추가설명_k-means 알고리즘

❖ k-means의 계산 절차

Lloyd 알고리즘

1. 시작: 데이터 집합 $[\mathbf{x}_1, ..., \mathbf{x}_N]$에서 K개의 벡터를 임의로 선택하여 K개의 초기 중심 집합 $[y_1, ..., y_K]$를 만든다.
2. E단계: 만약 데이터 $\mathbf{x}_n$이 $\mathbf{y}_i$에 가장 가깝다면 클러스터 X_i에 속하도록 라벨링한다. 결국 데이터 집합은 K개의 클러스터들 $\{X_1, ..., X_K\}$로 나누어진다.

$$X_i = \{\mathbf{x}_n \mid d(\mathbf{x}_n, \mathbf{y}_i) \le d(\mathbf{x}_n, \mathbf{y}_j),\ j = 1, ..., K\} \tag{7.5}$$

3. M단계: E단계에서 구한 새로운 클러스터들에서 각각의 중심을 갱신한다.

$$\mathbf{y_i} = c(X_i),\ i = 1, ..., K \tag{7.6}$$

4. 데이터와 가장 가까운 클러스터 중심들과 거리의 합으로 총 왜곡(distortion)을 구한다.

$$D = \sum_{n=1}^{N} d(\mathbf{x_n}, \mathbf{y_{i(n)}}), \tag{7.7}$$

※ $\mathbf{x_n} \in X_k$라면 $i(n) = k$

5. 총 왜곡이 적절하게 변하지 않거나 설정된 반복 횟수에 도달할 때까지 2~4단계를 반복한다.

$$\Delta\mathbf{D} = \frac{\mathbf{D}_{\text{prev}} - \mathbf{D}_{\text{curr}}}{\mathbf{D}_{\text{prev}}} < 10^{-4} \quad : \text{왜곡 검증} \tag{7.8}$$

추가설명_ 비균일 이진 분할

❖ 이진 분할 계산 절차

LBG(Linde-Buzo-Gray) 알고리즘

1. 한 개의 클러스터 X_1과 관련된 중심이 $\mathbf{y}_1 = c(X_1)$인 모든 데이터 점 $\mathbf{x}_n$으로 시작한다. 클러스터 카운트는 $k=1$로 둔다.
2. K개의 중심들을 얻기 위해서 3~6단계를 (K−1)번 반복한다.
3. 클러스터 내의 점들과 중심의 평균 거리로 측정된 가장 큰 왜곡이 있는 클러스터 X_i를 선택한다.

$$D = \frac{1}{N_j}\sum_{n=1}^{N_j} d(\mathbf{x}_n^{(j)}, \mathbf{y}_j),\ X_j = \{\mathbf{x}_n^{(j)} \mid n = 1 \ldots N_j\},\ D_i \geq D_j,\ j = 1, \ldots, k \qquad (7.9)$$

