기계학습 개론

- Unsupervised Learning (비지도 학습)

정보통신공학과

Prof. Jinkyu Kang

MYONGJI UNIVERSITY

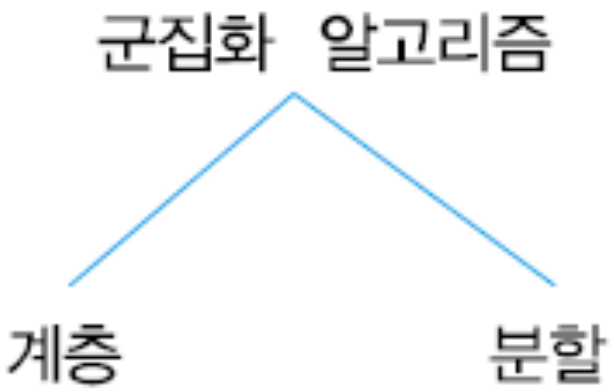
이번주 수업의 목차

- 군집화
  - 분할 군집화
    - 순차 알고리즘
    - k-means 알고리즘
    - 모델 기반 알고리즘
  - 계층 군집화
    - 응집 계층 알고리즘
    - 분열 계층 알고리즘
  - 신경망
    - 자기 조직화 맵

MYONGJI UNIVERSITY

군집화 알고리즘의 분류

- 분류 체계
  - 계층 군집화: 군집 결과를 계층을 나타내는 덴드로그램으로 표현
  - 분할 군집화: 각 샘플을 군집에 배정하는 연산 사용



MYONGJI UNIVERSITY

계층 군집화

우선 군집 해의 포함 관계를 정의하자. 군집 해 $C_2 = \{c_{21}, c_{22}, \cdots, c_{2n}\}$의 모든 군집 c_{2i}가 다른 군집 해 $C_1 = \{c_{11}, c_{12}, \cdots, c_{1k}\}$에 속한 군집의 부분 집합일 때 C_2는 C_1에 포함된다고 말한다. 예를 들어 $C_2 = \{\{\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_6\}, \{\mathbf{x}_2\}, \{\mathbf{x}_4, \mathbf{x}_5\}\}$는 $C_1 = \{\{\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_6\}, \{\mathbf{x}_2, \mathbf{x}_4, \mathbf{x}_5\}\}$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.

MYONGJI UNIVERSITY

계층 군집화 | 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 $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$

출력: 덴드로그램

알고리즘:

1. $C_0 = \{c_1 = \{\mathbf{x}_1\}, c_2 = \{\mathbf{x}_2\}, \cdots, c_N = \{\mathbf{x}_N\}\}$; // 각 샘플이 하나의 군집
2. **for** $(t = 1$ **to** $N-1)$ {
3. C_{t-1}의 모든 군집 쌍 (c_i, c_j)를 조사하여 아래를 만족하는 쌍 (c_p, c_q)를 찾아라.

$$D(c_p, c_q) = \min_{c_i, c_j \in C_{t-1}} D(c_i, c_j)$$ // 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. }

MYONGJI UNIVERSITY

계층 군집화 | 응집 계층 알고리즘

예제 10.3 응집 계층 알고리즘 (단일 연결 알고리즘)

일곱 개의 샘플이 주어진 상황

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

$\mathbf{x}_7 = (6,20)^T$

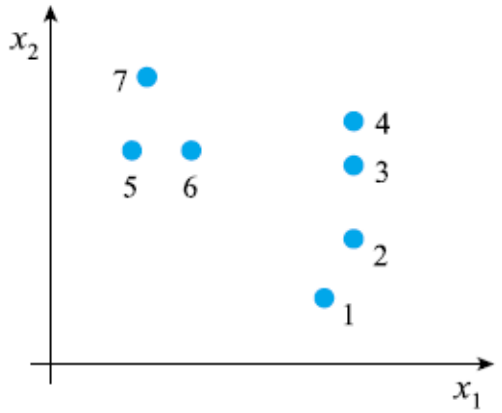


그림 10.7 군집화 예제

(라인 1의) 초기화에 의해,

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

MYONGJI UNIVERSITY

계층 군집화 | 응집 계층 알고리즘

예제 10.3 응집 계층 알고리즘 (단일 연결 알고리즘)

