

Machine Learning

Contents

1. Introduction
2. K-Nearest Neighbor Algorithm
3. LDA(Linear Discriminant Analysis)
4. Perceptron
5. Feed-Forward Neural Networks
6. RNN(Recurrent Neural Networks)
7. SVM(Support Vector Machine)
8. Ensemble Learning
9. CNN(Convolutional Neural Network)
10. PCA(Principal Component Analysis)
11. ICA(Independent Component Analysis)
- 12. Clustering**
13. GAN(Generative Adversarial Network)

12.1. Clustering

- Classification (known categories)
- Clustering (creation of new categories)



그림 12.1. 국기들의 대륙별 분류

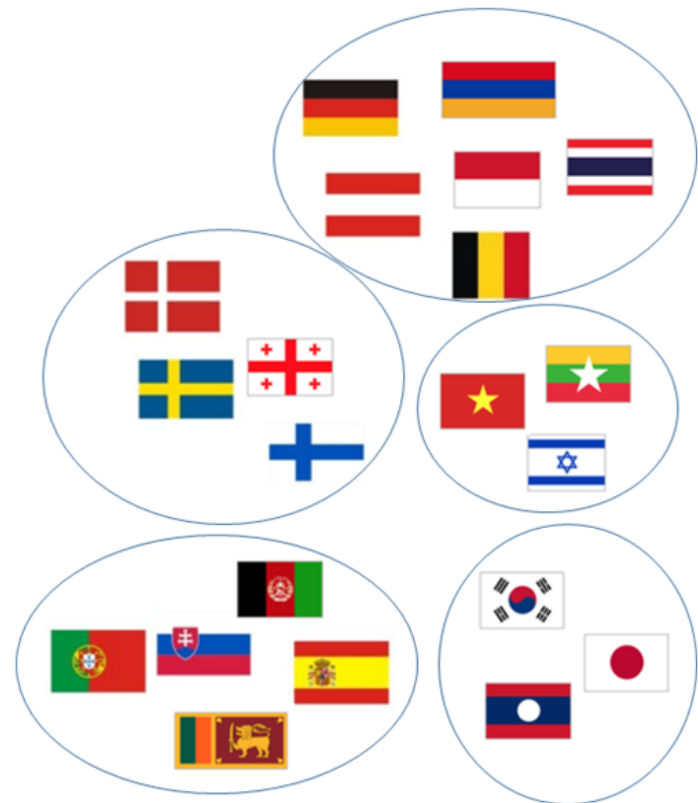


그림 12.2. 국기들의 클러스터

Clustering

- Clustering: grouping a set of objects into classes of similar objects
 - Subjective process
- Unsupervised learning: learning from unlabelled data, as opposed to supervised data where a classification of examples is given
- Many applications in science and engineering (data mining, statistical data analysis, pattern recognition, image analysis, information retrieval, bioinformatics, data compression, computer graphics)
- Characteristics
 - Dense in a certain area of space
 - high intra-class similarity
 - low inter-class similarity
 - It is the commonest form of unsupervised learning

12.2. Issues for clustering

- "Groupness": What is a natural grouping among these objects?
- "Similarity/Distance": What makes objects related?
- "Representation": How do we represent objects? Vectors? Do we normalise?
- "Parameters": How many clusters? Fixed a priori? Data-driven?
- "Algorithms": Partitioning the data? Hierarchical algorithm?
- Formal foundation and convergence

Groupness: What is a natural grouping among these objects?

