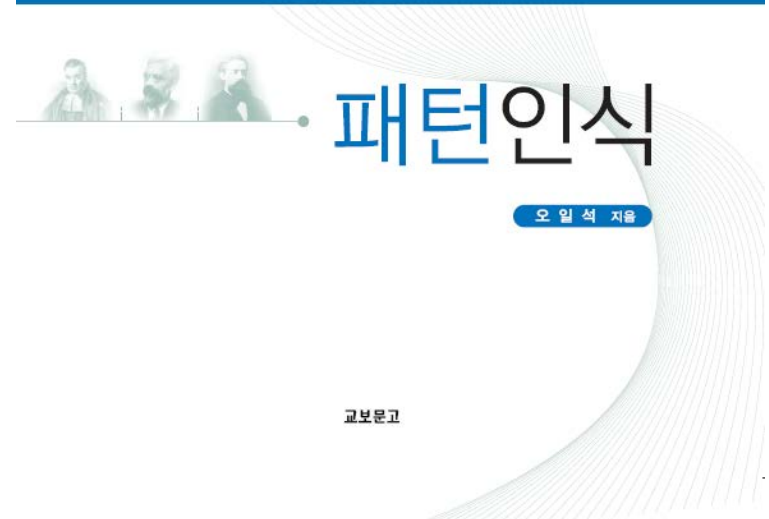


## 10장. 군집화



## 들어가는 말

재미있는 응용 시나리오를 생각해 보자. 지난 몇 년간 온라인 쇼핑몰을 성공적으로 운영했는데 이제 제 2의 도약을 꿈꾸고 있다. 그 동안 한 종류의 홍보 팜플렛을 만들어 발송했는데 이제부터는 고객의 취향을 분석하여 4~6종의 팜플렛을 만들어 맞춤 홍보를 하려 한다.<sup>1</sup> 일종의 개인화(personalization) 홍보 전략이다. 고객에 대한 각종 정보는 데이터베이스에 저장되어 있어 이것을 기초 자료로 활용하면 된다. 고객 정보는 월평균 구매액, 선호하는 물품의 종류와 수준, 결제 방법, 반품 성향, 직업, 성별, 나이, 거주 지역 등 아주 다양하다. 하지만 수백 만 명이나 되는 고객을 어떻게 4~6개의 그룹으로 분류할 수 있을까?

- 패턴인식 문제로 공식화 가능
  - 고객이 샘플, 샘플은 **특징 벡터  $\mathbf{x}=(x_1, x_2, \dots, x_d)^T$ 로 표현**
  - 직업, 월평균 구매액 등이 특징이 됨
  - 유사한 (거리가 가까운) 샘플 집합을 군집이라 함
- 군집화(clustering) 구현에는 두 가지 필요
  - 1) **거리 척도**, 2) 유사한 샘플을 **군집으로 만드는 알고리즘**

## 들어가는 말

### ■ 지도 학습 과 비지도 학습

#### □ 지도 학습 supervised learning

- 2~7장에서 공부한 분류기 학습 (MLP, SVM, HMM 등)
- 각 샘플이 그가 속한 부류를 알고 있다.

(훈련 집합  $X=\{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \dots, (\mathbf{x}_N, t_N)\}$ 으로 표기)

#### □ 비지도 학습 unsupervised learning

- 샘플은 부류 정보가 없다. ( $X=\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ 으로 표기)
- 군집화는 비지도 학습에 해당
- **군집(K)이 몇 개인지 모르는 경우도 많다.**
- 군집화를 부류 발견 작업이라고도 부른다.

