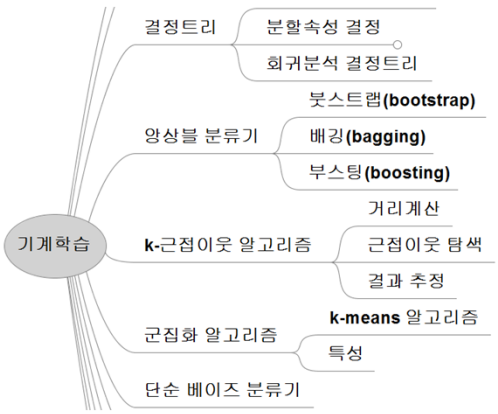
기계 학습

Part Ⅱ

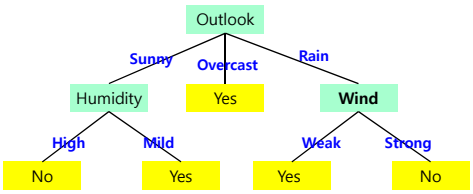
충북대학교 소프트웨어학과
이건명



3. 결정트리(decision tree)

❖ **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

(출처: Machine Learning, Tom Mitchell, 1995)

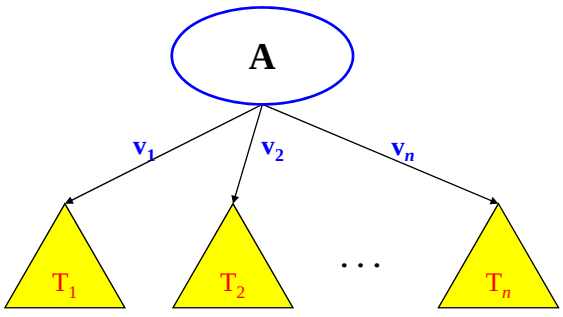
IF Outlook = Sunny **AND** Humidity = High **THEN** Answer = No

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

결정트리(decision tree)

❖ **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**



분할 속성 A

A의 값

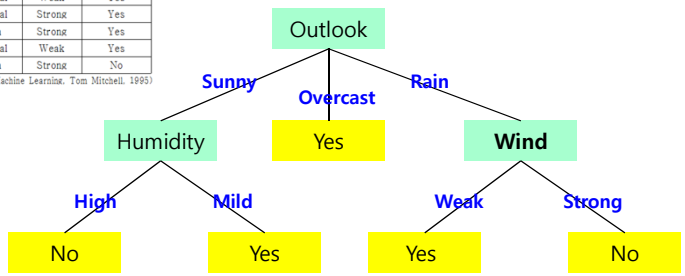
서브 트리

결정트리(decision tree)

❖ **결정 트리** (decision tree)
- 간단한 트리

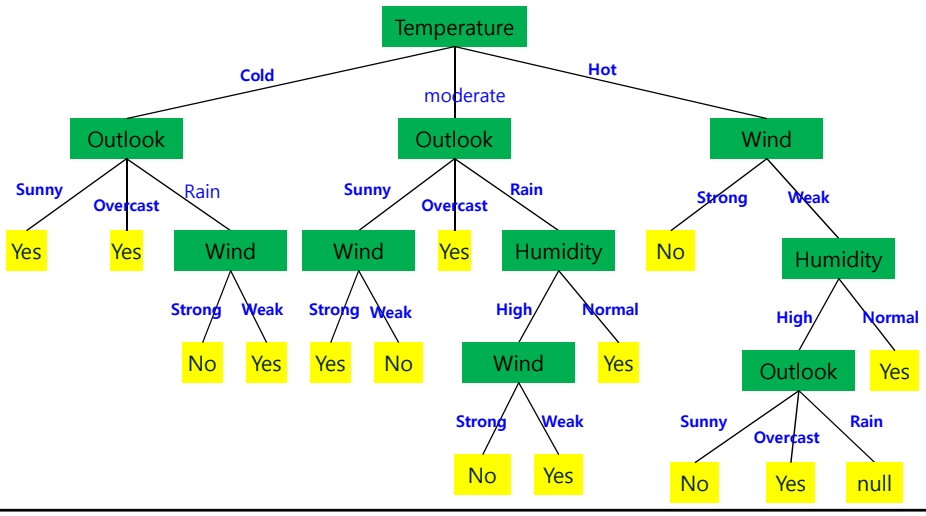
Day 날짜	Outlook 조망	Temperature 기온	Humidity 습도	Wind 바람	PlayTennis 테니스 여부
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny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

(출처: Machine Learning, Tom Mitchell, 1995)



결정트리(decision tree)

❖ **결정 트리** (decision tree)
- 복잡한 트리



결정트리(decision tree)

❖ **분할 속성(splitting attribute) 결정**

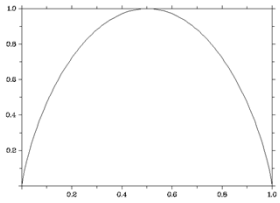
- 어떤 속성을 선택하는 것이 효율적인가
  - **분할한 결과**가 가능하면 동질적인(pure) 것으로 만드는 속성 선택