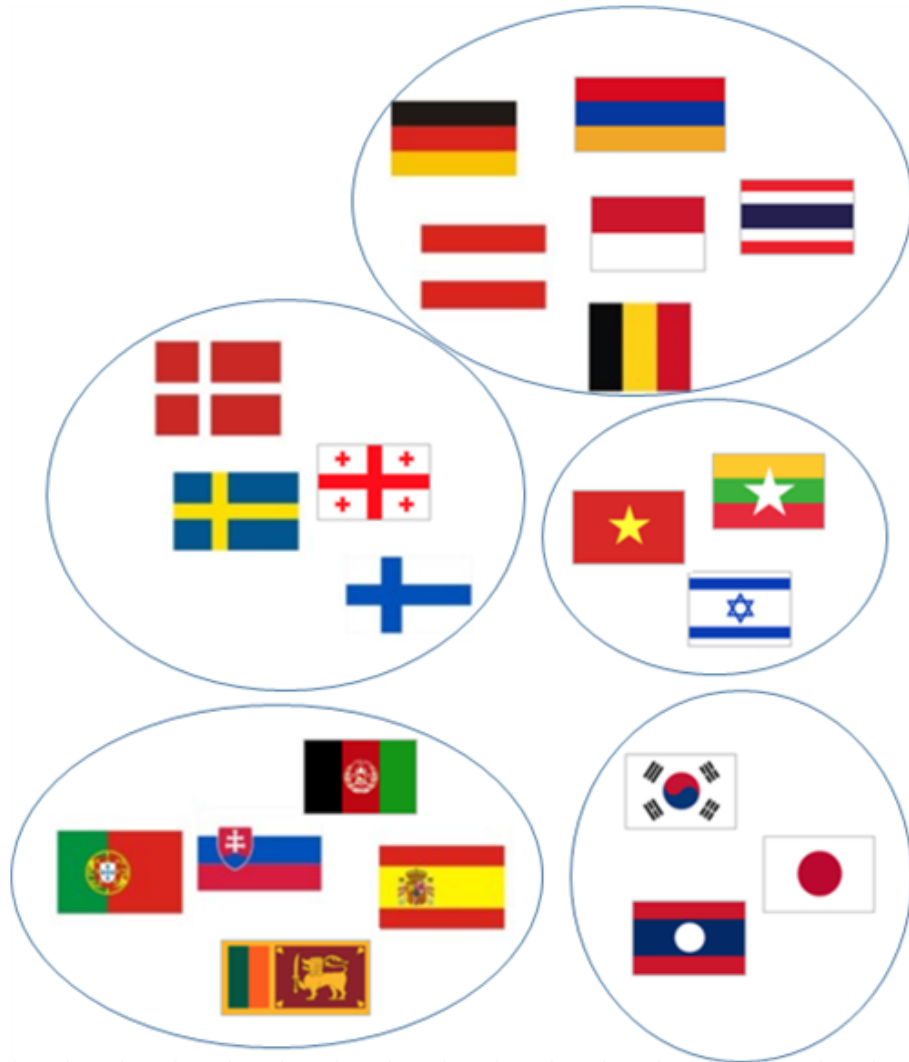


그림 12.2. 국가들의 클러스터

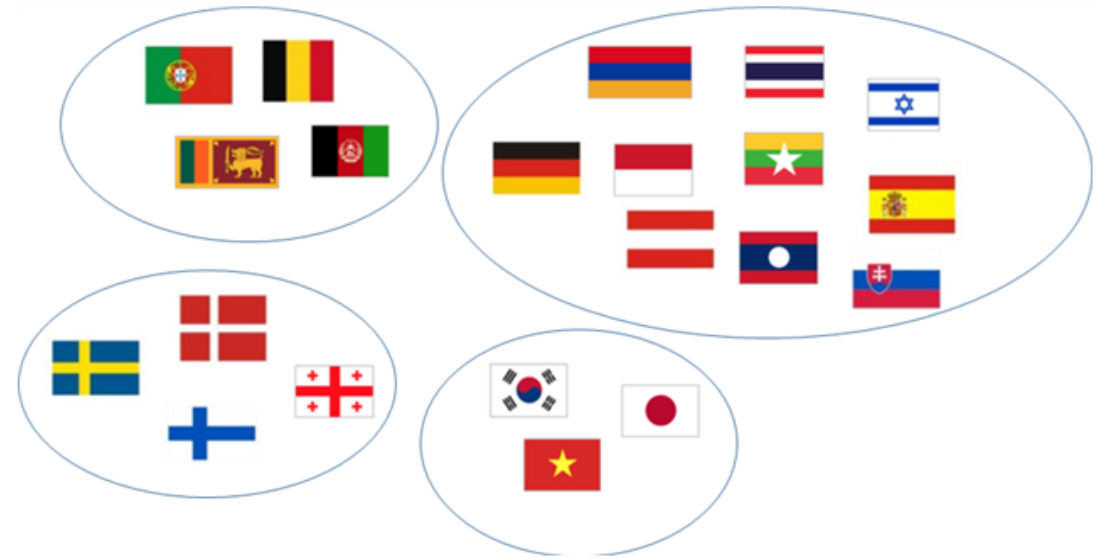


그림 12.3. 국가들의 클러스터2

How do we define similarity?



Hard to define!
But we *know it*
when we see it

A distance measure : properties?

- Symmetry

$$d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$$

- Self-similarity

- Triangular-Inequality

$$d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}) \geq d(\mathbf{x}, \mathbf{z})$$

Distance measures

- Two objects (p features)

$$\mathbf{x} = (x_1, x_2, \dots, x_p) \text{ and } \mathbf{y} = (y_1, y_2, \dots, y_p) \quad (12.2.1)$$

- The Minkowski Distance Metric

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^p |x_i - y_i|^r \right)^{1/r} \quad (12.2.2)$$

- Euclidean distance, Manhattan distance, supreme distance, infimum distance

An example

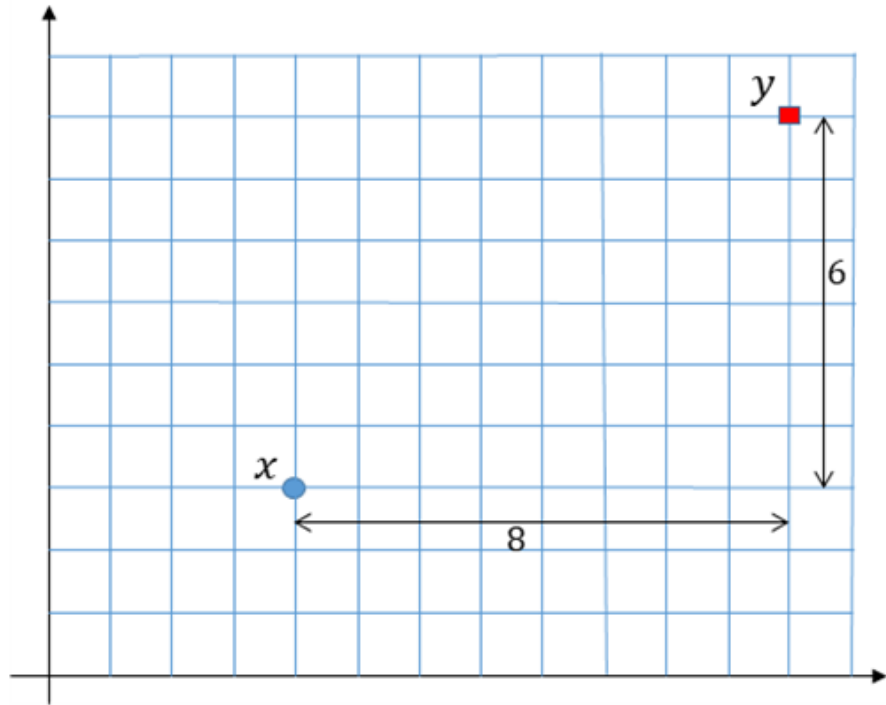


그림 12.4. 2차원 공간상의 두 점 위치

$$\text{Euclidean } d(\mathbf{x}, \mathbf{y}) = \sqrt{8^2 + 6^2} = 10 \quad (12.2.3)$$