## 들어가는 말

### ■ 군집화의 특성

- 주관성: 군집화 결과의 품질은 응용이 처한 상황과 요구 사항에 따라 다름

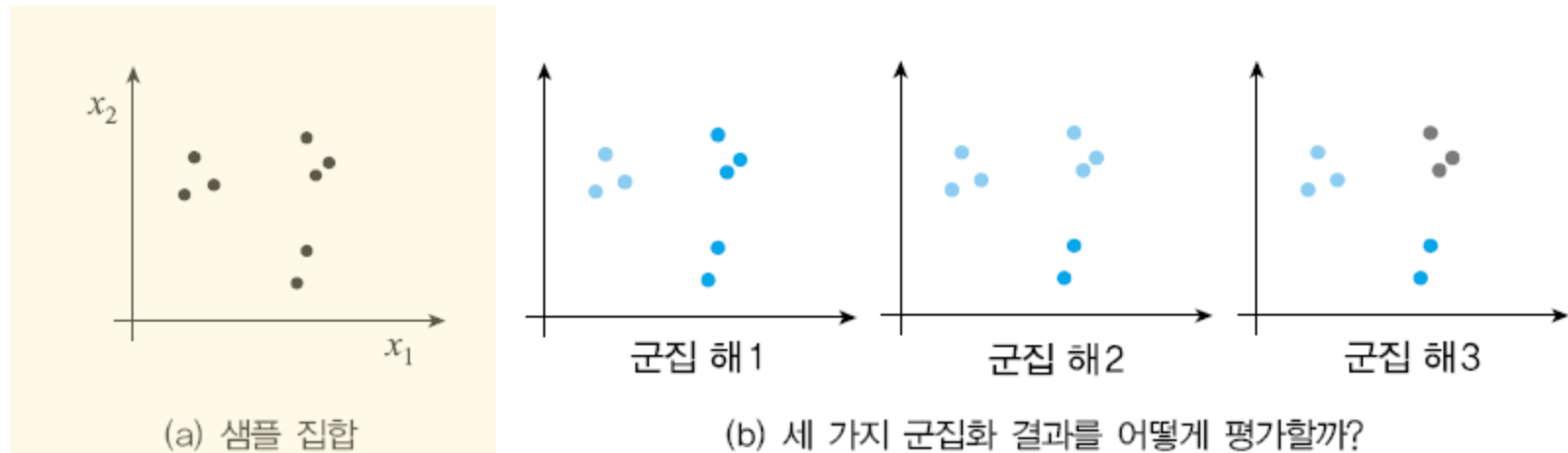


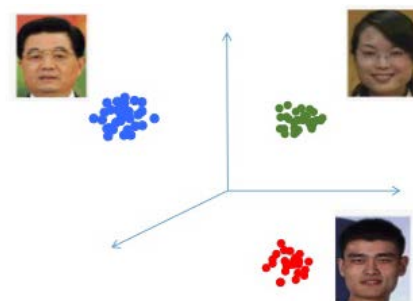
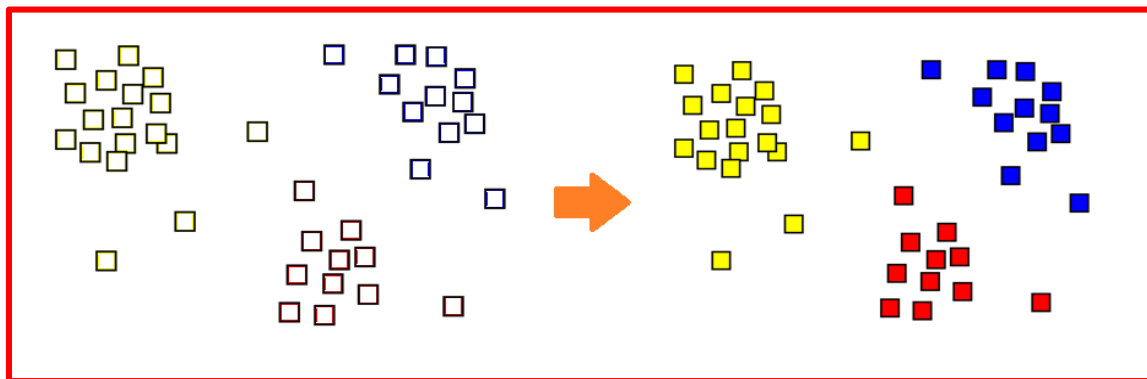
그림 10.1 군집화의 주관성

**군집(K)이 몇 개인지 모르는 경우도 많다.**

## 10.1 정의

- 군집화란? 샘플 집합  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ 이 주어졌다고 하자.  $\mathbf{x}_i$ 는  $i$  번째 샘플로  $d$  차원 특징 벡터  $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{id})^T$ 로 표기한다. 이 입력에 대해 아래 조건을 만족하는  $k$  개의 군집으로 구성되는 군집 해 clustering solution  $C = \{c_1, c_2, \dots, c_k\}$ 를 찾아라. 보통 군집의 개수  $k$ 는  $N$ 에 비해 매우 작다. 상황에 따라  $k$ 가 주어지는 경우도 있고 그렇지 않고 이 값을 찾아야 하는 경우도 있다.

$$\left. \begin{aligned} c_i &\neq \emptyset, i = 1, \dots, k \\ \cup_{i=1, k} c_i &= X \\ c_i \cap c_j &= \emptyset, i \neq j \end{aligned} \right\} \quad (10.1)$$



## 10.2 거리와 유사도 (Distance vs Similarity)

군집화가 추구하는 본질적인 목표는 같은 군집 내의 샘플은 서로 가깝고 다른 군집에 속한 샘플 사이의 거리는 멀게 하는 것이다. 따라서 군집화에서 거리 개념은 매우 중요하고 여러 가지 계산 방법이 개발되어 있다. 어떤 계산 방법을 사용하느냐에 따라 군집화 결과는 달라지고 상황에 적합할 수도 있고 그렇지 않을 수도 있다. 따라서 주어진 문제에 적합한 거리 측정 방법을 선택하는 것이 매우 중요하다. 거리와 distance 유사도는 similarity 반대 개념이고 하나를 알면 다른 것은 공식을 이용하여 쉽게 계산할 수 있다.



## 10.2.1 특징 값의 종류

- 특징 값의 종류는 다양하다. (군집화를 공부하는데 이에 대한 이해가 필요하다.)

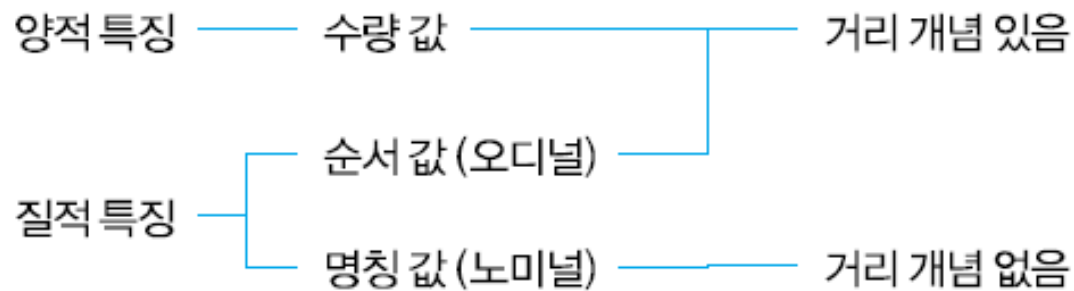


그림 10.3 특징의 유형

양적특징의 예) 무게, 높이, 사람의 수

질적특징은 적절한 방법을 사용하여 양적특징으로 변환 후 사용해야함.

전문가는 12개의 특징 중  
군집화에 필요한 특징을  
선택하는 능력이 필요함

고객 레코드 (샘플)을 12개의 특징으로 표현한다고 하자.

$\mathbf{x} = (\text{나이}, \text{직업}, \text{연봉}, \text{성별}, \text{월평균 구매액}, \text{반품 성향}, \text{선호하는 물품 수준},$   
 $\text{의류 선호도}, \text{전자제품 선호도}, \text{식품 선호도}, \text{팬시 선호도}, \text{DVD 선호도})^T$

나이: [1,100]의 정수

직업: [1,10]의 정수 (1 = 회사원, 2 = CEO, 3 = 교사, ...)

연봉: [1,5]의 정수 (1 = 2천만원 미만, 2 = 2천~3천만원, ..., 5 = 1억 이상)

성별: [0,1]의 정수 (0 = 여자, 1 = 남자)

월평균 구매액: 평균을 계산하여 실수로 표현

반품 성향: [1,4]의 정수 (1 = 년간 반품 횟수 0, 2 = 2회 미만, ...)