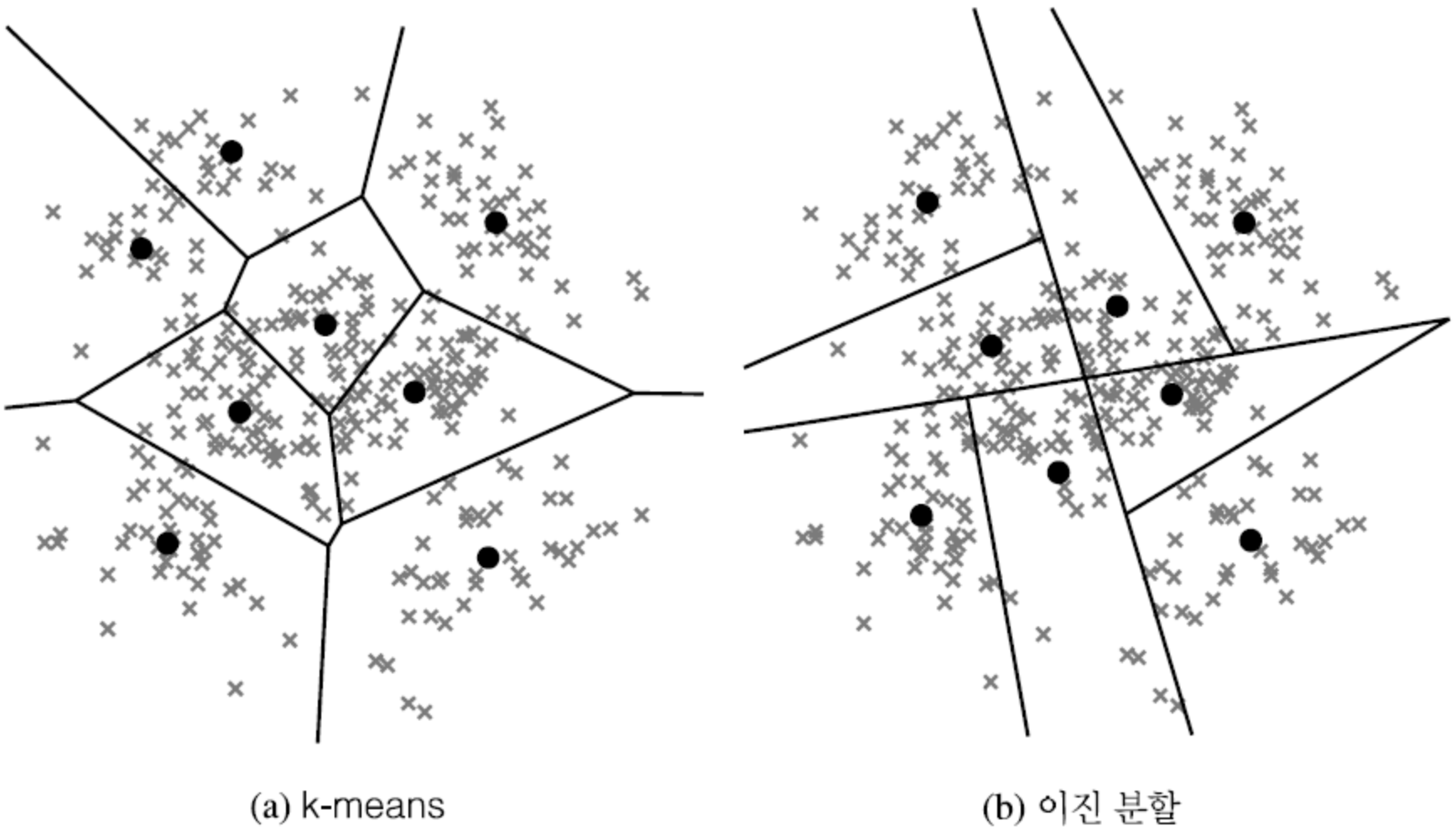
4. 선택된 클러스터 X_i를 다음 방법들 중에서 하나를 선택하여 두 개의 부클러스터 X_a와 X_b로 나눈다.
   (a) K=2로 집합 X_i상에서 k-means를 행한다.
   (b) 집합 X_i의 주고유벡터 v_i를 결정하고 부클러스터 X_i에 있는 점들이 $\mathbf{y}_i + \mathbf{v}_i$에 가장 가까운 점들을 X_a로, $\mathbf{y}_i - \mathbf{v}_i$에 가장 가까운 점들을 X_b로 한다.
5. 중심 $\mathbf{y}_i$를 대치하고 새로운 중심을 다음과 같이 둔다.

$$\mathbf{y}_i = c(X_a)\ \mathbf{y}_{k+1} = c(X_b) \qquad (7.10)$$

6. 클러스터 카운트를 증가시킨다.