$$\text{Manhattan } d(\mathbf{x}, \mathbf{y}) = 8 + 6 = 14 \quad (12.2.4)$$

$$\text{Sup } d(\mathbf{x}, \mathbf{y}) = \max\{8, 6\} = 8 \quad (12.2.5)$$

$$\text{Inf } d(\mathbf{x}, \mathbf{y}) = \min\{8, 6\} = 6 \quad (12.2.6)$$

두 대상 x 와 y 의 비슷함을 측정하는 기준으로는 내적(Inner Product)

$$\langle x, y \rangle = \|x\| \|y\| \cos(\theta) = \sum_{i=1}^p x_i y_i \quad (12.2.7)$$

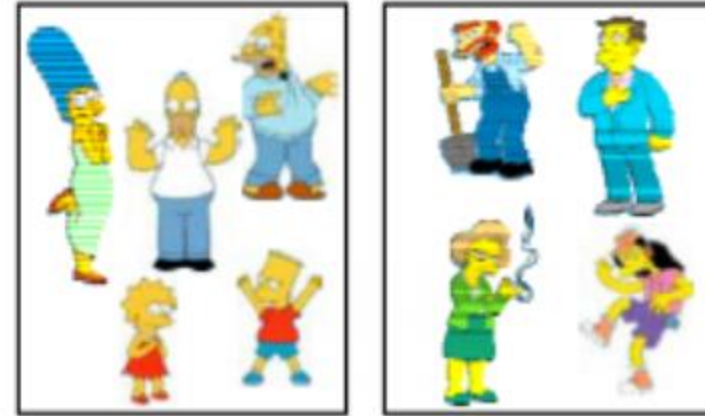
을 생각할 수 있다. 이를 정규화 시킨 Cosine Similarity는

$$\cos(\theta) = \frac{\langle x, y \rangle}{\|x\| \|y\|} = \frac{\sum_{i=1}^p x_i y_i}{\sqrt{\sum_{i=1}^p x_i^2} \sqrt{\sum_{i=1}^p y_i^2}} \quad (12.2.8)$$

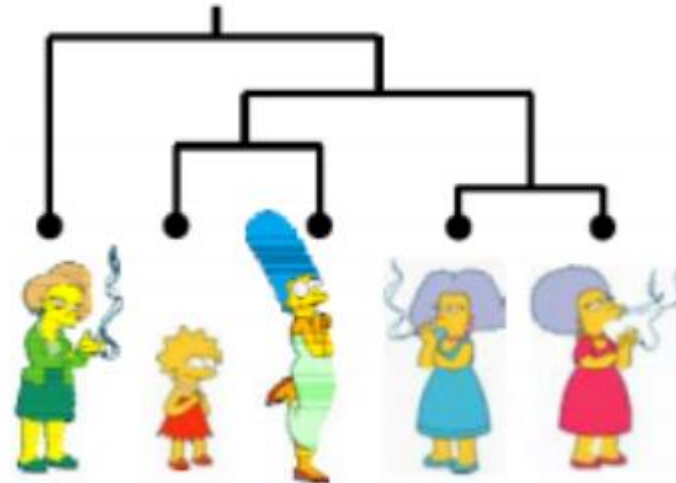
로 정의된다.

Clustering algorithms

- Partitional Algorithm
 - Usually start with a random partitioning
 - K means clustering



- Hierarchical Algorithm
 - Bottom-up
 - Top-down



Clustering algorithms

상향식 집적 클러스터링:

- ① 각 대상이 개별적으로 하나의 클러스터에 배정되게 한 후
- ② 가장 가까운 쌍을 반복적으로 합치면서 상위 클러스터를 만들어
- ③ 하나의 클러스터가 될 때까지 이 과정을 반복함.