선호하는 물품 수준: [1,4]의 정수 (1 = 저가, 2 = 보통, 3 = 고가, 4 = 명품)

제품 선호도: [1,5]의 정수 (1 = 구매한 적 없음, ..., 5 = 아주 선호)

수량 값을 갖는 특징: 나이, 연봉, 월평균 구매액

순서 값을 갖는 특징: 반품 성향, 선호하는 물품 수준, 제품 선호도

명칭 값을 갖는 특징: 직업, 성별



## 10.2.2 거리와 유사도 측정

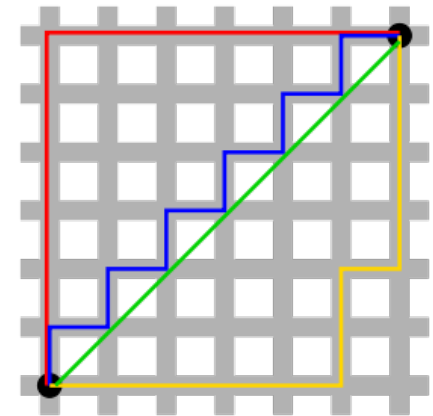
### ■ Minkowski 거리 (10.4)

- 두 점  $\mathbf{x}_i=(x_{i1},\dots,x_{id})^T$ 와  $\mathbf{x}_j=(x_{j1},\dots,x_{jd})^T$ 간의 거리 척도
- $p=2$ 면 **유클리디언 거리 (10.5)**,  $p=1$ 이면 도시블록 거리(**맨하탄거리**) (10.6)

$$d_{ij} = \left( \sum_{k=1}^d |x_{ik} - x_{jk}|^p \right)^{1/p}$$

유클리디언 거리 ( $p=2$ )  $d_{ij} = \sqrt{\sum_{k=1}^d (x_{ik} - x_{jk})^2}$

맨하탄 거리 ( $p=1$ )  $d_{ij} = \sum_{k=1}^d |x_{ik} - x_{jk}|$



(10.6)

### ■ Hamming 거리 (Bit Operation)

- 이진 특징 벡터에 적용 가능 (서로 다른 비트의 개수)
- 예)  $(1,0,1,0,0,0,1,1)^T$ 과  $(1,0,0,1,0,0,1,0)^T$ 의 해밍 거리는 3

## 10.2.2 거리와 유사도 측정

### ■ 코사인 유사도

- 문서 검색 응용에서 주로 사용 (단어가 특징, 출현 빈도가 특징 값)

$$s_{ij} = \cos \theta = \frac{\mathbf{x}_i^T \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|} \quad (10.8)$$

### ■ 이진 특징 벡터의 유사도

$$s_{ij} = \frac{n_{00} + n_{11}}{n_{00} + n_{11} + n_{01} + n_{10}} \quad (10.9)$$

$n_{01} = x_i$ 는 0이고,  $x_j$ 는 1인 특징의 개수

$$s_{ij} = \frac{n_{11}}{n_{11} + n_{01} + n_{10}} \quad (10.10)$$

### ■ 유사도 vs 거리

$$s_{ij} = d_{\max} - d_{ij} \quad (10.7)$$

$$d_{ij} = s_{\max} - s_{ij} \quad (10.11)$$

## 10.2.2 거리와 유사도 측정

**연구 주제와 응용에 따라 거리와 유사도 측정 방법은 다르다.**

따라서 서베이를 통해 기존 연구들의 방법을 파악하는 것이 가장 좋다.

만약, 새롭게 거리와 유사도를 제안해야 하는 상황이라면,  
문제에 맞게 논리적으로 **formulation** 하면 된다.

### 10.2.3 점-집합을 위한 거리

- 여기에서는 점과 점 집합 또는 두 점 집합 간의 거리 (10.2.2 절은 두 점간의 거리 정의)
  - 점  $\mathbf{x}_i$ 와 군집 (점 집합)  $c_j$  간의 거리를  $D(\mathbf{x}_i, c_j)$ 로 표기
  - 두 군집  $c_i$ 와  $c_j$  간의 거리는  $D(c_i, c_j)$ 로 표기

- 점과 군집 사이의 거리 (모든 샘플이 참여)

$$D_{\max}(\mathbf{x}_i, c_j) = \max_{\mathbf{y}_k \in c_j} d_{ik} \quad (10.12)$$

$$D_{\min}(\mathbf{x}_i, c_j) = \min_{\mathbf{y}_k \in c_j} d_{ik} \quad (10.13)$$

$$D_{\text{ave}}(\mathbf{x}_i, c_j) = \frac{1}{|c_j|} \sum_{\mathbf{y}_k \in c_j} d_{ik} \quad (10.14)$$

- $d_{ik}$ 는  $\mathbf{x}_i$ 와  $\mathbf{y}_k$ 는 간의 거리 ( $\mathbf{y}_k$ 는  $c_j$ 에 속한 샘플)
- $D_{\max}$ 와  $D_{\min}$ 은 외톨이에 outlier 민감

## 예제 10.2 점과 군집 사이의 거리

$$c_j = \{y_1 = (1,1)^T, y_2 = (1,2)^T, y_3 = (2,1)^T, y_4 = (3,1)^T\}, \quad x_i = (4,2)^T$$

$$D_{\max} = \max(3.162, 3.0, 2.236, 1.414) = 3.162$$

$$D_{\min} = \min(3.162, 3.0, 2.236, 1.414) = 1.414$$

$$D_{\text{ave}} = (3.162 + 3.0 + 2.236 + 1.414) / 4 = 2.453$$

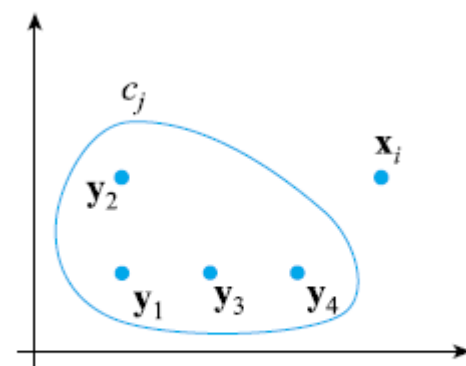


그림 10.4 군집과 점 간의 거리

$$\begin{aligned} D_{\max}(\mathbf{x}_i, c_j) &= \max_{y_k \in c_j} d_{ik} \\ D_{\min}(\mathbf{x}_i, c_j) &= \min_{y_k \in c_j} d_{ik} \\ D_{\text{ave}}(\mathbf{x}_i, c_j) &= \frac{1}{|c_j|} \sum_{y_k \in c_j} d_{ik} \end{aligned}$$

## 10.2.3 점 집합을 위한 거리

### ■ 두 군집 사이의 거리

- $d_{kl}$ 는  $\mathbf{x}_k$ 와  $\mathbf{y}_l$  간의 거리 ( $\mathbf{x}_k$ 는  $c_i$ ,  $\mathbf{y}_l$ 은  $c_j$ 에 속한 샘플)

$$D_{\max}(c_i, c_j) = \max_{\mathbf{x}_k \in c_i, \mathbf{y}_l \in c_j} d_{kl} \quad (10.17)$$

$$D_{\min}(c_i, c_j) = \min_{\mathbf{x}_k \in c_i, \mathbf{y}_l \in c_j} d_{kl} \quad (10.18)$$

$$D_{\text{ave}}(c_i, c_j) = \frac{1}{|c_i||c_j|} \sum_{\mathbf{x}_k \in c_i} \sum_{\mathbf{y}_l \in c_j} d_{kl} \quad (10.19)$$