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

$$C_1 = \{c_1 = \{\mathbf{x}_1\},\ c_2 = \{\mathbf{x}_2\},\ c_3 = \{\mathbf{x}_3,\mathbf{x}_4\},\ c_4 = \{\mathbf{x}_5\},\ c_5 = \{\mathbf{x}_6\},\ c_6 = \{\mathbf{x}_7\}\}$$

$$C_2 = \{c_1 = \{\mathbf{x}_1\},\ c_2 = \{\mathbf{x}_2\},\ c_3 = \{\mathbf{x}_3,\mathbf{x}_4\},\ c_4 = \{\mathbf{x}_5,\mathbf{x}_6\},\ c_5 = \{\mathbf{x}_7\}\}$$

$$C_3 = \{c_1 = \{\mathbf{x}_1,\mathbf{x}_2\},\ c_2 = \{\mathbf{x}_3,\mathbf{x}_4\},\ c_3 = \{\mathbf{x}_5,\mathbf{x}_6\},\ c_4 = \{\mathbf{x}_7\}\}$$

$$C_4 = \{c_1 = \{\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\mathbf{x}_4\},\ c_2 = \{\mathbf{x}_5,\mathbf{x}_6\},\ c_3 = \{\mathbf{x}_7\}\}$$

$$C_5 = \{c_1 = \{\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\mathbf{x}_4\},\ c_2 = \{\mathbf{x}_5,\mathbf{x}_6,\mathbf{x}_7\}\}$$

$$C_6 = \{c_1 = \{\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\mathbf{x}_4,\mathbf{x}_5,\mathbf{x}_6,\mathbf{x}_7\}\}$$