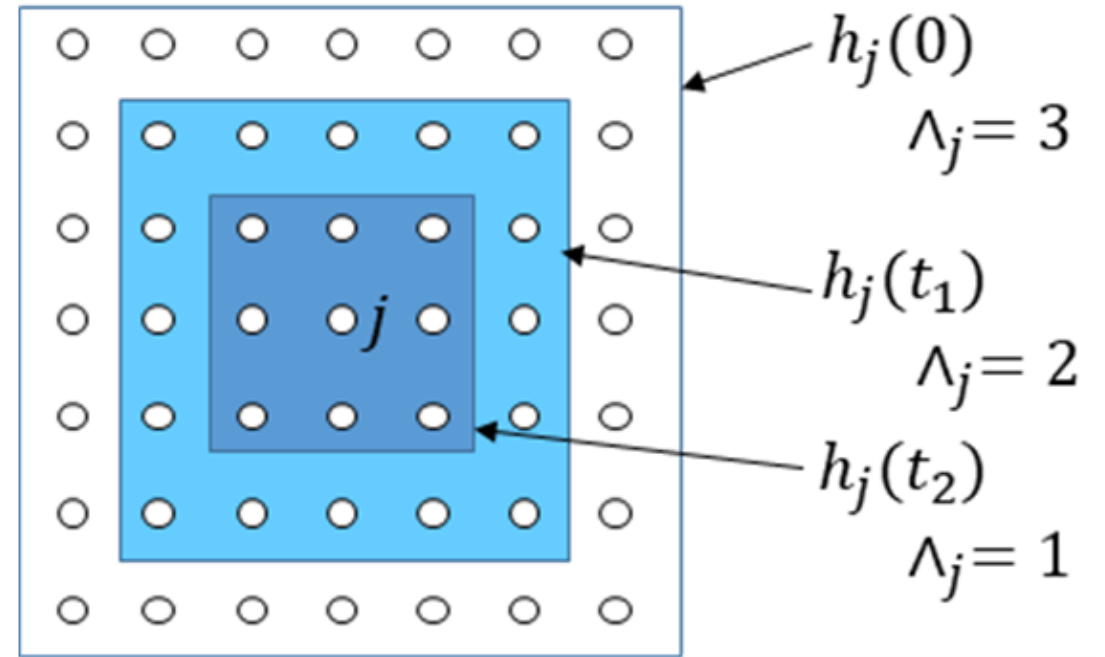
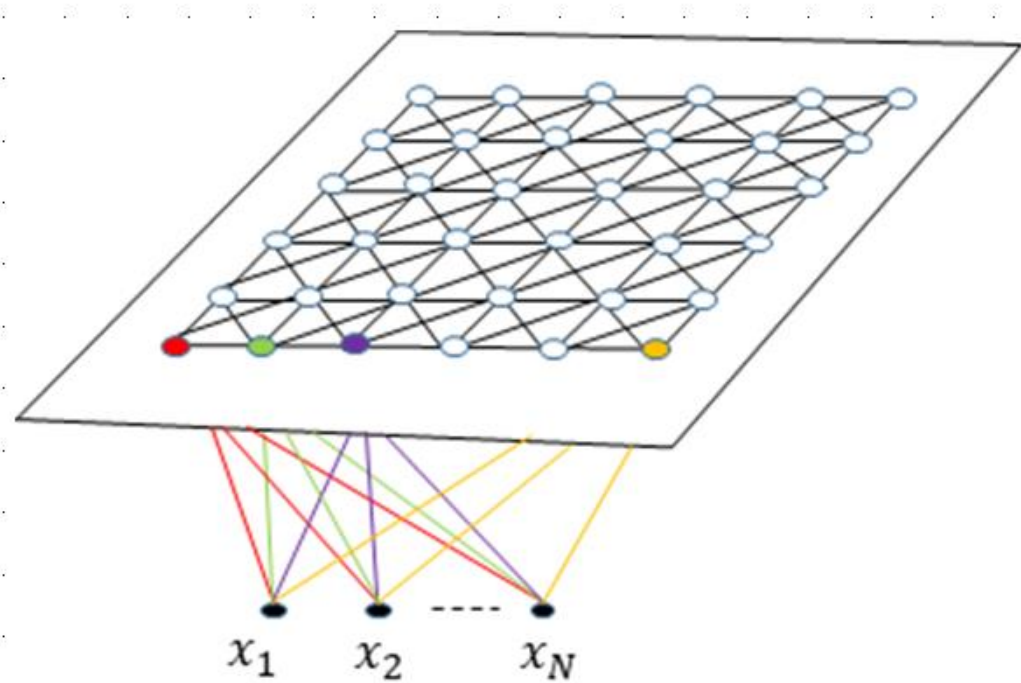
하향식 구분 클러스터링:

- ① 모든 대상을 하나의 클러스터로 묶은 후
- ② 한 클러스터를 두 개의 클러스터로 나누는 가능한 방법 중에서 최상의 결과를 택하며
- ③ 이 과정을 모든 클러스터를 대상으로 반복함.

Distance between clusters

- Single-link nearest neighbor: distance between their closest members
- Complete-link farthest neighbor: distance between their farthest members
- Centroid: distance between the centers of gravity of the two clusters
- Average: distance between the average of all cross-cluster pairs

12.3. SOM(Self-Organizing Map)



SOM 학습 알고리즘

① 초기화

N 입력노드에서 M 출력노드로 연결된 가중치 벡터들을 임의의 작은 값으로 초기화 한다. 그리고, 위상정보의 초기 이웃 반경 λ_j 를 설정한다.

② 입력 부여

새로운 입력벡터 x 를 입력노드에 부여한다.

③ 모든 출력노드의 거리 계산

모든 출력노드가 지닌 가중치 벡터와 입력벡터 사이의 거리

$$d_j = \sum_{i=1}^N (x_i - w_{ji}(t))^2 \quad (12.3.1)$$

를 계산한다. 여기서, t 는 시간 인덱스이다.

④ 최소거리 출력노드 선정

모든 출력노드들의 거리 중에서 가장 거리가 작은 출력노드를

$$j^* = \arg \min_j d_j \quad (12.3.2)$$

와 같이 결정한다. 여기서, 출력노드 y_{j^*} 를 BMU(Best Matching Unit)이라고 한다.