- **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

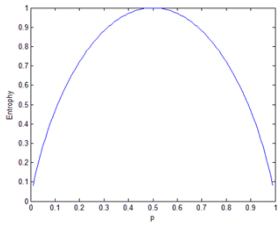
  - 2개 부류가 있는 경우 엔트로피

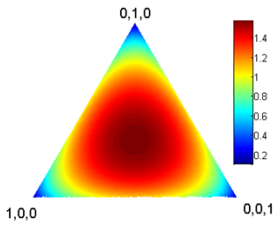


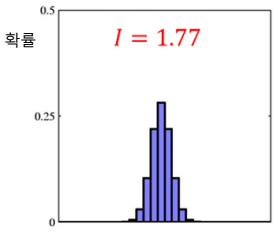
결정트리(decision tree)

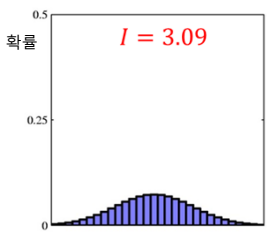
❖ **엔트로피의 특성**

- 섞인 정도가 클 수록 큰 값









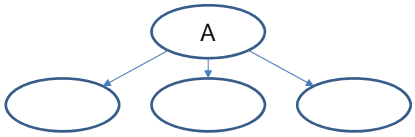
결정트리(decision tree)

❖ **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(decision tree)

❖ **학습 데이터의 예**

- 부류(class) 정보가 있는 데이터

	속성			부류
	Pattern	Outline	Dot	Shape
1	vertical	dashed	no	triangle
2	vertical	dashed	yes	triangle
3	diagonal	dashed	no	square
4	horizontal	dashed	no	square
5	horizontal	solid	no	square
6	horizontal	solid	yes	triangle
7	vertical	solid	no	square
8	vertical	dashed	no	triangle
9	diagonal	solid	yes	square
10	horizontal	solid	no	square
11	vertical	solid	yes	square
12	diagonal	dashed	yes	square
13	diagonal	solid	no	square
14	horizontal	dashed	yes	triangle

결정트리(decision tree)

❖ 엔트로피 계산

- 9 □ (square)
- 5 Δ (triangle)

- **부류별 확률(class probability)**

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

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

- **엔트로피(entropy)**

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

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

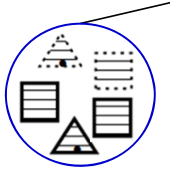
결정트리(decision tree)

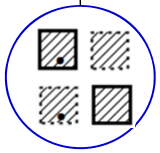
❖ 데이터 집합 분할과 정보이득

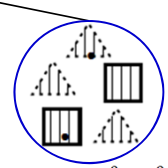
▪ *Pattern* 기준 분할

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

horizontal | Pattern | diagonal | vertical







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

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

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

결정트리(decision tree)

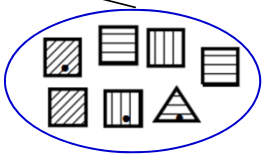
❖ 데이터 집합 분할과 정보이득

- *Outline* 기준 분할

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

dashed — Outline — solid

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



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

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

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

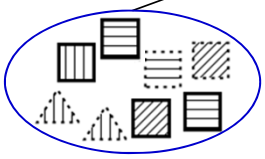
결정트리(decision tree)

❖ 데이터 집합 분할과 정보이득

- *Dot* 기준 분할

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

no — Dot — yes



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

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

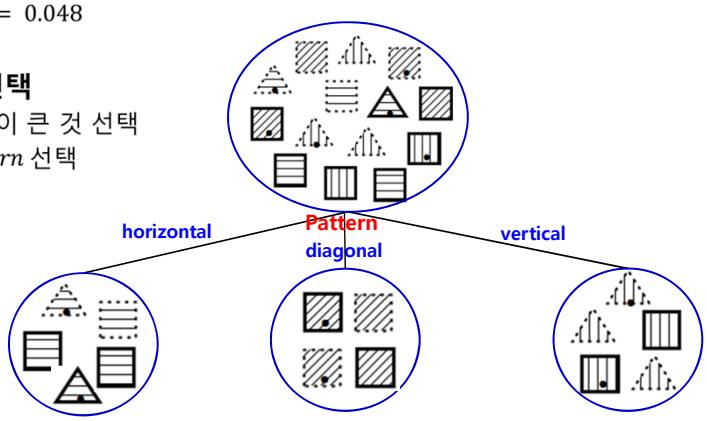
$$I_{res}(Dot) = \sum p(v)I(v) = \frac{8}{14}\cdot 0.811 + \frac{6}{14}\cdot 1.000 = 0.892$$