## 10.3 군집화 알고리즘의 분류

### ■ 매우 다양한 알고리즘

- 군집화 문제의 본질적인 성질에 기인 (주관이 많이 개입되는 성질)

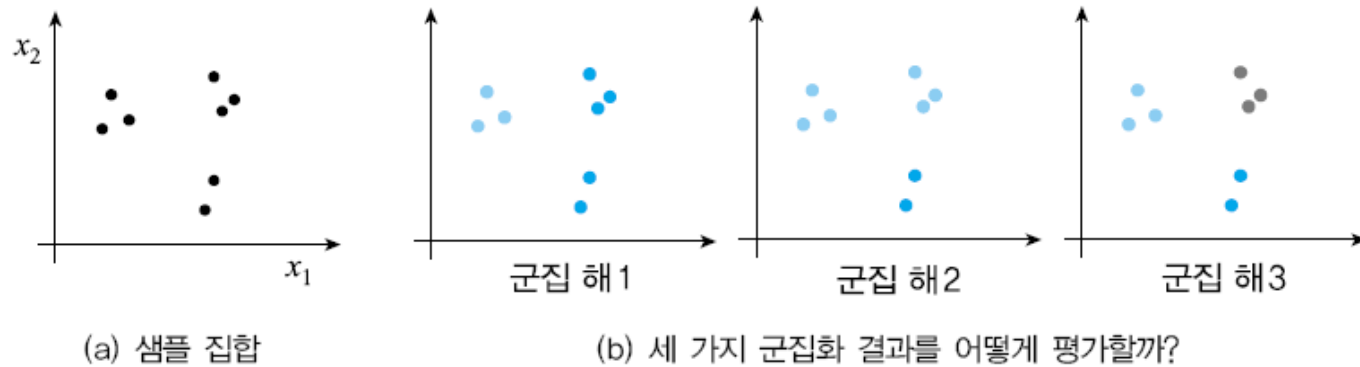


그림 10.1 군집화의 주관성

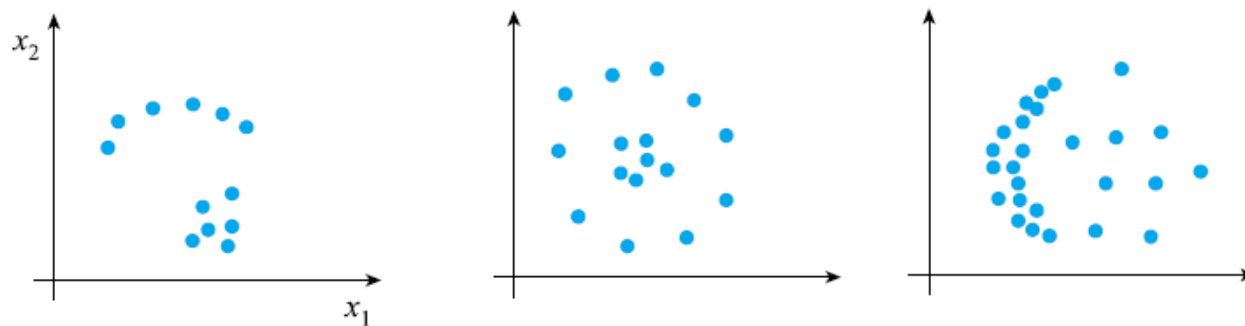
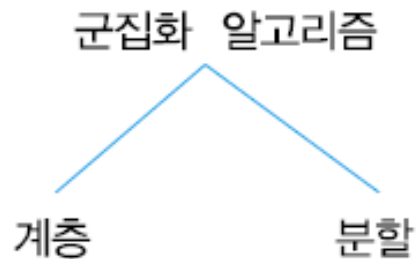


그림 10.5 다양한 군집 상황

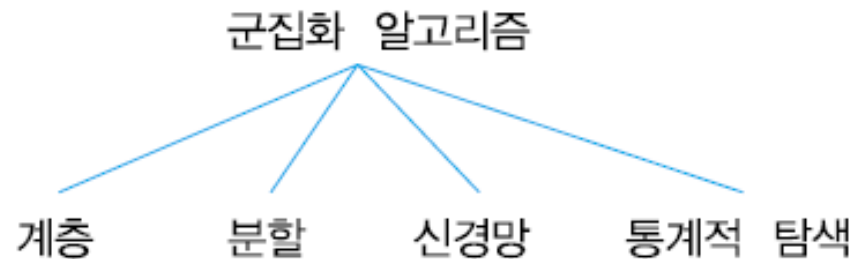
## 10.3 군집화 알고리즘의 분류

### ■ 분류 체계

- 계층 군집화: 군집 결과를 계층을 나타내는 덴드로그램으로 표현
- 분할 군집화: 각 샘플을 군집에 배정하는 연산 사용
- 신경망: SOM, ART
- 통계적 탐색: 시뮬레이티드 어닐링, 유전 알고리즘 등



(a) Jain의 분류 체계

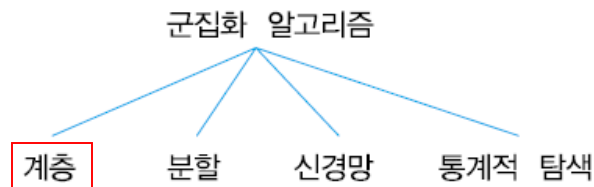


(b) 이 책의 분류 체계

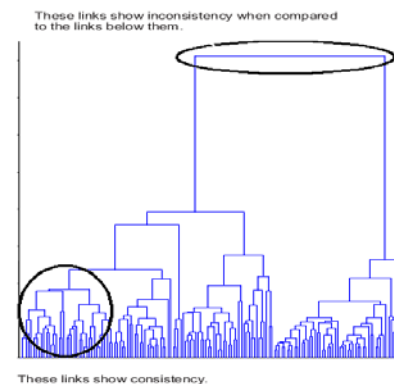
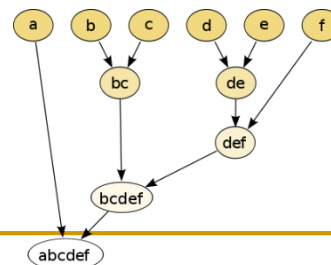
그림 10.6 군집화 알고리즘의 분류