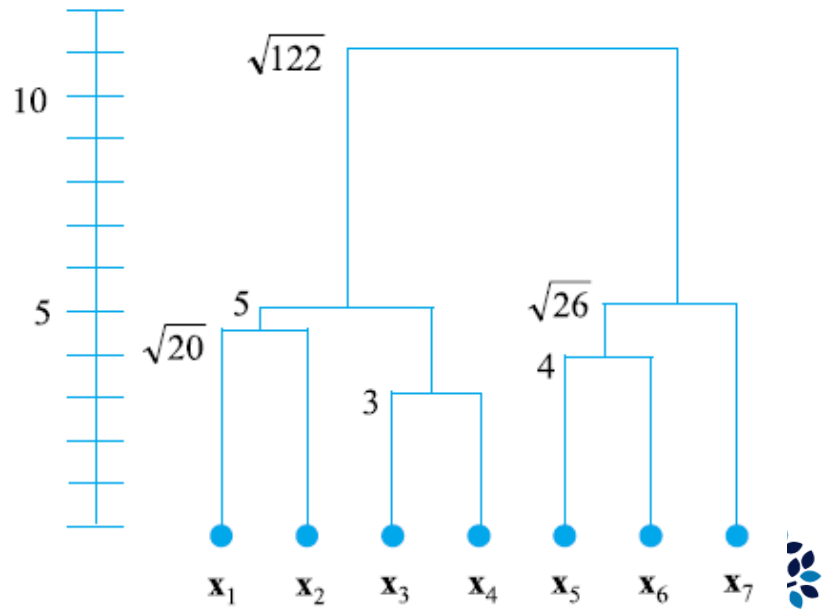


그림 10.8 단일 연결 알고리즘이 만드는 덴드로그램

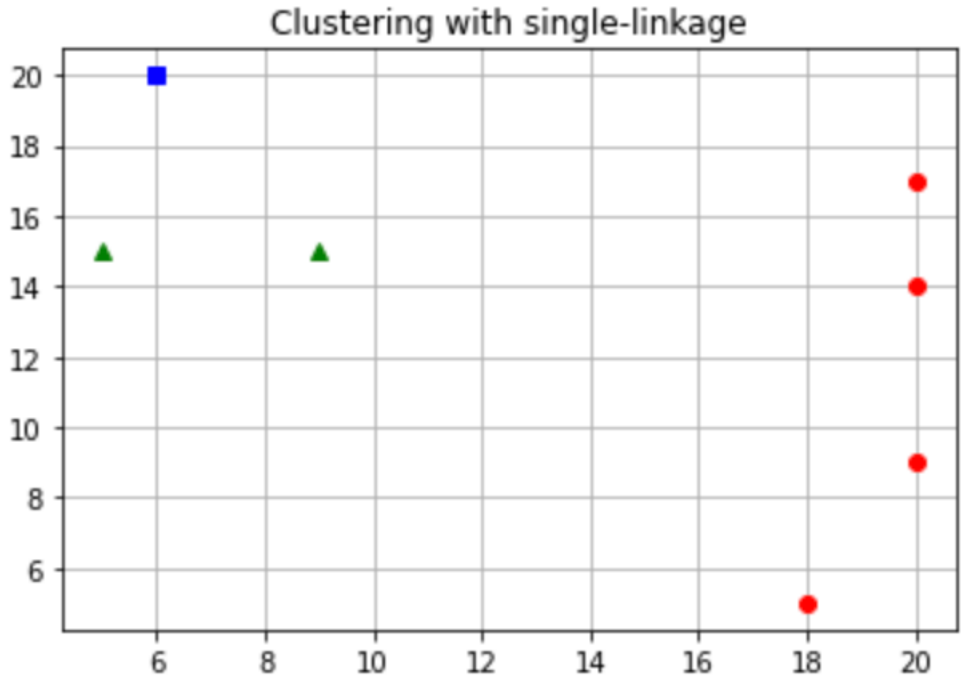
계층 군집화 I 응집 계층 알고리즘 (Python)

예제 10.3 응집 계층 알고리즘 (단일 연결 알고리즘)

일곱 개의 샘플이 주어진 상황

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

$\mathbf{x}_7 = (6,20)^T$



MYONGJI UNIVERSITY

계층 군집화 | 응집 계층 알고리즘

- 사용하는 거리 척도에 따라 다른 이름의 알고리즘
  - $D_{\min}$ 사용하면 단일 연결single-linkage, $D_{\max}$ 사용하면 완전 연결complete-linkage, D_{ave} 사용하면 평균 연결average-linkage 알고리즘
- 세 알고리즘의 동작 특성
  - 단일 연결은 긴 군집, 완전 연결은 둥근 군집을 선호 (평균 연결은 중간)

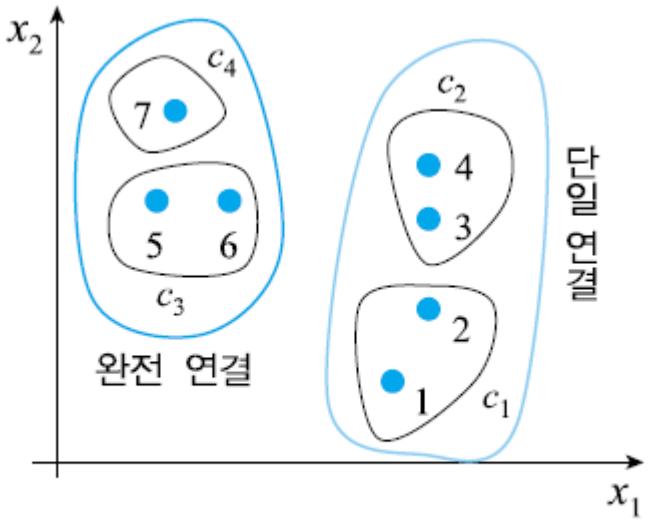


그림 10.10 단일 연결과 완전 연결 알고리즘의 동작 차이

MYONGJI UNIVERSITY

계층 군집화 | 응집 계층 알고리즘

- 부연 설명
  - 군집의 개수를 어떻게 알아낼까?
    - 모든 군집화 알고리즘이 안고 있는 문제임
    - 사용자가 지정 또는 자동 결정 (자동 결정은 어렵다.)
  - 단일 연결과 완전 연결은 외톨이에outlier 민감
    - 평균 연결은 외톨이에 덜 민감