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

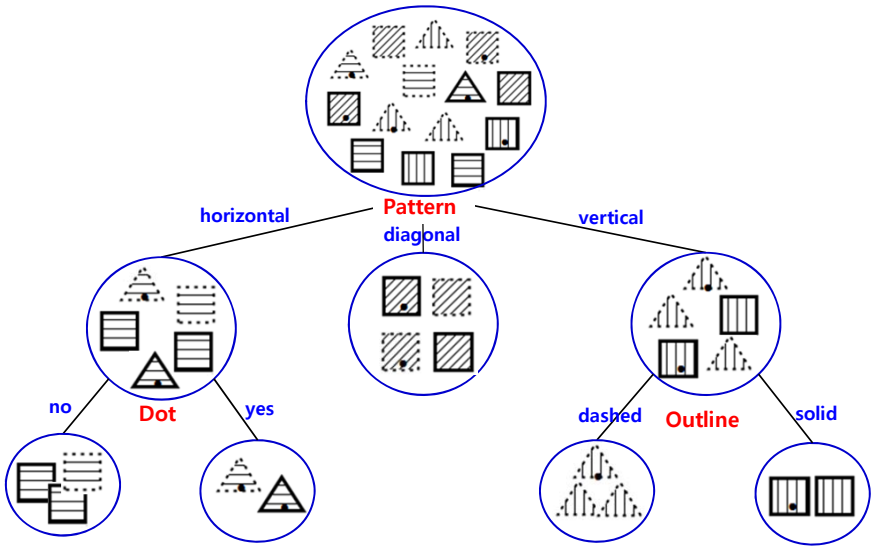
결정트리(decision tree)

❖ 속성별 정보 이득
- $IG(Pattern) = 0.246$
- $IG(Outline) = 0.151$
- $IG(Dot) = 0.048$

❖ 분할속성 선택
- 정보이득이 큰 것 선택
  - $Pattern$ 선택

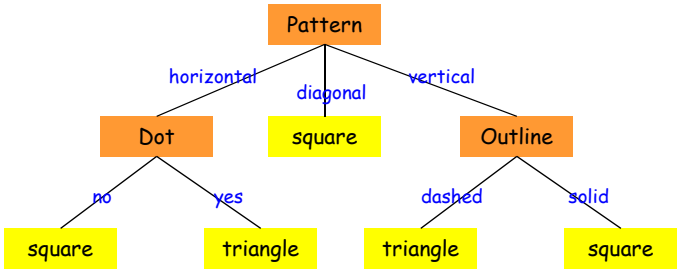


결정트리(decision tree)



결정트리(decision tree)

❖ 최종 결정트리



결정트리(decision tree)

❖ 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(decision tree)

❖ **정보이득 비(information gain ratio) 척도**

- 정보이득(information gain) 척도를 개선한 것
  - **속성값이 많은 속성**에 대해 **불이익**

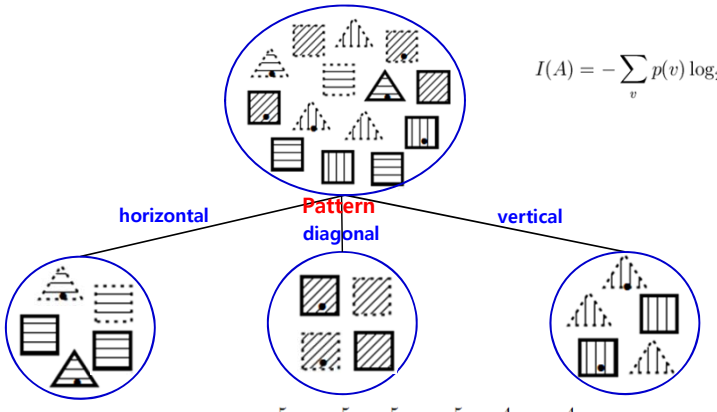
$$GainRatio(A) = \frac{IG(A)}{I(A)} = \frac{I - I_{res}(A)}{I(A)}$$

  - $I(A)$
    - **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(decision tree)

❖ **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(decision tree)

❖ **정보이득** vs **정보이득 비**

A	|v(A)|	IG(A)	GainRatio(A)
Pattern	3	0.247	0.156
Outline	2	0.152	0.152
Dot	2	0.048	0.049

결정트리(decision tree)

❖ **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**

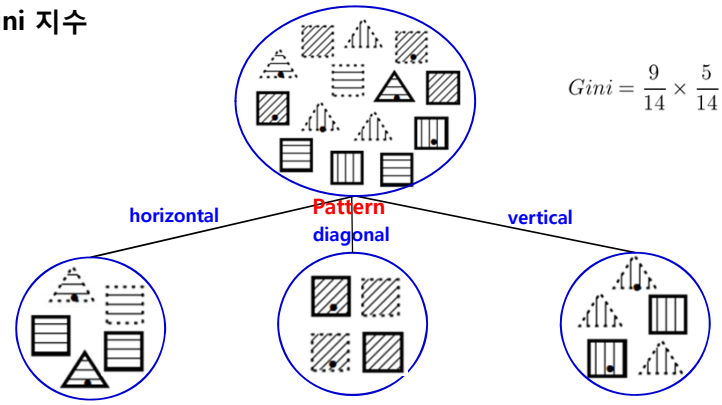
$$Gini(A) = \sum_{v} p(v) \sum_{i \neq j} p(i|v)p(j|v)$$