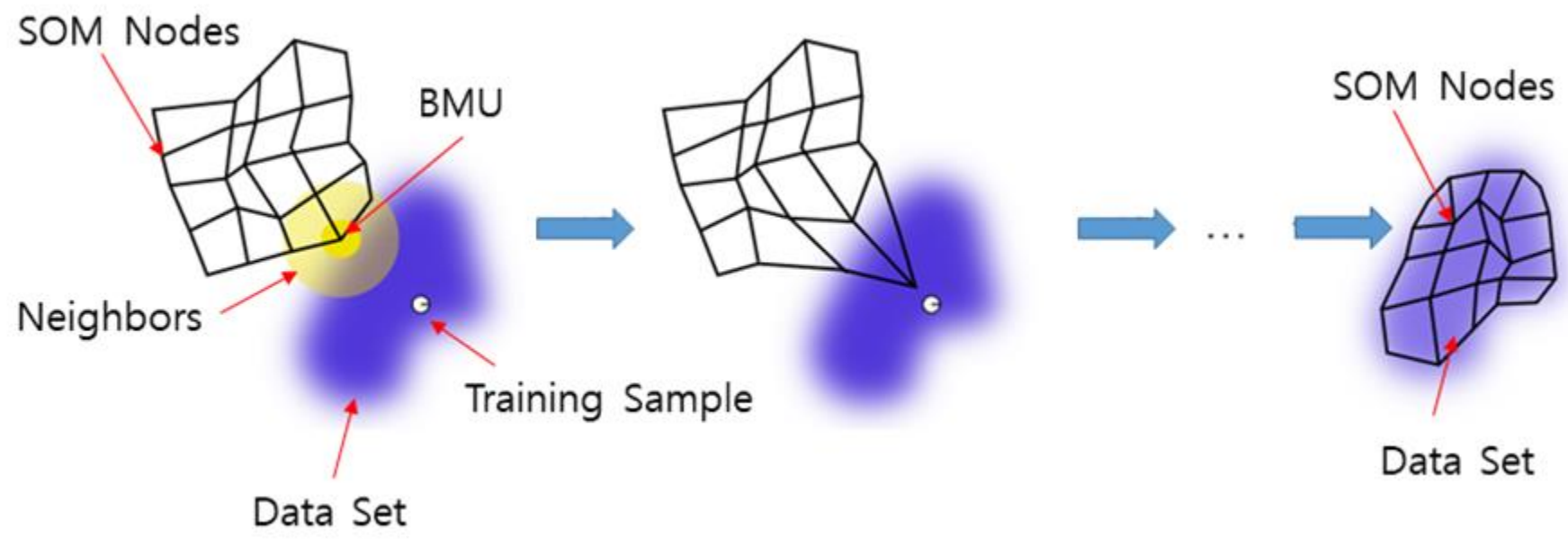
⑤ BMU와 이웃 노드의 가중치 변경

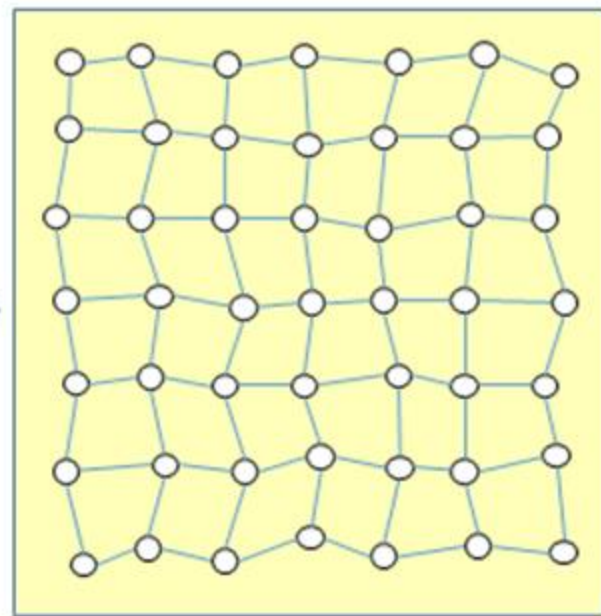
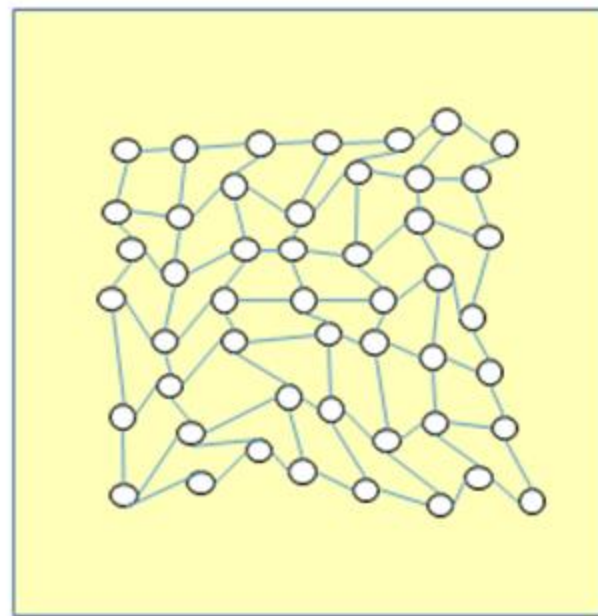
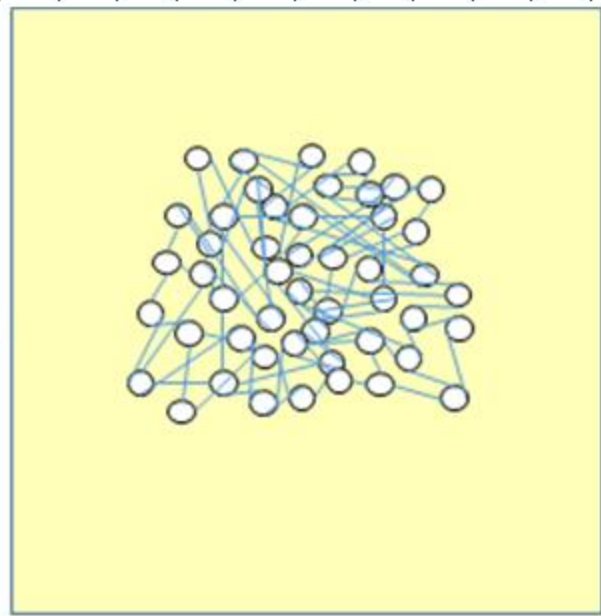
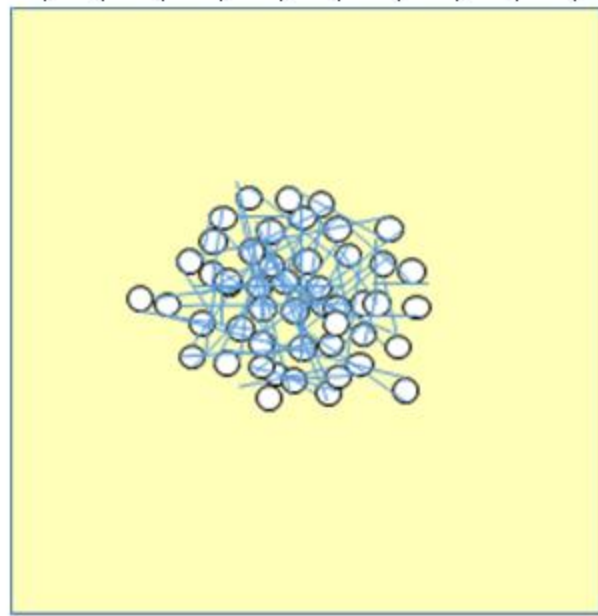
BMU y_{j^*} 와 $h_{j^*}(t)$ 에 의해 결정된 이웃 출력노드들의 가중치를