(a) 단일 연결 (b) 완전 연결

그림 10.11 외톨이에 의한 왜곡된 거리 계산

계층 군집화 | 분열 계층 알고리즘

- 분열 계층은 하향 방식top-down
  - 모든 샘플이 하나의 군집으로 출발, 군집을 나누는 작업을 반복
    (앞 절의 응집 계층은 상향 방식bottom-up: 샘플 각각이 군집이 되어 출발, 유사한 군집을 모으는 작업을 반복)

알고리즘 [10.2] 분열 계층 알고리즘

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

출력: 덴드로그램

알고리즘:

1. $C_0 = \{c_1 = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}\}$; // 모든 샘플이 하나의 군집이 되도록 초기화
2. **for** (t=1 **to** $N-1$) {
3. **for** (C_{t-1}의 모든 군집 c_i에 대하여)
4. c_i의 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_q를 찾고 그것의 이진 분할을 c_q^1과 c_q^2라 한다.
6. $C_t = (C_{t-1} - c_q) \cup \{c_q^1, c_q^2\}$; // c_q를 제거하고 두 개의 새로운 군집을 추가
7. }

지수적 복잡도

신경망

- 신경망
  - 지도 학습: 퍼셉트론, 다층 퍼셉트론
  - 비지도 학습 (군집화): SOM (self-organizing map), ART (adaptive resonance theory)

신경망 | 자기 조직화 맵

- 자기 조직화 맵self-organizing map (SOM)
  - 샘플들을 상호 비교하며 스스로 군집을 조직해 냄
  - 경쟁 학습을 사용
    - 하나의 샘플이 입력되면 여러 대표 벡터가 경쟁
    - 가장 가까운 벡터가 승자가 되어 그것을 취함 (승자 독식 전략)
    - 승자는 그 샘플에 적응하는 방향으로 벡터 값을 조금 변화시킴

신경망 | 자기 조직화 맵

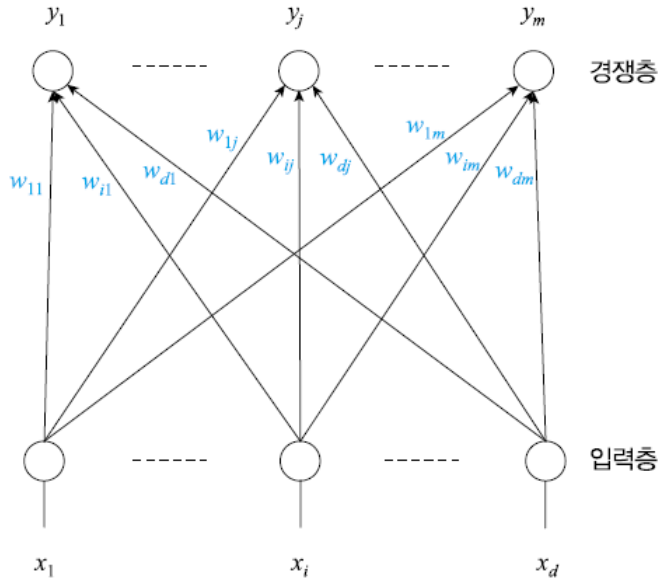


그림 10.14 SOM 아키텍처

SOM의 원리를 이해하기 위해서는 노드 y_j로 들어오는 d 개의 가중치의 역할을 이해하는 것이 매우 중요하다. 이들을 가중치 벡터로 보고 $\mathbf{w}_j = (w_{1j}, w_{2j}, \cdots, w_{dj})^T$로 표기하자. 이 벡터는 d 차원으로서 샘플의 특징 벡터 $\mathbf{x} = (x_1, x_2, \cdots, x_d)^T$와 차원이 같다. 샘플 $\mathbf{x}$가 입력되면 m 개의 가중치 벡터가 서로 경쟁하는데 그들 중 $\mathbf{x}$에 가장 가까운 벡터가 승자가 된다. 이때 승자를 결정하기 위해서는 가중치 벡터와 샘플 벡터 간의 거리를 이용한다. 경쟁의 결과 q 번째 벡터가 승자라 하면 $\mathbf{w}_q$는 $\mathbf{x}$에 적응하기 위해 자신의 값을 변화시킨다. 이 변화가 학습에 해당한다.