- **지니 지수 이득** (gini index gain)

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

결정트리(decision tree)

❖ **Gini 지수**



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

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

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

$$GiniGain(Pattern) = 0.230 - 0.171 = 0.058$$

결정트리(decision tree)

❖ **분할속성 평가 척도 비교**

A	Gain(A)	GainRatio(A)	GiniGain(A)
Pattern	0.247	0.156	0.058
Outline	0.152	0.152	0.046
Dot	0.048	0.049	0.015

결정트리(decision tree)

❖ **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

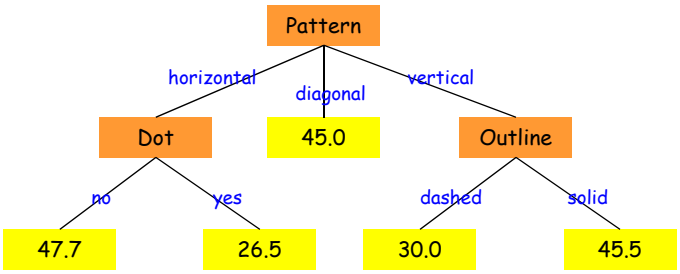
결정트리(decision tree)

❖ **회귀분석(regression analysis)를 위한 결정트리**
- **출력값이 수치값**

	속성			출력
	Pattern	Outline	Dot	Area
1	vertical	dashed	no	26
2	vertical	dashed	yes	30
3	diagonal	dashed	no	48
4	horizontal	dashed	no	46
5	horizontal	solid	no	62
6	horizontal	solid	yes	23
7	vertical	solid	no	43
8	vertical	dashed	no	36
9	diagonal	solid	yes	38
10	horizontal	solid	no	48
11	vertical	solid	yes	48
12	diagonal	dashed	yes	62
13	diagonal	solid	no	44
14	horizontal	dashed	yes	30

결정트리(decision tree)

❖ 회귀분석(regression analysis)를 위한 결정트리



결정트리(decision tree)

❖ 회귀분석(regression analysis)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) SDR를 최대로 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 후**의 **부분 집합별 표준편차**의 **가중평균**

결정트리(decision tree)

❖ **회귀분석(regression analysis)를 위한 결정트리**

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

← $SD = 9.67$

		Area의 표준편차	개수
$Pattern$	horizontal	12.15	5
	vertical	9.36	5
	diagonal	5.77	4
			14

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

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

4. 앙상블 분류기

❖ **앙상블 분류기 (ensemble classifier)**

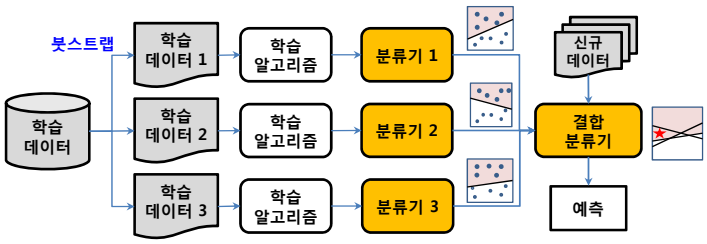
- 주어진 학습 데이터 집합에 대해서 **여러 개의 서로 다른 분류기**를 만들고, 이들 분류기의 판정 결과를 **투표 방식**(voting method)이나 **가중치 투표 방식**(weighted voting method)으로 결합
- **붓스트랩**(bootstrap)
  - 주어진 학습 데이터 집합에서 **복원추출**(resampling with replacement)하여 **다수**의 **학습 데이터 집합**을 만들어내는 기법

- **배깅**(bagging, **b**ootstrap **agg**regat**ing**)
- **부스팅**(boosting)

앙상블 분류기

❖ **배깅**(bagging, **b**ootstrap **agg**regat**ing**)

- **붓스트랩**을 통해 **여러 개**의 **학습 데이터 집합**을 만들고, 각 학습 데이터 집합별로 **분류기**를 만들어, 이들이 **투표**나 **가중치 투표**를 하여 최종 판정을 하는 기법