$$k \leftarrow k+1 \qquad (7.11)$$

추가설명_ 비균일 이진 분할



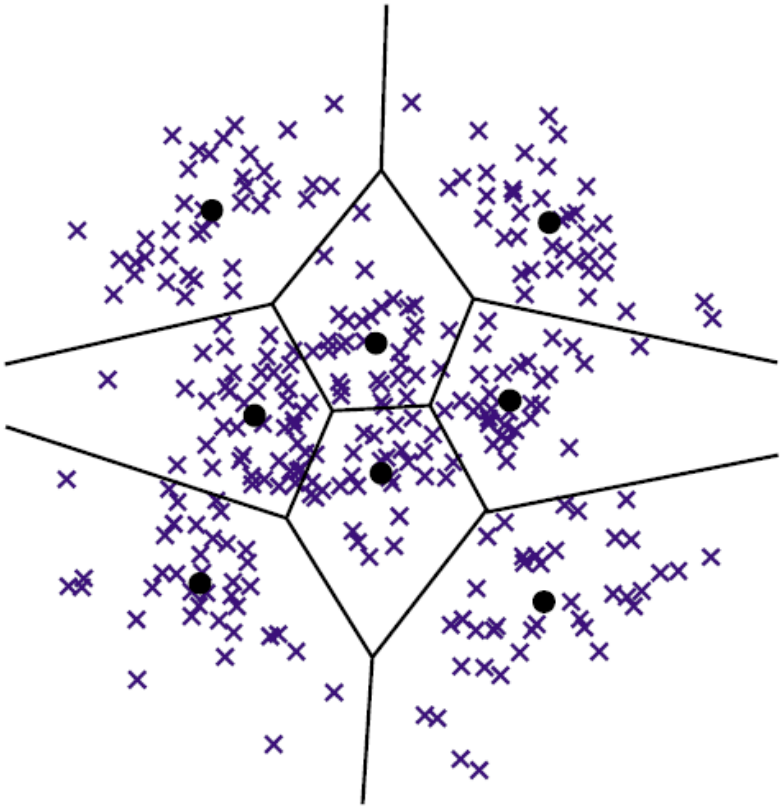
(a) k-means　　(b) 이진 분할

[그림 7-2] k-means와 이진 분할 비교

07_k-means와 이진 분리의 비교와 개선 : LBG 알고리즘

❖ k-means의 초기값

- 랜덤 데이터 점으로 하는 대신에 이진 분리로 구한 중심을 사용하면 표준 k-means 방법을 사용하는 경우보다 성능이 좀 더 나아진다.
- 이진 분리와 k-means를 결합된 알고리즘을 LBG알고리즘이라고 한다.