## 10.4 계층 군집화



우선 군집 해의 포함 관계를 정의하자. 군집 해  $C_2 = \{c_{21}, c_{22}, \dots, c_{2n}\}$ 의 모든 군집  $c_{2i}$ 가 다른 군집 해  $C_1 = \{c_{11}, c_{12}, \dots, c_{1k}\}$ 에 속한 군집의 부분 집합일 때  $C_2$ 는  $C_1$ 에 포함된다고 말한다. 예를 들어  $C_2 = \{\{X_1, X_3, X_6\}, \{X_2\}, \{X_4, X_5\}\}$ 는  $C_1 = \{\{X_1, X_3, X_6\}, \{X_2, X_4, X_5\}\}$ 에 포함된다. 물론  $n > k$ 이어야 한다. 계층 군집화 알고리즘은 이러한 포함 관계를 군집화 결과로 출력한다. 계층 군집화 hierarchical clustering 알고리즘에는 작은 군집들에서 출발하여 이들을 모아 나가는 응집 agglomerative 방식과 큰 군집에서 출발하여 이들을 나누어 나가는 분열 divisive 방식이 있다.



## 10.4.1 응집 계층 알고리즘

- 샘플 각각이 군집이 되어 출발, 유사한 군집을 모으는 작업을 반복
  - 출력은 군집화 결과를 트리로 표현한 덴드로그램

### 알고리즘 [10.1]    응집 계층 군집화

입력: 샘플 집합  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$

출력: 덴드로그램

알고리즘:

$O(N^3)$

1.  $C_0 = \{c_1 = \{\mathbf{x}_1\}, c_2 = \{\mathbf{x}_2\}, \dots, c_N = \{\mathbf{x}_N\}\};$  // 각 샘플이 하나의 군집
2. **for** ( $t = 1$  **to**  $N-1$ ) {  $O(N)$
3.  $C_{t-1}$ 의 모든 군집 쌍  $(c_i, c_j)$ 를 조사하여 아래를 만족하는 쌍  $(c_p, c_q)$ 를 찾아라.  
 $O(N^2)$   $D(c_p, c_q) = \min_{c_i, c_j \in C_{t-1}} D(c_i, c_j)$  // 가장 가까운 쌍을 찾는 조건
4.  $c_r = c_p \cup c_q;$  // 두 군집을 하나로 합쳐라.
5.  $C_t = (C_{t-1} - c_p - c_q) \cup c_r;$  // 두 군집을 제거하고 새로운 군집을 추가하라.
6. }

### 예제 10.3 응집 계층 알고리즘 (단일 연결 알고리즘)

일곱 개의 샘플이 주어진 상황

$$\mathbf{x}_1 = (18, 5)^T, \mathbf{x}_2 = (20, 9)^T, \mathbf{x}_3 = (20, 14)^T, \mathbf{x}_4 = (20, 17)^T, \mathbf{x}_5 = (5, 15)^T, \mathbf{x}_6 = (9, 15)^T, \\ \mathbf{x}_7 = (6, 20)^T$$

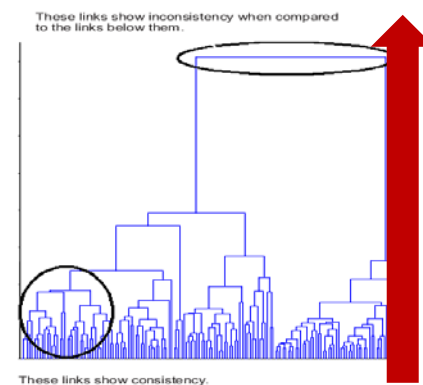
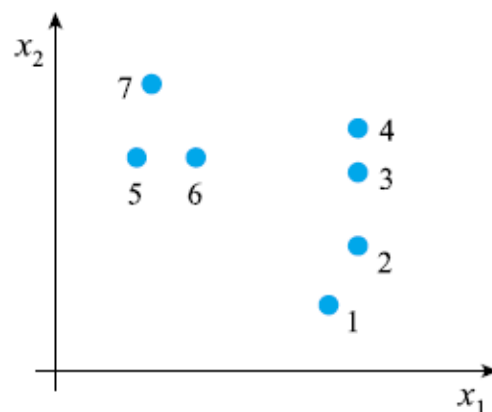


그림 10.7 군집화 예제

(라인 1의) 초기화에 의해,

$$C_0 = \{c_1 = \{\mathbf{x}_1\}, c_2 = \{\mathbf{x}_2\}, c_3 = \{\mathbf{x}_3\}, c_4 = \{\mathbf{x}_4\}, c_5 = \{\mathbf{x}_5\}, c_6 = \{\mathbf{x}_6\}, c_7 = \{\mathbf{x}_7\}\}$$

### 예제 10.3 응집 계층 알고리즘 (단일 연결 알고리즘)

루프를 반복하면, (거리 척도는 유클리언 거리와  $D_{\min}$ 을 가정)

$$C_1 = \{c_1 = \{X_1\}, c_2 = \{X_2\}, c_3 = \{X_3, X_4\}, c_4 = \{X_5\}, c_5 = \{X_6\}, c_6 = \{X_7\}\}$$

$$C_2 = \{c_1 = \{X_1\}, c_2 = \{X_2\}, c_3 = \{X_3, X_4\}, c_4 = \{X_5, X_6\}, c_5 = \{X_7\}\}$$

$$C_3 = \{c_1 = \{X_1, X_2\}, c_2 = \{X_3, X_4\}, c_3 = \{X_5, X_6\}, c_4 = \{X_7\}\}$$

$$C_4 = \{c_1 = \{X_1, X_2, X_3, X_4\}, c_2 = \{X_5, X_6\}, c_3 = \{X_7\}\}$$

$$C_5 = \{c_1 = \{X_1, X_2, X_3, X_4\}, c_2 = \{X_5, X_6, X_7\}\}$$

$$C_6 = \{c_1 = \{X_1, X_2, X_3, X_4, X_5, X_6, X_7\}\}$$

연필을 들고 직접 계산해 보자.  
직접 해보는 것의 힘은 생각보다 크다.

$D_{\min}$  대신  $D_{\max}$ 를 사용하면 어떻게 될까?