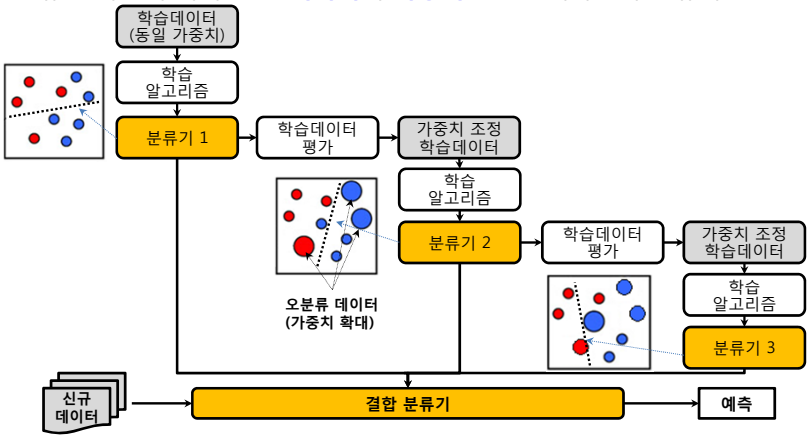


- **임의 숲**(**random forest**)
  - 분류기로 **결정트리**를 사용하는 배깅 기법

앙상블 분류기

❖ **부스팅**(boosting)

- k개의 분류기를 **순차적**으로 만들어 가는 **앙상블 분류기** 생성 방법
- 분류 정확도에 따라 **학습 데이터**에 **가중치**를 **변경**해가면서 분류기 생성

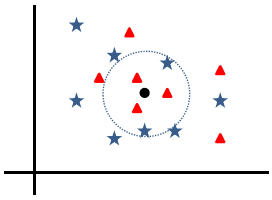


- **에이더부스트**(**AdaBoost**)

5. k-근접이웃 알고리즘

❖ **k-근접이웃 (k-nearest neighbor, KNN) 알고리즘**

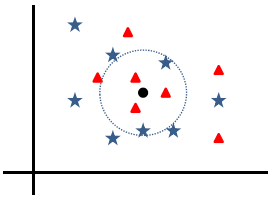
- **(입력, 결과)**가 있는 데이터들이 주어진 상황에서,
  새로운 입력에 대한 결과를 추정할 때
  결과를 아는 **최근접**한 **k개**의 **데이터**에 대한 결과정보를 이용하는 방법



- 질의(query)와 데이터간의 **거리 계산**
- 효율적으로 **근접이웃 탐색**
- 근접 이웃 k개로 부터 **결과**를 **추정**

k-근접이웃 알고리즘

❖ **k-nearest neighbor (KNN) 알고리즘 – cont.**



- 데이터간의 **거리 계산**
  - **수치 데이터**의 경우
    - **유클리디언 거리**(Euclidian distance)

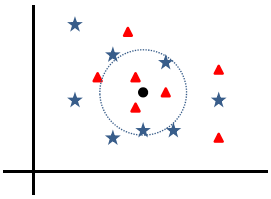
$$X = (x_1, x_2, \ldots, x_n) \quad Y = (y_1, y_2, \ldots, y_n)$$

$$d = (x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2$$

    - 응용분야의 특성에 맞춰 개발
  - **범주형 데이터**가 포함된 경우
    - 응용분야의 특성에 맞춰 개발

k-근접이웃 알고리즘

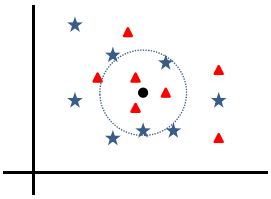
❖ **k-nearest neighbor (KNN) 알고리즘 – cont.**



- 효율적인 **근접 이웃 탐색**
  - 데이터의 개수가 많아지면 계산시간 증가 문제
  - **색인**(indexing) 자료구조 사용
    - R-트리, k-d 트리 등

k-근접이웃 알고리즘

❖ **k-nearest neighbor (KNN) 알고리즘 – cont.**



- 최근접 k개로 부터 **결과**를 **추정**하는 방법
  - **분류**
    - 출력이 **범주형** 값
    - **다수결 투표**(majority voting) : 개수가 많은 범주 선택
  - **회귀분석**
    - 출력이 수치형 값
    - 평균 : 최근접 k개의 평균값
    - **가중합**(weighted sum) : 거리에 반비례하는 가중치 사용

k-근접이웃 알고리즘

❖ **k-nearest neighbor (KNN) 알고리즘 – cont.**

- **특징**
  - **학습단계**에서는 실질적인 학습이 일어나지 않고 **데이터**만 **저장**
    - 학습데이터가 크면 **메모리 문제**
    - **게으른 학습(lazy learning)**
  - 새로운 데이터가 주어지면 저장된 데이터를 이용하여 학습
    - 시간이 많이 걸릴 수 있음