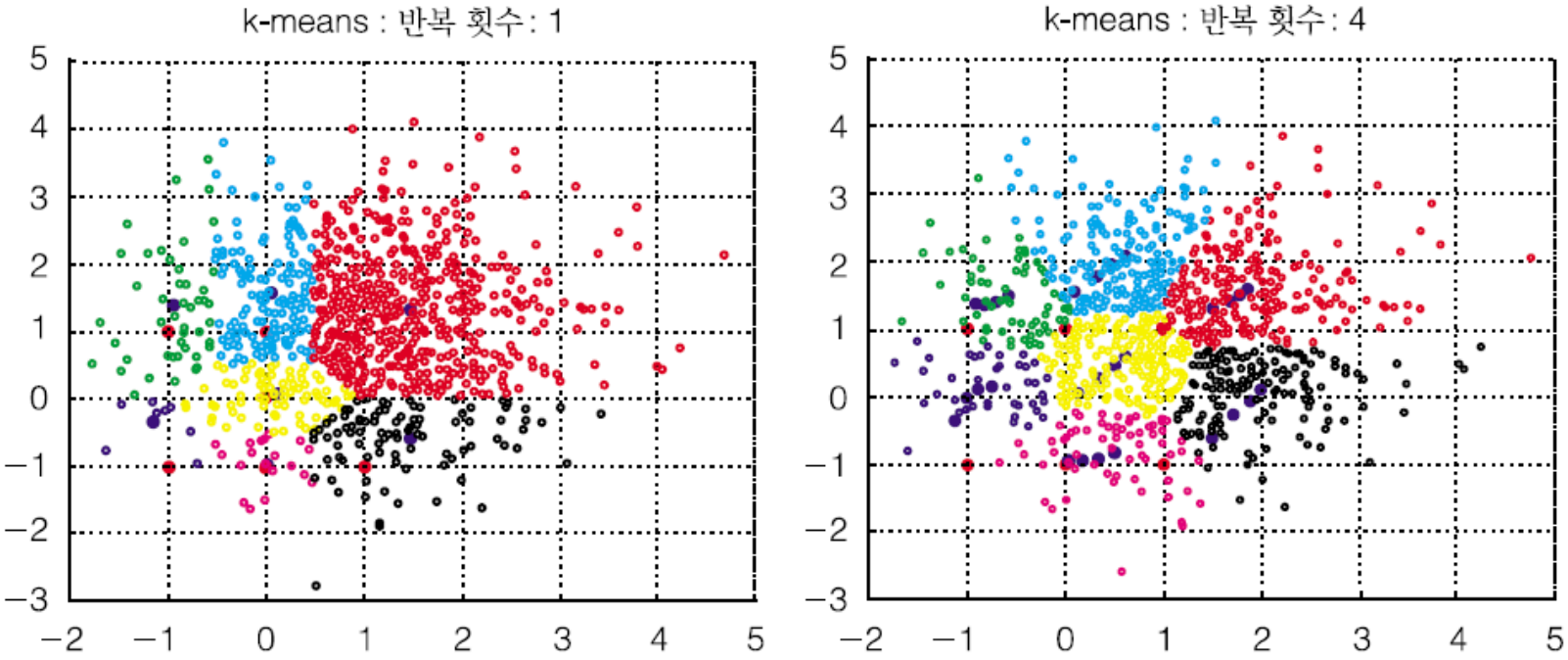


[그림 7-3] LBG 알고리즘

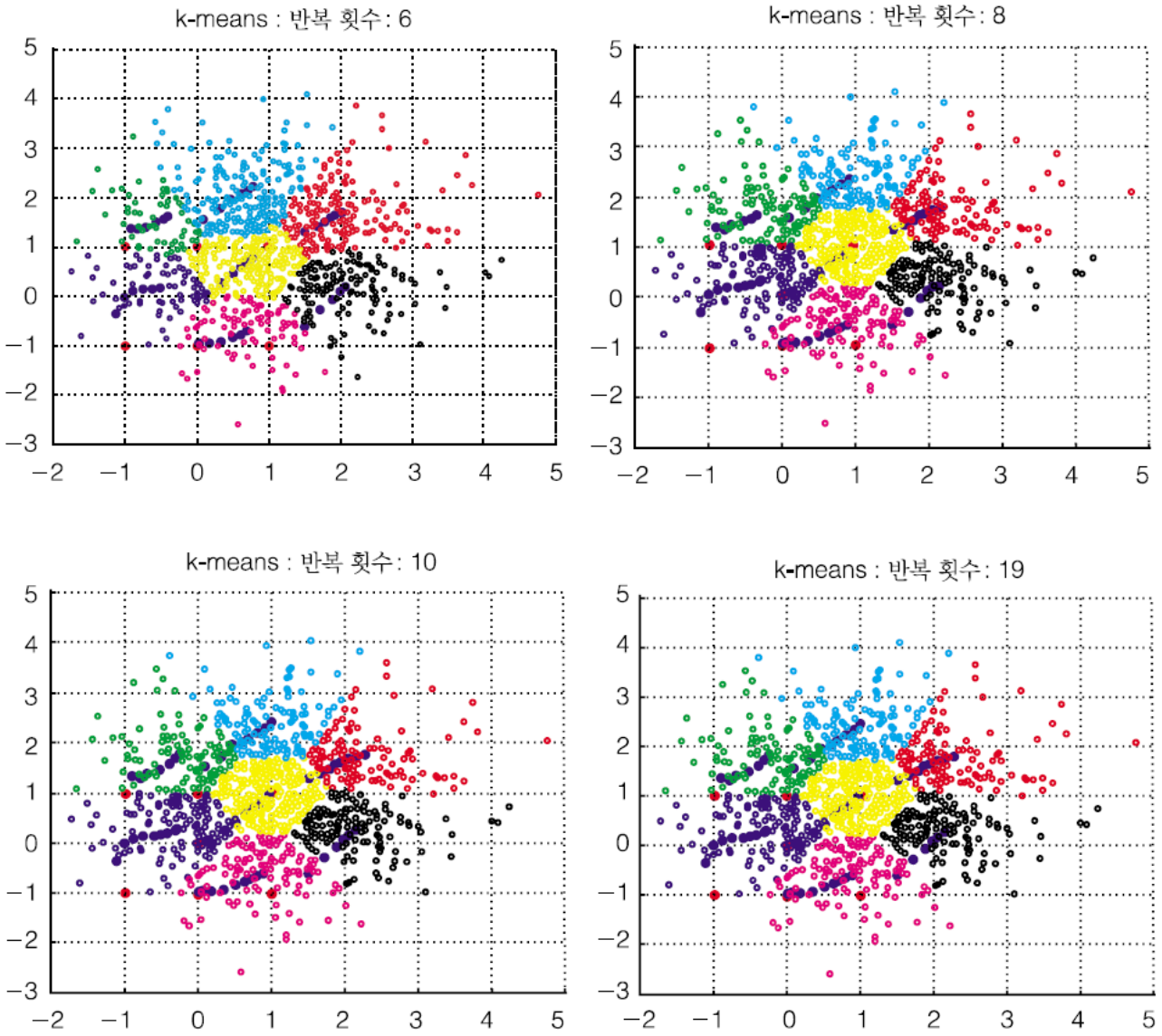
예시

실습 1 k-means 알고리즘을 이용한 벡터 양자화 시뮬레이션

k-means 알고리즘을 이용하여 벡터가 양자화되는 과정을 직접 시뮬레이션해 보자. 스크립트 파일을 실행하면, [그림 7-4]와 같이 붉은 점으로 표시된 클러스터링 중심을 초기값에서부터 시작하여 찾아가는 궤적을 확인해 볼 수 있다.