$$w_{ji}(t+1) = w_{ji}(t) + \eta(t)h_{j^*}(t)(x_i - w_{ji}(t)) \quad (12.3.3)$$

에 따라 변경시킨다. 여기서, $\eta(t)$ 는 학습률이다.

⑥ ②부터 반복한다.





12.4. Conscience Learning

$$y_j = \begin{cases} 1 & \text{if } \|\mathbf{w}_j - \mathbf{x}\|^2 \leq \|\mathbf{w}_i - \mathbf{x}\|^2 \forall i \neq j \\ 0 & \text{otherwise} \end{cases} \quad (12.4.1)$$

$$p_j(t+1) = p_j(t) + B(y_j(t) - p_j(t)) \quad (12.4.2)$$

$$b_j = C(1/M - p_j) \quad (12.4.3)$$

$$z_j = \begin{cases} 1 & \text{if } \|\mathbf{w}_j - \mathbf{x}\|^2 - b_j \leq \|\mathbf{w}_i - \mathbf{x}\|^2 - b_i \forall i \neq j \\ 0 & \text{otherwise} \end{cases} \quad (12.4.4)$$

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \eta(t)(\mathbf{x} - \mathbf{w}_j(t))z_j \quad (12.4.5)$$



Figure 8. Probability density function showing regions of equal area.

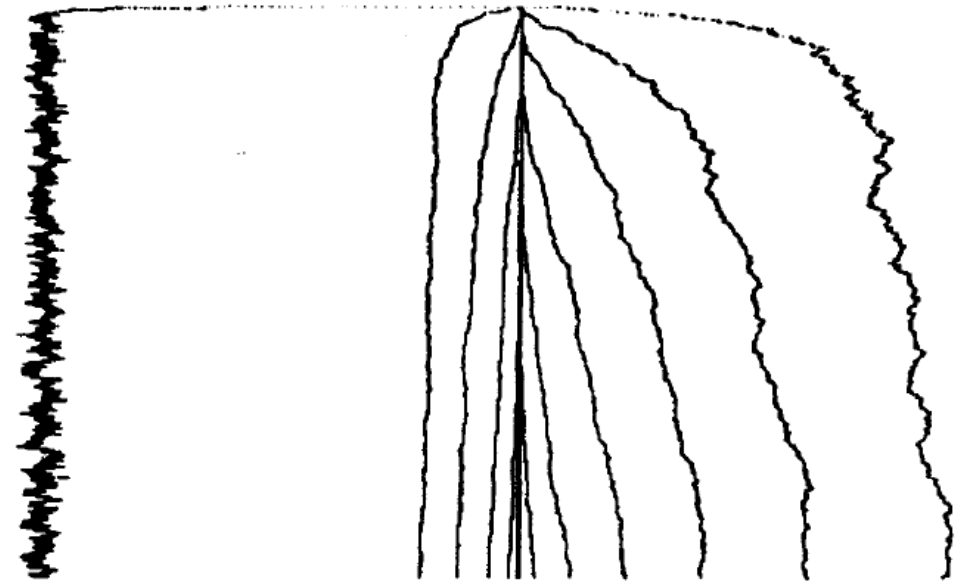


Figure 9. Kohonen learning. $A = 0.03$ for 16000 iterations.

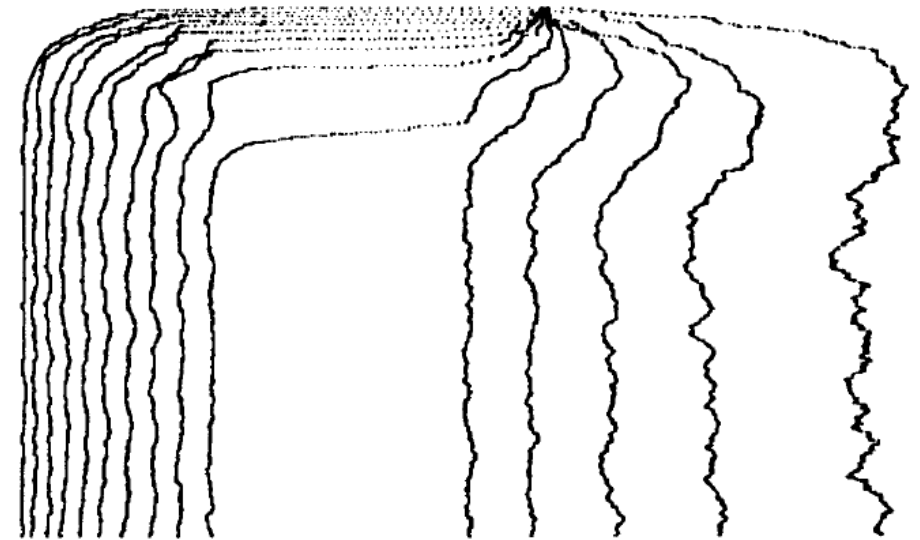


Figure 10. Conscience learning. $A = 0.03$ for 16000 iterations.