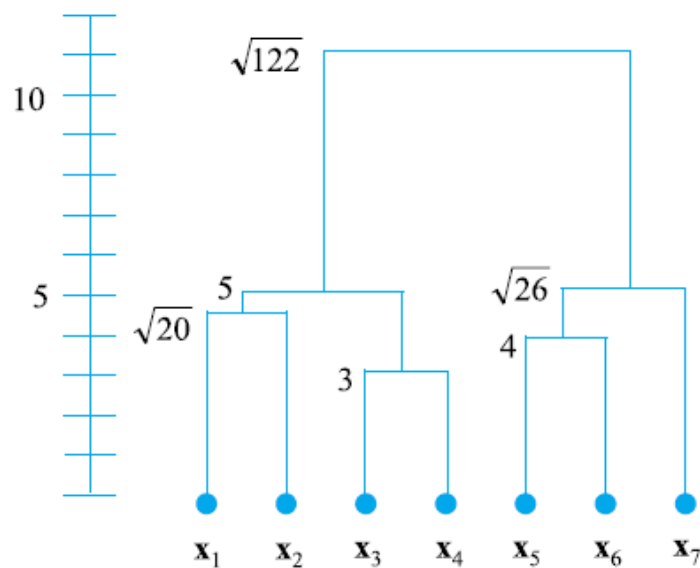


그림 10.8 단일 연결 알고리즘이 만드는 덴드로그램

## 10.4.1 응집 계층 알고리즘

- 사용하는 거리 척도에 따라 다른 이름의 알고리즘
  - $D_{\min}$  사용하면 단일 연결 single-linkage,  $D_{\max}$  사용하면 완전 연결 complete-linkage,  $D_{\text{ave}}$  사용하면 평균 연결 average-linkage 알고리즘
- 세 알고리즘의 동작 특성
  - 단일 연결은 긴 군집, 완전 연결은 둥근 군집을 선호 (평균 연결은 중간)
  - 예) 그림 10.10

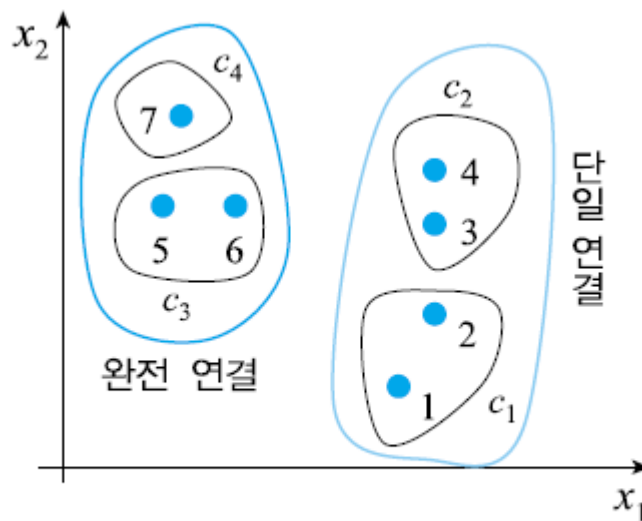


그림 10.10 단일 연결과 완전 연결 알고리즘의 동작 차이

## 10.4.1 응집 계층 알고리즘

### ■ 세가지 부연 설명

#### □ 군집의 개수를 어떻게 알아낼까?

- 모든 군집화 알고리즘이 안고 있는 문제임
- 사용자가 지정 또는 자동 결정 (자동 결정은 어렵다.)

#### □ 단일 연결과 완전 연결은 외톨이에 **outlier** 민감

- 평균 연결은 외톨이에 덜 민감

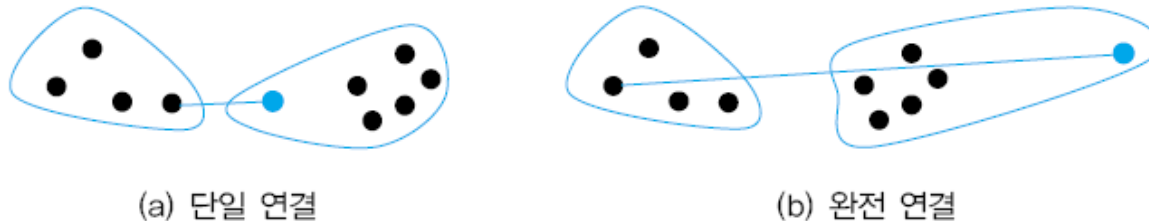


그림 10.11 외톨이에 의한 왜곡된 거리 계산

□ 계산 복잡도 
$$\sum_{t=1}^{N-1} {}_{N-t+1}C_2 = \sum_{t=1}^{N-1} \frac{(N-t+1)(N-t)}{2} = \Theta(N^3) \quad (10.22)$$

- 세제곱에 비례하므로 효율적인 구현 필요
- CURE, ROCK, CHAMELEON, BIRCH 등 [Xu05]

## 10.4.2 분열 계층 알고리즘

### ■ 분열 계층은 하향 방식 top-down

- 모든 샘플이 하나의 군집으로 출발, 군집을 나누는 작업을 반복  
(앞 절의 응집 계층은 상향 방식 bottom-up: 샘플 각각이 군집이 되어 출발, 유사한 군집을 모으는 작업을 반복)

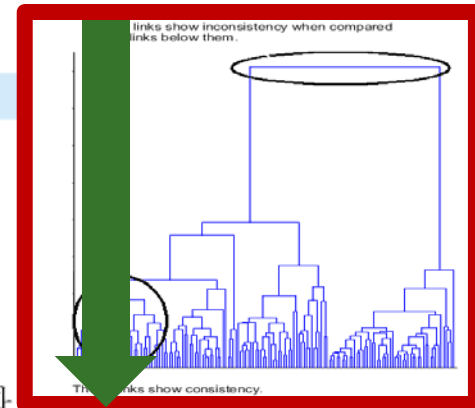
#### 알고리즘 [10.2] 분열 계층 알고리즘

입력: 샘플 집합  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$

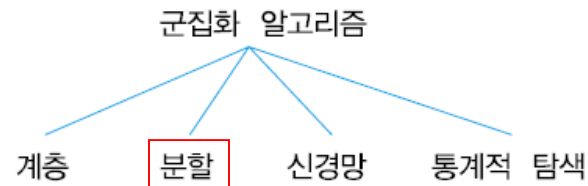
출력: 덴드로그램

알고리즘:

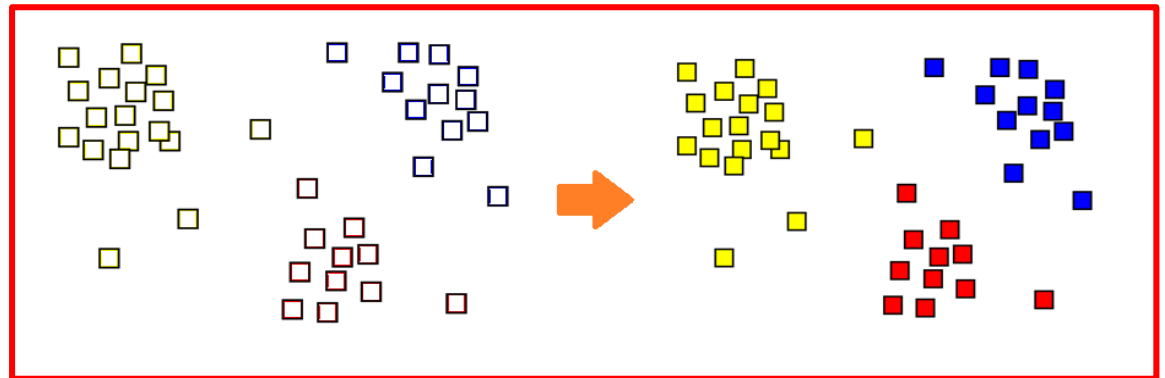
1.  $C_0 = \{c_1 = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}\}$ ; // 모든 샘플이 하나의 군집이 되도록 초기화
2. **for** ( $t=1$  **to**  $N-1$ ) {
3.   **for** ( $C_{t-1}$ 의 모든 군집  $c_i$ 에 대하여)
4.      $c_i$ 의 모든 이진 분할 중에 거리가 가장 먼 것을 찾아라.
5.   라인 3-4에서 찾은  $t$  개의 이진 분할을 비교하여, 가장 먼 거리를 가진 군집  $c_q$ 를 찾고 그것의 이진 분할을  $c_q^1$ 과  $c_q^2$ 라 한다.
6.    $C_t = (C_{t-1} - c_q) \cup \{c_q^1, c_q^2\}$ ; //  $c_q$ 를 제거하고 두 개의 새로운 군집을 추가
7. }



## 10.5 분할 군집화partitional clustering



- 다양한 알고리즘
  - 순차 알고리즘
  - k-means 알고리즘
  - MST알고리즘
  - GMM 알고리즘
  - ...



계층이 존재하지 않고, 평면적인 방법론



## 10.5.1 순차 알고리즘

### ■ 특성

- 군집 개수를 찾아준다. (대신 임계값을 설정해주어야 한다.)
- 순서에 민감하다. 빠르다 (선형 시간).

알고리즘 [10.3] 순차 알고리즘

입력: 샘플 집합  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ , 임계값  $T$

출력: 군집 해  $C$ , 군집의 개수  $k$

알고리즘:

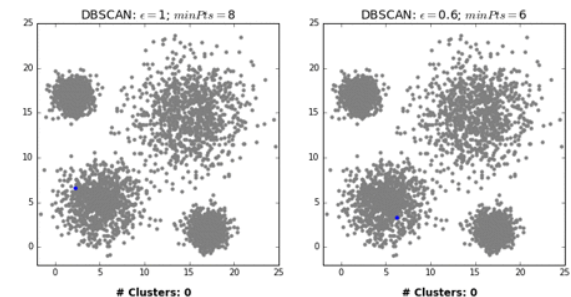
1.  $m = 1$ ; // 현재 군집 개수를 뜻한다.
2.  $C = \{c_1 = \{\mathbf{x}_1\}\}$ ; // 한 개의 군집을 가지고 출발한다.
3. **for** ( $i = 2$  **to**  $N$ ) {
4. 아래 식을 이용하여  $\mathbf{x}_i$ 와 가장 가까운 군집  $c_q$ 를 찾는다.

$$D(\mathbf{x}_i, c_q) = \min_{1 \leq j \leq m} D(\mathbf{x}_i, c_j)$$

5. **if** ( $D(\mathbf{x}_i, c_q) > T$ ) {  $m++$ ;  $c_m = \{\mathbf{x}_i\}$ ;  $C = C \cup c_m$ ; } //  $c_q$ 로부터 멀기 때문에 새로운 군집을 만든다.

6. **else**  $c_q = c_q \cup \mathbf{x}_i$ ; // 가까우니까 그것에 넣는다.
7. }
8.  $k = m$ ;

### DBSCAN



## 10.5.2 $k$ -means 알고리즘

### ■ 특성

- 가장 널리 쓰인다. (직관적 이해, 구현 간편)
- **군집 개수를 설정해주어야 한다.**

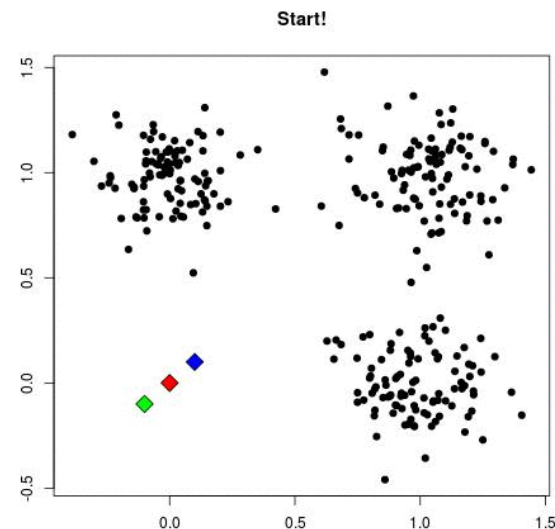
#### 알고리즘 [10.4] $k$ -means 알고리즘

입력: 샘플 집합  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ , 군집의 개수  $k$

출력: 군집 해  $C$

알고리즘:

1.  $k$  개의 군집 중심  $Z = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k\}$ 를 초기화 한다.
2. **while** (TRUE) {
3.     **for** ( $i = 1$  to  $N$ )  $\mathbf{x}_i$ 를 가장 가까운 **군집 중심에 배정한다.**
4.     **if** (이 배정이 이전 루프의 배정과 같음) **break**;
5.     **for** ( $j = 1$  to  $k$ )  $\mathbf{z}_j$ 에 배정된 샘플의 **평균으로  $\mathbf{z}_j$ 를 대체한다.**
6. }



## 예제 10.5 $k$ -means 알고리즘

7개 샘플을  $k=3$ 개의 군집으로 만드는 상황

$$\mathbf{x}_1 = (18, 5)^T, \mathbf{x}_2 = (20, 9)^T, \mathbf{x}_3 = (20, 14)^T, \mathbf{x}_4 = (20, 17)^T, \mathbf{x}_5 = (5, 15)^T, \mathbf{x}_6 = (9, 15)^T, \\ \mathbf{x}_7 = (6, 20)^T$$

