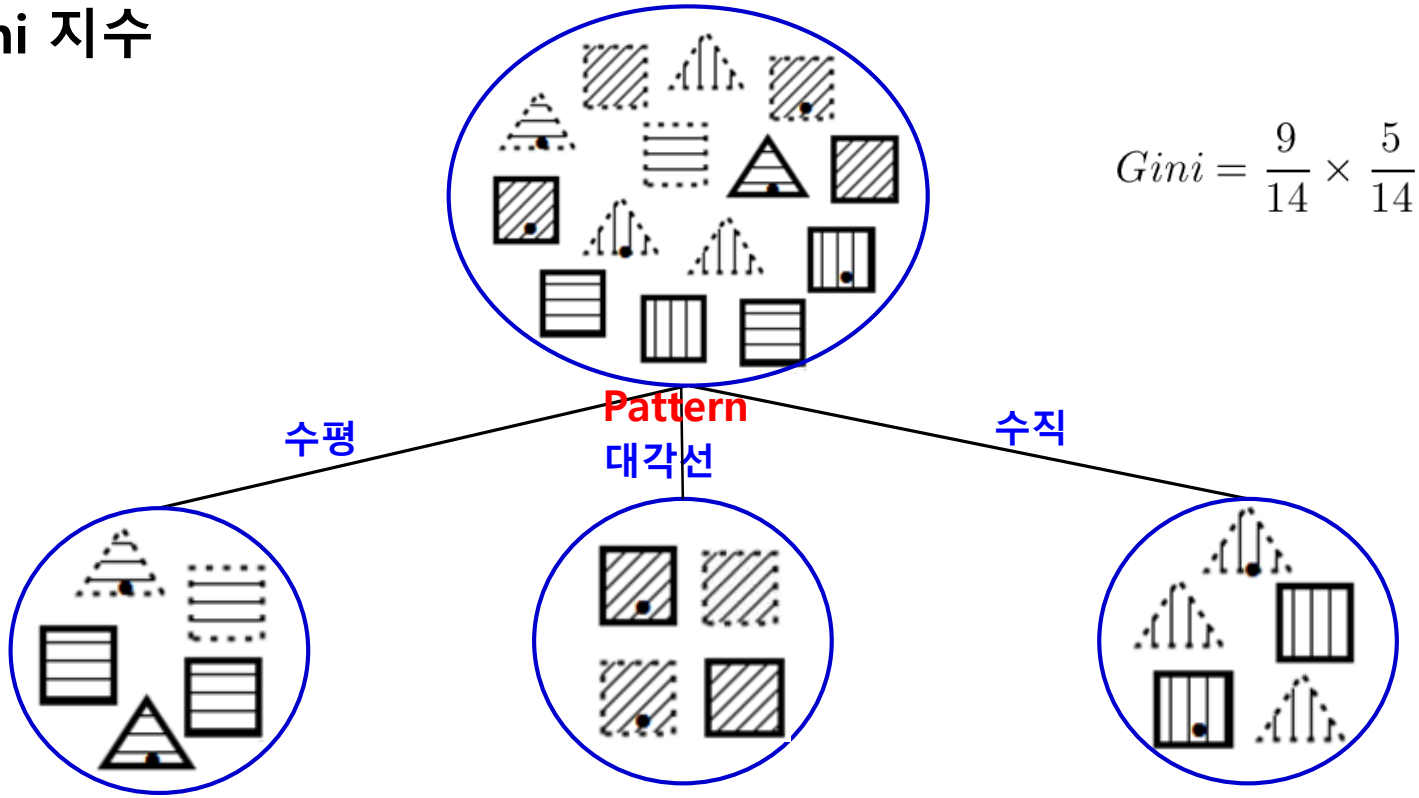
$$Gini(A) = \sum_{v} p(v) \sum_{i \neq j} p(i|v)p(j|v)$$

- **지니 지수 이득** (gini index gain)

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

결정트리 학습 알고리즘

❖ **Gini 지수**



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

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

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

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

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) 속성값**을 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