신경망 | 자기 조직화 맵

- SOM 학습 규칙
  - (10.25)에 의해 $\mathbf{w}_{new}$는 $\mathbf{w}_{old}$보다 $\mathbf{x}$에 가깝게 된다.

$$\mathbf{w}_{new} = \mathbf{w}_{old} + \rho(\mathbf{x} - \mathbf{w}_{old}) \qquad (10.25)$$

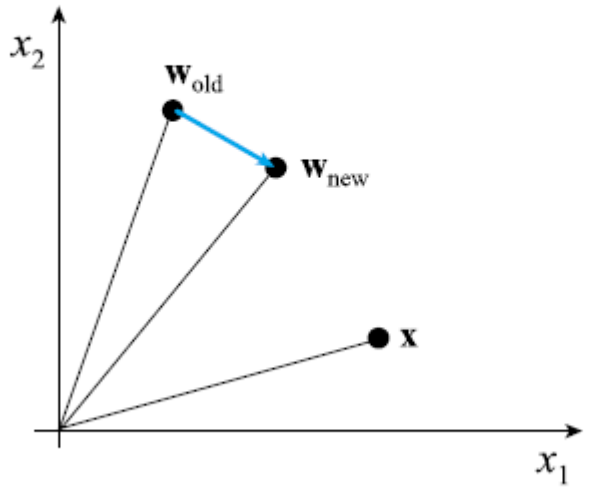


그림 10.16 SOM의 학습을 기하학적으로 해석

신경망 | 자기 조직화 맵

알고리즘 [10.7] SOM 알고리즘

입력: 샘플 집합 $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$, 최대 군집 개수 m, 학습률 ρ, 이웃 반경 r

출력: 군집 해 C

알고리즘:

1. m 개의 가중치 벡터 $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_m$을 초기화 한다.
2. **repeat** {
3. **for** ($n = 1$ **to** N) { // 샘플 각각에 대해
4. $\mathbf{x}_n$에 가장 가까운 가중치 벡터 $\mathbf{w}_q$를 찾는다. // 승자 찾기 (군집 배정)
5. **for** (q의 이웃 노드 각각에 대해) // 학습
6. $\mathbf{w}_{\text{new}} = \mathbf{w}_{\text{old}} + \rho(\mathbf{x}_n - \mathbf{w}_{\text{old}})$;
7. }
8. ρ와 r을 조정한다. // 필요하다면
9. } **until** (stop-condition);
// SOM 학습이 끝났으니 이제는 이것을 이용하여 군집 해를 구한다.
10. **for** ($n = 1$ **to** N)
11. $\mathbf{x}_n$에 가장 가까운 가중치 벡터 $\mathbf{w}_q$를 찾아 $\mathbf{x}_n$을 군집 c_q에 배정한다.
12. 군집 c_j, $j = 1, \cdots, m$ 중에 비지 않을 것을 가지고 군집 해 C를 만든다.

MYONGJI UNIVERSITY

신경망 | 자기 조직화 맵

예제 10.6 SOM을 이용한 군집화

예제 10.3의 그림 10.7을 다시 사용해 보자. 편의상 일곱 개 샘플의 좌표를 다시 적어 주면 아래와 같다.

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

SOM의 매개 변수를 아래와 같이 설정했다 하자.