6. 군집화 알고리즘

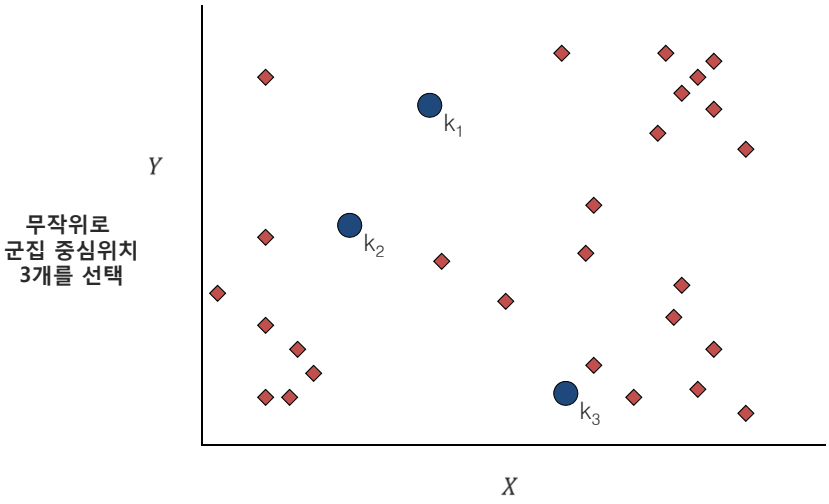
❖ **군집화(clustering) 알고리즘**

- 데이터를 유사한 것들끼리 모우는 것
- **군집내**의 **유사도(similarity)**는 **크게**, **군집간**의 **유사도**는 **작게**

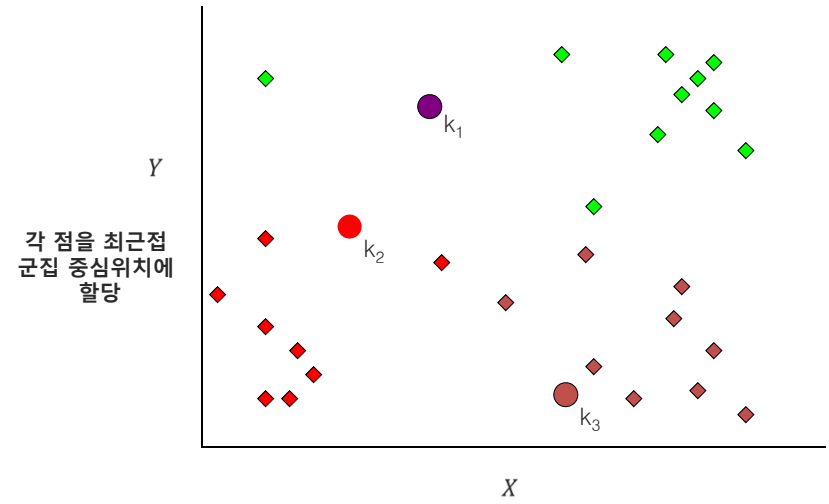
❖ **k-means 알고리즘**

- 군집화 알고리즘
- **군집화 과정**
  1. **군집**의 **중심 위치** 선정
  2. 군집 중심을 기준으로 **군집 재구성**
  3. 군집별 **평균 위치** 결정
  4. 군집 평균 위치로 **군집 중심 조정**
  5. 수렴할 때까지 2-4 과정 **반복**