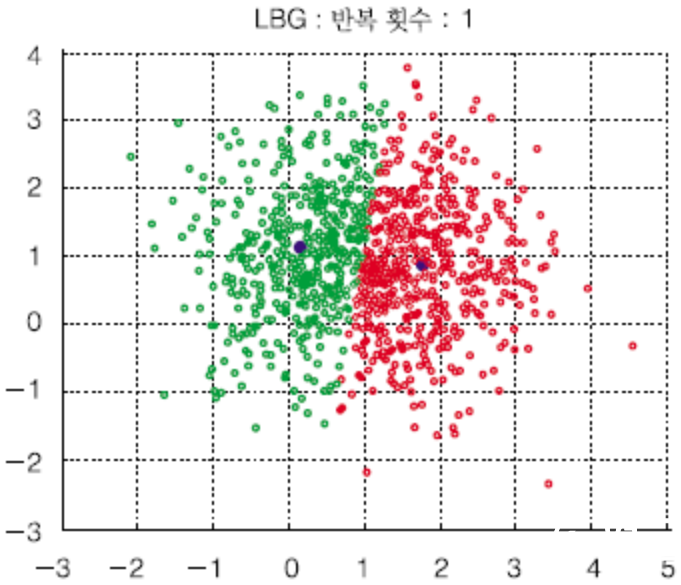


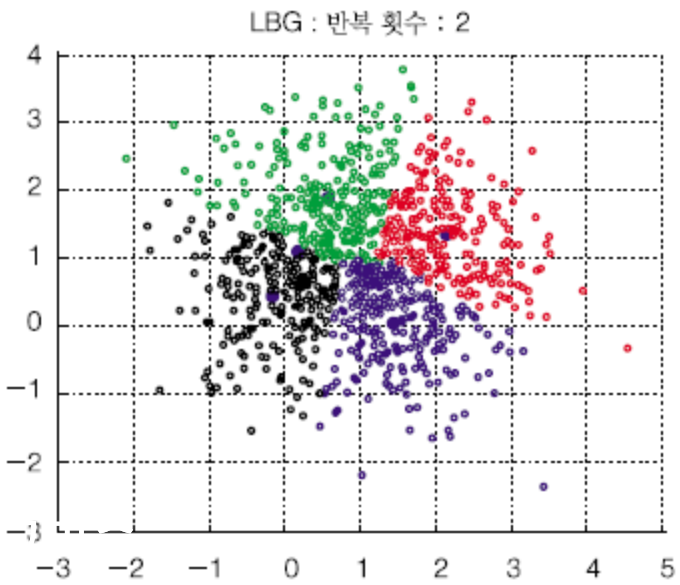
예시

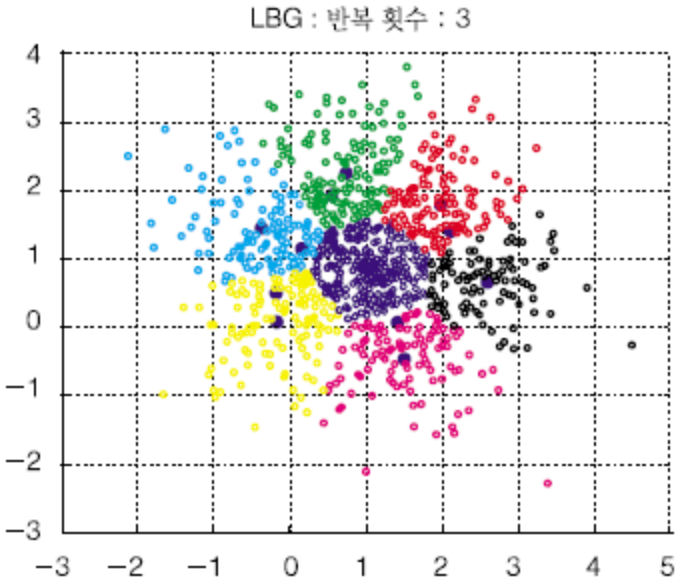


[그림 7-4] k-means 알고리즘을 이용한 클러스터링 시뮬레이션 결과

예시

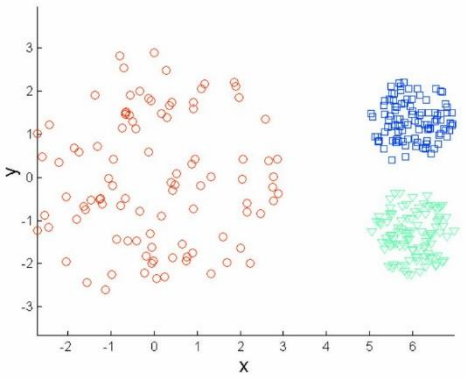




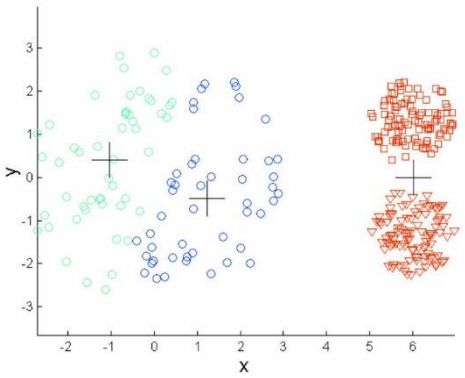


[그림 7-5] LBG 알고리즘을 이용한 클러스터링 시뮬레이션 결과

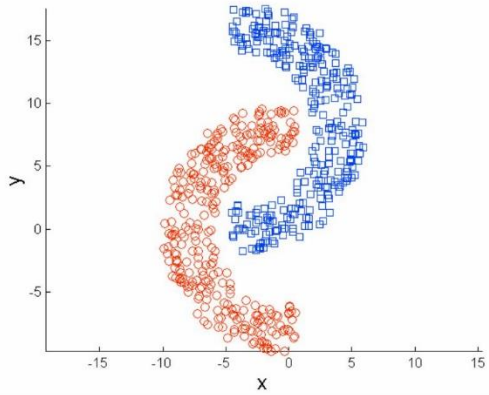
한계점



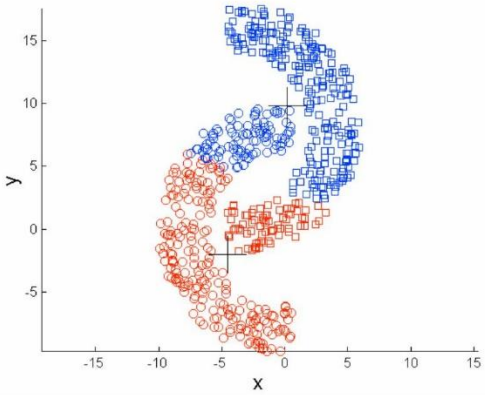
Original Points



K-means (3 Clusters)