12.5. K-means Clustering

- Vector Quantization
 - 24bits/pixel (16M colors) \Rightarrow 8 bits/pixel (256 colors)
 - Uniform quantizing ([0, 65535] \rightarrow color 0) : wastes the color map
 - Find k reference vectors which best represent the data
- Given $X = \{\mathbf{x}^t\}$, find k reference vectors $\mathbf{m}_j, j = 1, 2, \dots, k$

$$\|\mathbf{x}^t - \mathbf{m}_i\| = \min_j \|\mathbf{x}^t - \mathbf{m}_j\|$$

$$E(\{\mathbf{m}_i\} | X) = \sum_t \sum_i b_i^t \|\mathbf{x}^t - \mathbf{m}_i\|^2$$

$$b_i^t = \begin{cases} 1 & \text{if } \|\mathbf{x}^t - \mathbf{m}_i\| = \min_j \|\mathbf{x}^t - \mathbf{m}_j\| \\ 0 & \text{otherwise} \end{cases}$$

- k means Clustering Algorithm

- ① Assignment Step

모든 $x^t \in X$ 에 대하여

$$b_i^t \leftarrow \begin{cases} 1 & \text{if } \|x^t - m_i\| = \min_j \|x^t - m_j\| \\ 0 & \text{otherwise} \end{cases} \quad (12.5.4)$$

에 따라 b_i^t 를 할당한다. 그 결과 클러스터 S_i 는

$$S_i = \{x^t : \|x^t - m_i\| \leq \|x^t - m_j\| \forall j, 1 \leq j \leq k\} \quad (12.5.5)$$

와 같이 구성된다.

- ① Update Step

모든 $m_i (i = 1, 2, \dots, k)$ 에 대하여

$$m_i \leftarrow \frac{\sum_t b_i^t x^t}{\sum_i b_i^t} \quad (12.5.6)$$

와 같이 m_i 를 갱신한다.

$$b_i^t \leftarrow \begin{cases} 1 & \text{if } \|x^t - m_i\| = \min_j \|x^t - m_j\| \\ 0 & \text{otherwise} \end{cases} \quad (12.5.4)$$

k-평균 알고리즘

① 파라미터 설정

k 값을 결정한다.

② 초기화

k 클러스터의 중심 기준벡터 $m_j (j = 1, 2, \dots, k)$ 를 임의로 초기화 한다.

③ 할당 단계

모든 $x^t \in X$ 에 대하여 b_i^t 를 식 (12.5.4)에 따라 할당한다.

④ 갱신 단계

모든 $m_i (i = 1, 2, \dots, k)$ 에 대하여 식 (12.5.6)에 따라 m_i 를 갱신한다.

⑤ 종료/갱신

b_i^t 의 변동이 없으면 종료한다. 그렇지 않으면, ③부터 반복한다.

$$m_i \leftarrow \frac{\sum_t b_i^t x^t}{\sum_i b_i^t} \quad (12.5.6)$$

참조: 오차함수 최적화와 m_i 갱신

식 (12.5.6)과 같이 m_i 를 변경하는 것이 식 (12.5.2)로 주어진 오차함수를 최적화 시키는 지에 대하여 알아보자. 식 (12.5.2)로 주어진 오차함수를 다시 적으면

$$E(\{m_i\}|X) = \sum_t \sum_i b_i^t \|x^t - m_i\|^2 = \sum_t \sum_i b_i^t \sum_j (x_j^t - m_{ij})^2 \quad (12.5.7)$$