군집화 알고리즘 : K-means 실행과정 1

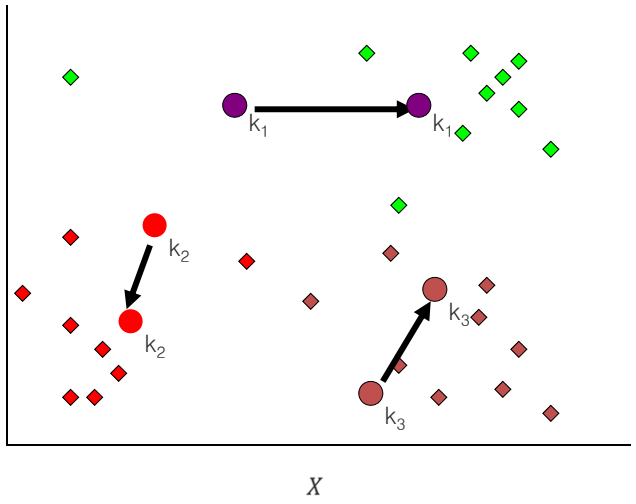


군집화 알고리즘 : K-means 실행과정 2



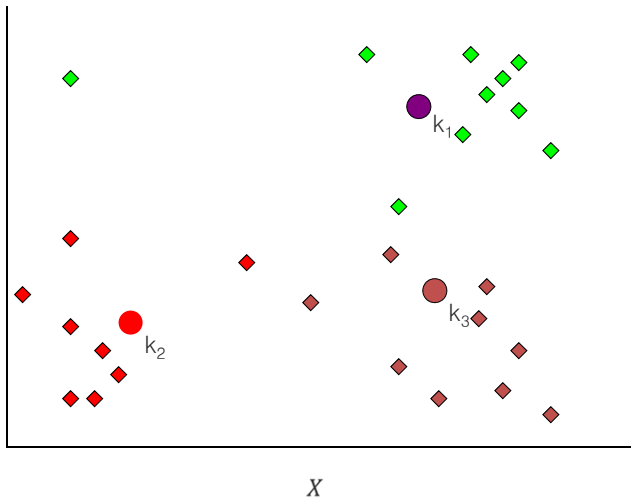
군집화 알고리즘 : K-means 실행과정 3

군집 중심 위치를 군집의 평균 위치로 이동

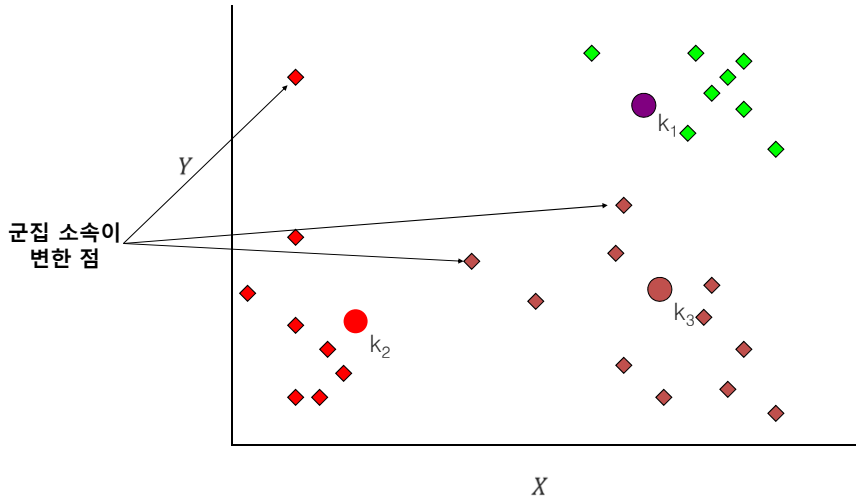


군집화 알고리즘 : K-means 실행과정 4

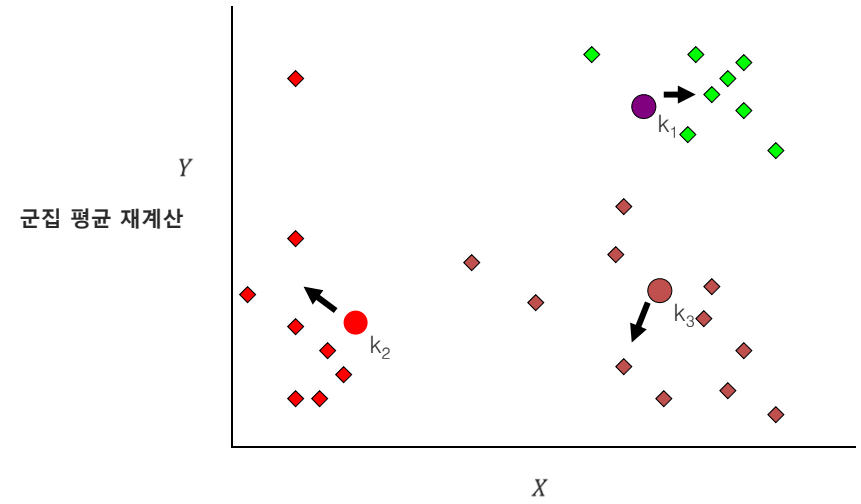
새로운 군집 중심을 기준으로 각 점의 소속을 재할당



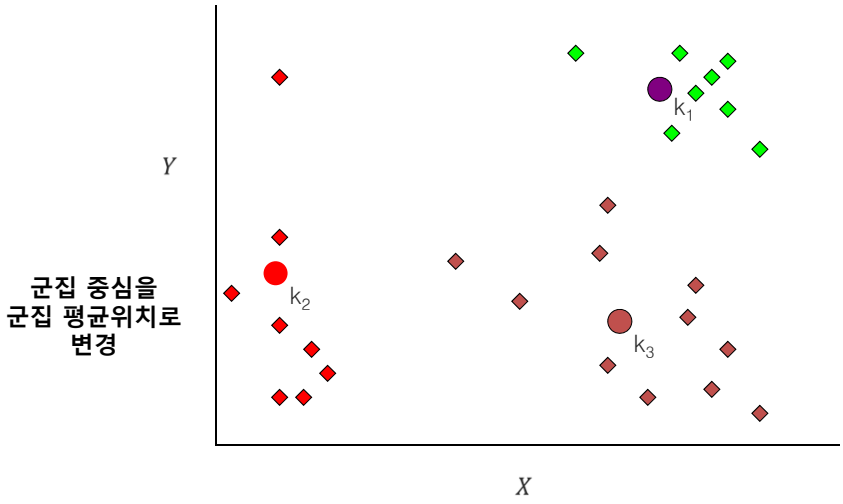
군집화 알고리즘 : K-means 실행과정 5



군집화 알고리즘 : K-means 실행과정 6



군집화 알고리즘 : K-means 실행과정 7



군집 중심을
군집 평균위치로
변경

군집화 알고리즘 : K-means 알고리즘

❖ **k-means 알고리즘**

- i 번째 클러스터의 **중심**을 μ_i, 클러스터에 속하는 **점의 집합** S_i을 라고 할 때, **전체 분산**

$$V = \sum_{i=1}^{k} \sum_{j \in S_i} |x_j - \mu_i|^2$$

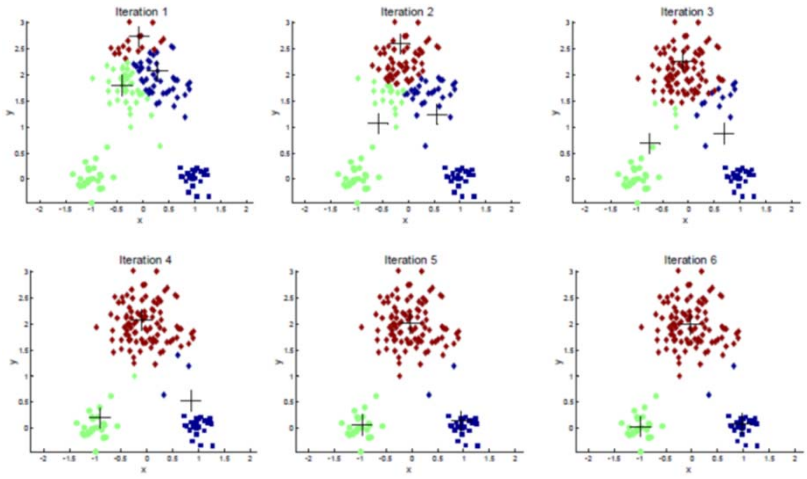
- **분산값 V**을 **최소화**하는 S_i를 찾는 것이 알고리즘의 목표

- **과정**
  1. 우선 초기의 μ_i 를 임의로 설정
  2. 다음 두 단계를 클러스터가 변하지 않을 때까지 반복
     - I. **클러스터 설정**: 각 점에 대해, 그 점에서 가장 가까운 클러스터를 찾아 배당한다.
     - II. **클러스터 중심 재조정**: μ_i를 각 클러스터에 있는 점들의 평균값으로 재설정해준다.

- **특성**
  - **군집의 개수** k는 **미리 지정**
  - **초기 군집 위치**에 **민감**

군집화 알고리즘 : K-means 알고리즘

❖ 초기 중심값에 대해 민감한 군집화 결과



https://sites.google.com/site/myecodriving/k-means-ju-lei-fen-xi

7. 단순 베이즈 분류기

❖ **단순 베이즈 분류기**(naïve Bayes classifier)

- **부류**(class) 결정지식을 **조건부 확률**(conditional probability)로 결정
  - $P(c|x_1, x_2, \cdots . x_n)$: 속성값에 대한 부류의 조건부 확률
    - c : 부류
    - x_i : 속성값
- **베이즈 정리** (Bayes theorem)

사후 확률 / 가능도 / 사전 확률