초기화에 의해  $\{\mathbf{x}_1\}$ 은  $\mathbf{z}_1$   
(그림 10.12(a)),  $\{\mathbf{x}_2\}$ 은  $\mathbf{z}_2$

라인 5에 의해  $\mathbf{z}_1 = \mathbf{x}_1 = (18, 5)^T$   
(그림 10.12(b)),  $\mathbf{z}_2 = \mathbf{x}_2 = (20, 9)^T$

$\{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\}$ 은  $\mathbf{z}_3$

$$\mathbf{z}_3 = (\mathbf{x}_3 + \mathbf{x}_4 + \mathbf{x}_5 + \mathbf{x}_6 + \mathbf{x}_7) / 5 = (12, 16.2)^T$$

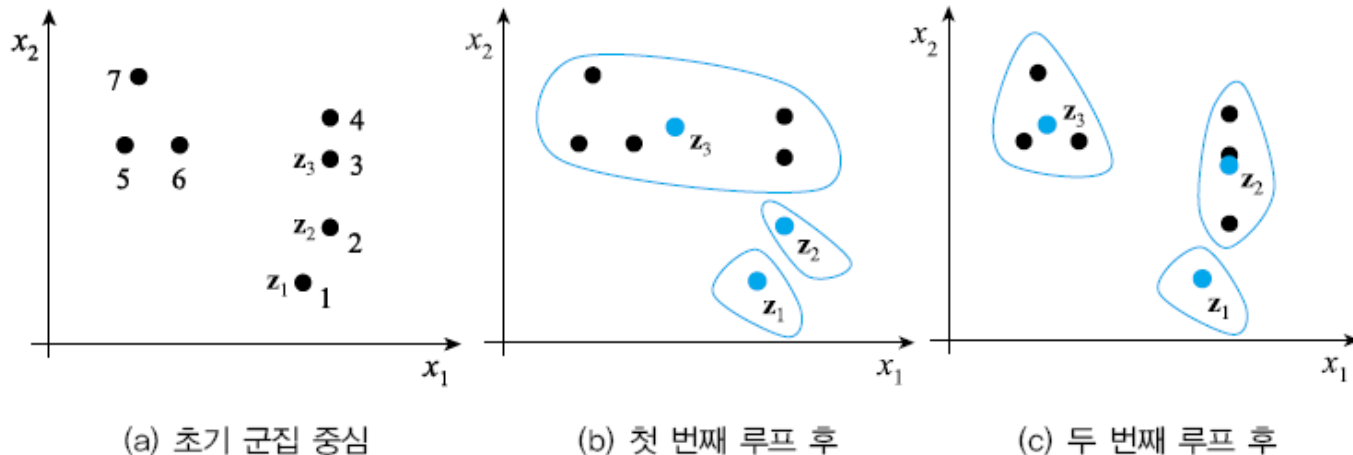


그림 10.12  $k$ -means의 동작 예

## 예제 10.5 $k$ -means 알고리즘

두 번째 루프를 실행하면  $\{\mathbf{x}_1\}$ 은  $\mathbf{z}_1$   
(그림 10.12(c)),  $\{\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$ 은  $\mathbf{z}_2$   
 $\{\mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\}$ 은  $\mathbf{z}_3$

$$\mathbf{z}_1 = \mathbf{x}_1 = (18, 5)^T$$

$$\mathbf{z}_2 = (\mathbf{x}_2 + \mathbf{x}_3 + \mathbf{x}_4)/3 = (20, 13.333)^T$$

$$\mathbf{z}_3 = (\mathbf{x}_5 + \mathbf{x}_6 + \mathbf{x}_7)/3 = (6.667, 16.667)^T$$

세 번째 루프는 그 이전과 결과가 같다. 따라서 멈춘다.  
결국 출력은

$$C = \{\{\mathbf{x}_1\}, \{\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\}\}$$

## 10.5.2 $k$ -means 알고리즘

### ■ 이론적 배경 (알고리즘의 우수성)

- (10.23)을 비용 함수로 하는 내리막 경사법의 일종

$$J(\mathbf{Z}, \mathbf{U}) = \sum_{i=1}^N \sum_{j=1}^k u_{ji} \|\mathbf{x}_i - \mathbf{z}_j\|^2 \quad (10.23)$$

- 항상 지역 최적점으로 수렴한다. (전역 최적점 보장 못함)
- 초기 군집 중심에 민감
- 빠르다.
- 외톨이에 민감하다. ( $k$ -medoids는 덜 민감)



그림 10.13  $k$ -means와  $k$ -medoids의 군집 중심을 계산하는 과정의 비교  
(진파랑은  $k$ -medoids, 연파랑은  $k$ -means)

## 10.5.3 모델 기반 알고리즘

- 샘플로부터 가우시언을 추정하고 그 결과에 따라 군집 배정
  - 가우시언 추정은 EM 알고리즘을 사용할 수 있다.

### 알고리즘 [10.6]

### 가우시언 모델 기반 알고리즘

입력: 샘플 집합  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ , 군집의 개수  $k$

출력: 군집 해  $C$

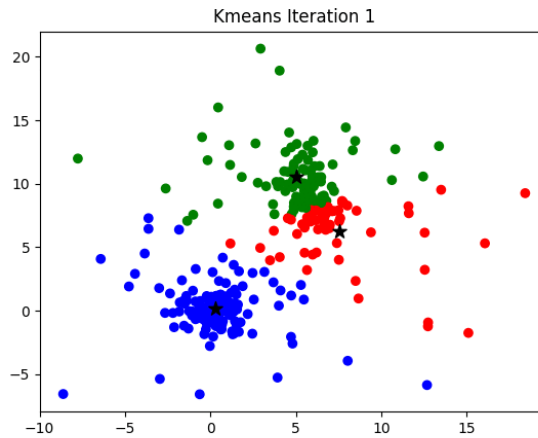
알고리즘:

1.  $X$ 를 가지고 EM 알고리즘을 수행하여 가우시언  $G_j, j = 1, \dots, k$ 를 추정한다.
2. (10.24)의 규칙으로 각각의 샘플을 군집에 배정한다.

Hard assignment 를  
대신한  
Soft assignment 방법

$$\left. \begin{array}{l} \mathbf{x}_i \text{를 } c_q \text{에 배정한다.} \\ \text{이때 } P(G_q | \mathbf{x}_i) > P(G_j | \mathbf{x}_i), j = 1, \dots, k, j \neq q \end{array} \right\} \quad (10.24)$$

## 10.5.3 모델 기반 알고리즘



Hard assignment를 사용하는  
Kmeans 는  
Outlier에 취약하다.