Original Points



K-means (2 Clusters)

https://bayanbox.ir/view/2525938851245838506/Kmeans-ICDM06.pdf

k-평균 알고리즘 한계점 검증

예제 6-1 **k-평균의 동작**

[그림 6-5]는 훈련집합이 7개의 샘플을 가진 n=7인 예를 보여 준다. 좌표는 다음과 같다.

$$\mathbf{x}_1 = \begin{pmatrix}18\\5\end{pmatrix}, \mathbf{x}_2 = \begin{pmatrix}20\\9\end{pmatrix}, \mathbf{x}_3 = \begin{pmatrix}20\\14\end{pmatrix}, \mathbf{x}_4 = \begin{pmatrix}20\\17\end{pmatrix}, \mathbf{x}_5 = \begin{pmatrix}5\\15\end{pmatrix}, \mathbf{x}_6 = \begin{pmatrix}9\\15\end{pmatrix}, \mathbf{x}_7 = \begin{pmatrix}6\\20\end{pmatrix}$$

군집의 개수 k=3이라 하자. 맨 왼쪽 그림은 초기 군집 중심을 보여 준다. [알고리즘 6-1]의 라인 3~4는 7개 샘플을 아래와 같이 배정할 것이다.

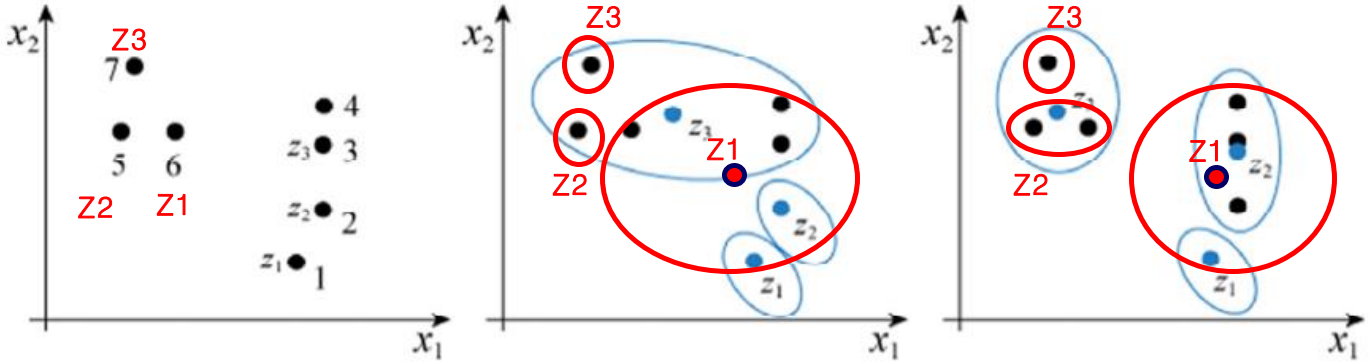


그림 6-5 k-평균의 동작 예제

6.3.1 k-평균 알고리즘

[그림 6-5]의 가운데 그림은 새로 계산한 군집 중심이다. $\mathbf{z}_1 = (18, 5)^{\mathrm{T}}$, $\mathbf{z}_2 = (20, 9)^{\mathrm{T}}$, $\mathbf{z}_3 = (12, 16.2)^{\mathrm{T}}$ 이고, 식 (6.2)에 대입하면 $J = 244.80$이 된다. 이때 거리함수 dist로 식 (1.7)의 유클리디언 거리를 사용한다.