❖ [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]
```

1.2 결정트리를 이용한 회귀

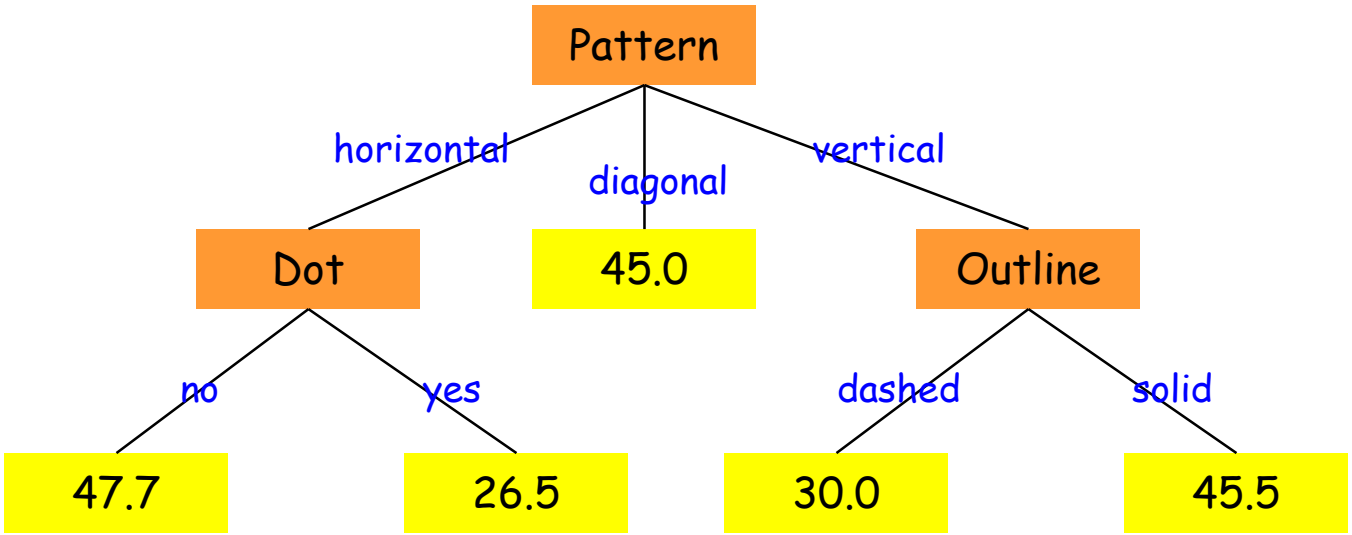
❖ 회귀(regression)를 위한 결정트리
- 출력값이 수치값

표 4.5 도형 면적에 대한 데이터

	속성			면적
	Pattern	*Outline*	*Dot*	*Area*
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**
  - **단말노드**가 부류(class)가 아닌 **수치값**(numerical value)임
  - 해당 조건을 만족하는 것들이 가지는 **대표값**

- **분할 속성 선택**
  - **표준편차 축소(reduction of standard deviation)** $\boldsymbol{SDR}$를 최대로 하는 속성 선택

$$\boldsymbol{SDR}(\boldsymbol{A}) = \boldsymbol{SD} - \boldsymbol{SD}(\boldsymbol{A})$$

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

– $\boldsymbol{SD}(A)$

» 속성 A를 기준으로 **분할 후**의 **부분 집합별 표준표차**의 **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
Pattern	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
```

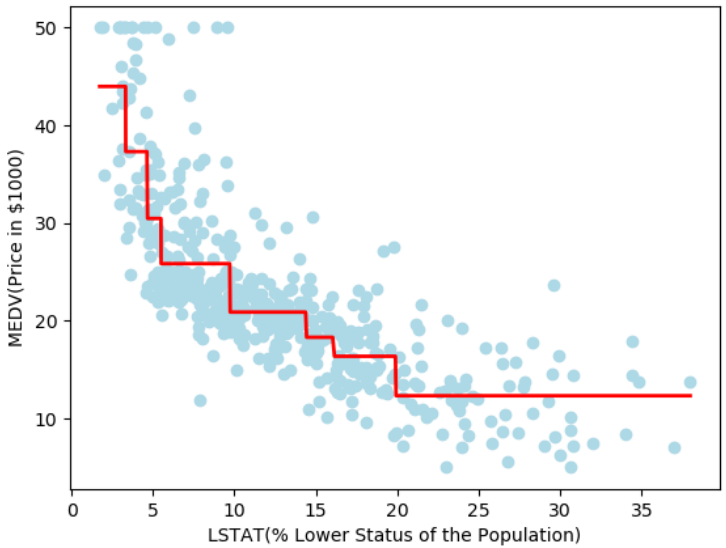
https://archive.ics.uci.edu/ml/machine-learning-databases/housing/housing.data

```
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 것을 선택하시오.**

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 않는 것을 선택하시오.**

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.

Quiz

❖ **결정트리에 대한 설명으로 옳지 않은 것을 선택하시오.**

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.

Quiz

1. 결정트리를 생성할 때 분할속성 선택에 따라 결정트리의 크기가 영향을 받는다. (O,X)
2. 엔트로피가 큰 데이터 집합일수록 동일한 부류에 속하는 경향이 크다. (O,X)
3. 정보이득이 큰 속성이 일반적으로 분할속성으로서 바람직하다. (O,X)
4. 정보이득비는 속성값의 개수가 많은 속성이 분할속성으로 선택될 가능성을 높게 한다. (O,X)
5. 결정트리는 분류 문제에만 사용된다. (O,X)