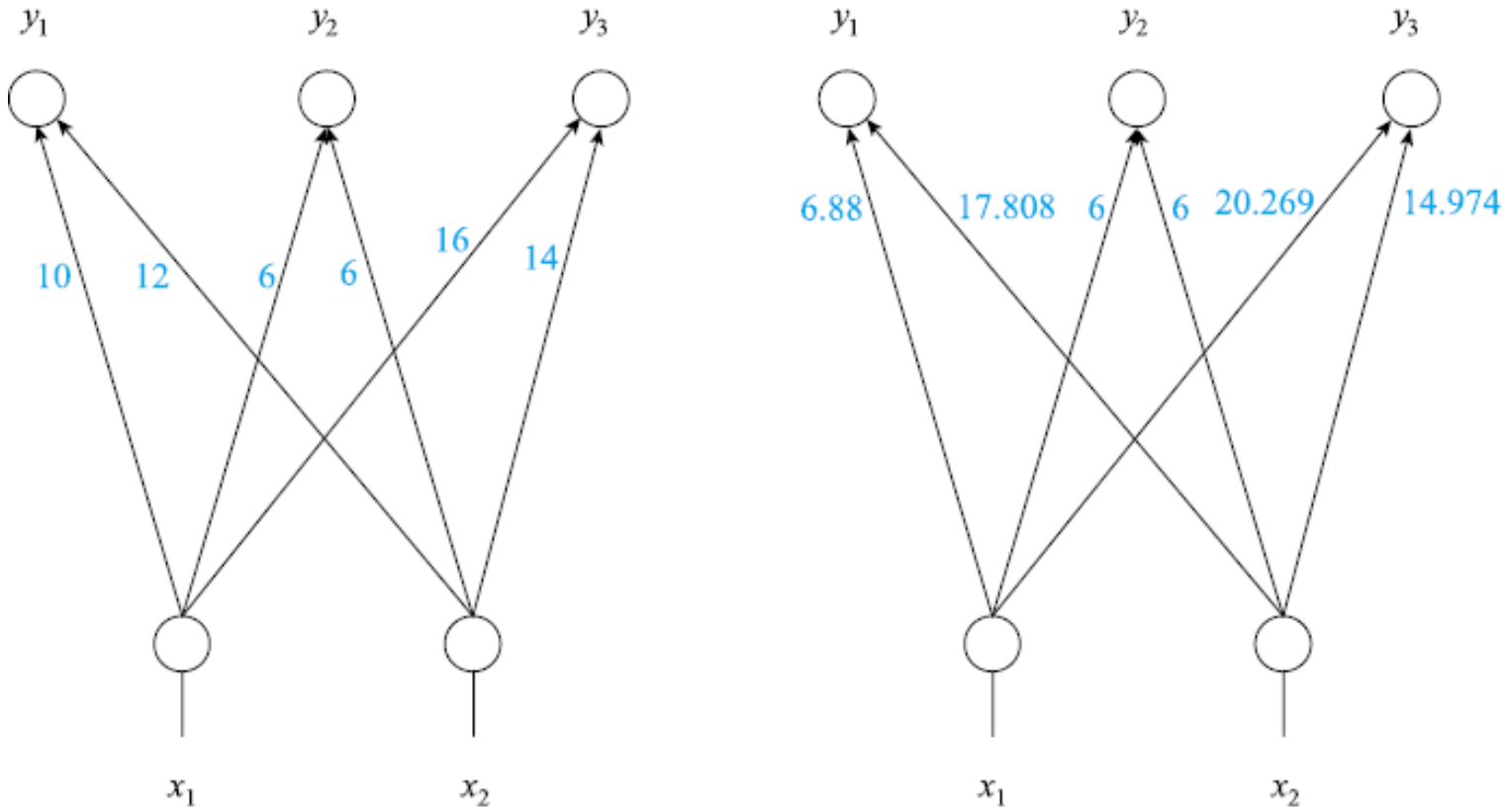
입력 노드 수 $d = 2$ (특징 벡터의 차원 수)

최대 군집 개수 $m = 3$

초기 학습률 $\rho = 0.6$, 루프를 돌 때마다 조정 $\rho_{new} = 0.8 * \rho_{old}$

이웃 반경 $r = 0$

신경망 | 자기 조직화 맵



(a) 초기 SOM

(b) 한 세대 후의 SOM

그림 10.17 예제 10.6의 SOM