두 번째 루프를 실행하면 행렬 **A**는 아래와 같이 바뀐다. 군집 중심은 $\mathbf{z}_1 = (18, 5)^{\mathrm{T}}$, $\mathbf{z}_2 = (20, 13.333)^{\mathrm{T}}$, $\mathbf{z}_3 = (6.667, 16.667)^{\mathrm{T}}$이다. 이것을 식 (6.2)에 대입하면 $J = 58.00$이 된다. [그림 6-5]의 맨 오른쪽 그림은 두 번째 루프 수행 후의 상황이다.

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

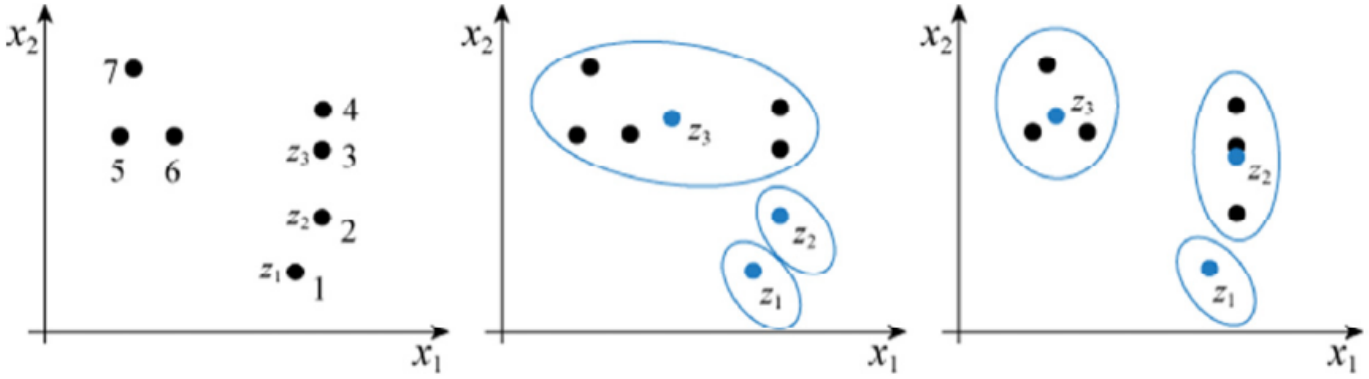


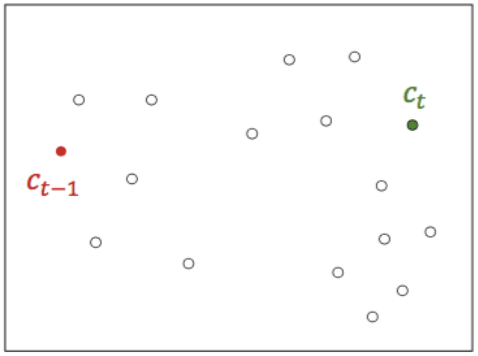
그림 6-5 k-평균의 동작 예제

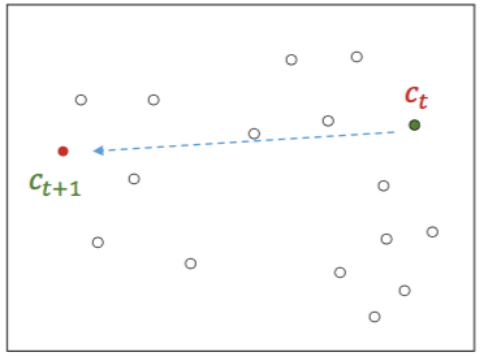
참고사항

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

k-means++ 초기화 방법

- 첫 initial point c_1은 임의로 선택
- 이후의 initial point c_t는 이전에 선택한 c_{t-1} 과의 거리가 큰 점이 선택되도록 확률분포를 만들고 이에 따라 초기화 중심점들을 선택





https://lovit.github.io/nlp/machine%20learning/2018/03/19/kmeans_initializer/

참고사항

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

주요 패키지	버전
librosa	0.8.1
matplotlib	3.4.3
numba	0.53.1
numpy	1.21.2
pandas	1.3.2
plotly	5.3.1
scikit-learn	1.1.3
scipy	1.7.1
seaborn	0.11.2
sklearn	0. 0