$$P(c|x_1, x_2, \cdots . x_n) = \frac{P(x_1, x_2, \cdots, x_n|c)P(c)}{P(x_1, x_2, \cdots, x_n)}$$

증거

- **가능도**(likelihood)의 **조건부 독립**(conditional independence) 가정

$$P(x_1, x_2, \cdots . x_n|c) = P(x_1|c)\ P(x_2|c) \cdots P(x_n|c)$$

$$P(c|x_1, x_2, \cdots, x_n) = \frac{P(x_1|c)\ P(x_2|c) \cdots P(x_n|c)P(c)}{P(x_1, x_2, \cdots, x_n)}$$

단순 베이즈 분류기

❖ 단순 베이즈 분류기 – cont.

	속성			부류
	Pattern	Outline	Dot	Shape
1	vertical	dashed	no	triangle
2	vertical	dashed	yes	triangle
3	diagonal	dashed	no	square
4	horizontal	dashed	no	square
5	horizontal	solid	no	square
6	horizontal	solid	yes	triangle
7	vertical	solid	no	square
8	vertical	dashed	no	triangle
9	diagonal	solid	yes	square
10	horizontal	solid	no	square
11	vertical	solid	yes	square
12	diagonal	dashed	yes	square
13	diagonal	solid	no	square
14	horizontal	dashed	yes	triangle

$P(triangle) = \frac{5}{14}$ $P(square) = \frac{9}{14}$

$P(vertical|triangle) = \frac{3}{5}$

$P(horizontal|triangle) = \frac{2}{5}$

$P(diagonal|triangle) = \frac{1}{5}$

$P(dashed|triangle) = \frac{4}{5}$

$P(solid|triangle) = \frac{1}{5}$

$P(yes|triangle) = \frac{3}{5}$

$P(no|triangle) = \frac{2}{5}$

$P(vertical, dashed, no) = \frac{1}{14}$

$$P(triangle|vertical, dashed, no) = \frac{P(vertical|triangle)\ P(dashed|triangle)P(no|triangle)P(triangle)}{P(vertical,dashed,no)} = \frac{3/5{\cdot}4/5{\cdot}2/5{\cdot}5/14}{1/14} = 0.96$$