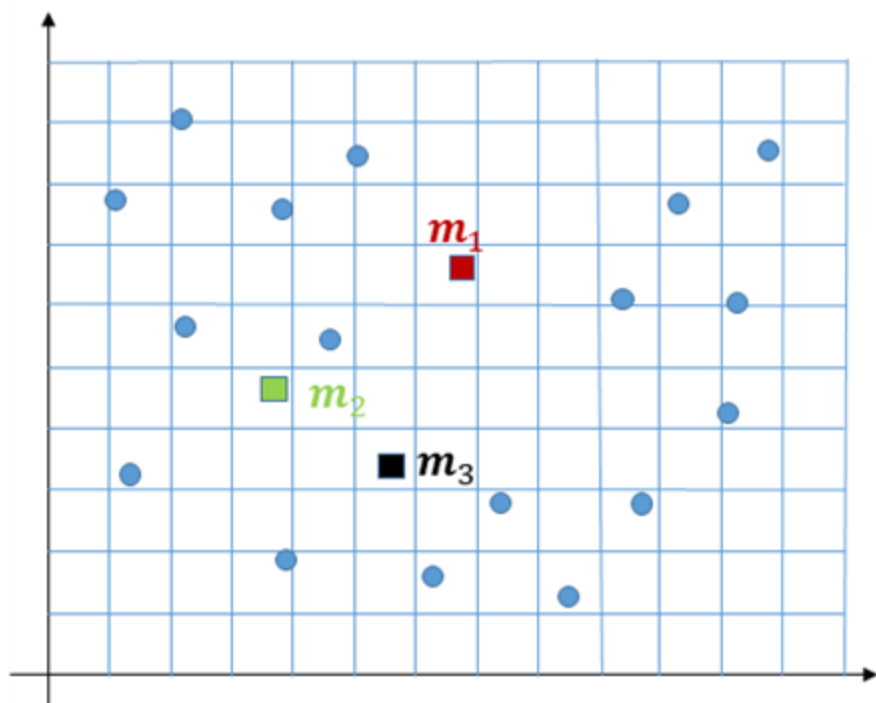
이 된다. 위 식을 m_{ij} 에 대하여 편미분을 하면

$$\frac{\partial E(\{m_i\}|X)}{\partial m_{ij}} = \frac{\partial [\sum_t \sum_i b_i^t \sum_j (x_j^t - m_{ij})^2]}{\partial m_{ij}} = \sum_t 2b_i^t (x_j^t - m_{ij}) \quad (12.5.8)$$

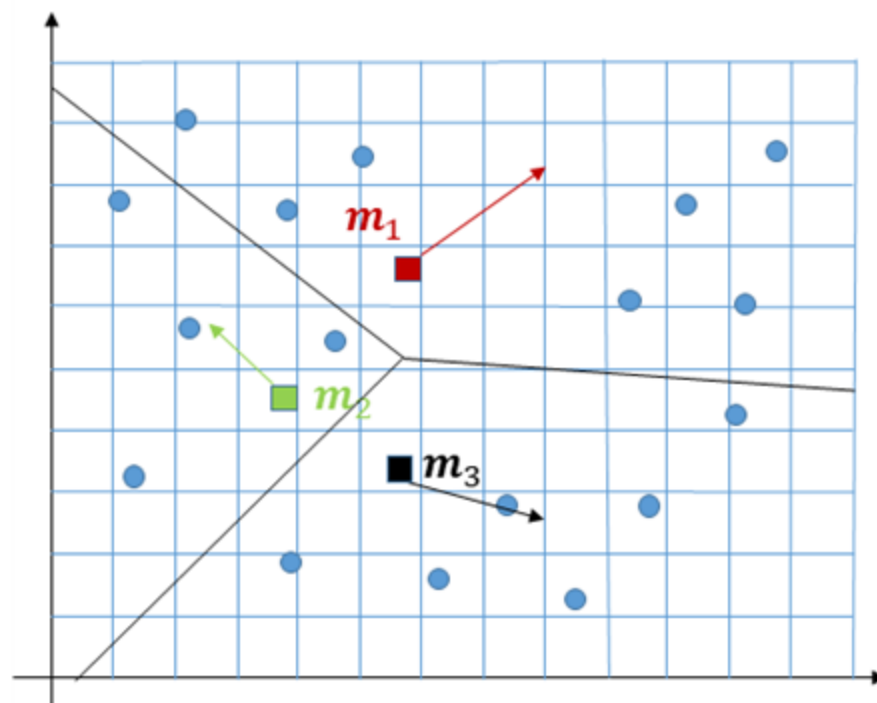
이 된다. 이 편미분이 0이 되는 조건에 의해

$$m_{ij} = \frac{\sum_t b_i^t x_j^t}{\sum_i b_i^t} \quad (12.5.9)$$

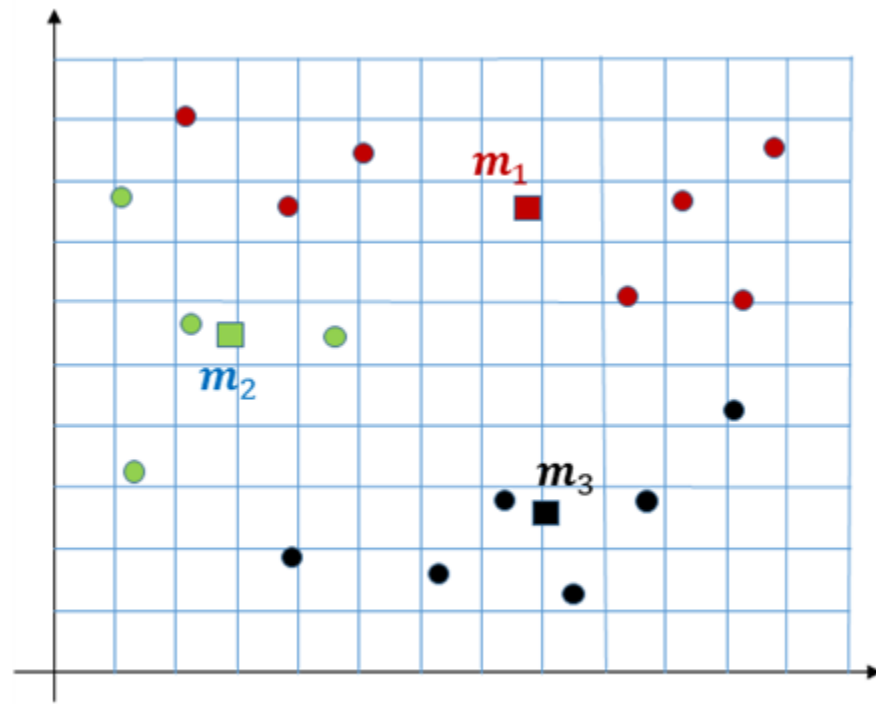
을 얻게 되며, 이를 벡터로 표현하면 식 (12.5.6)이 된다.



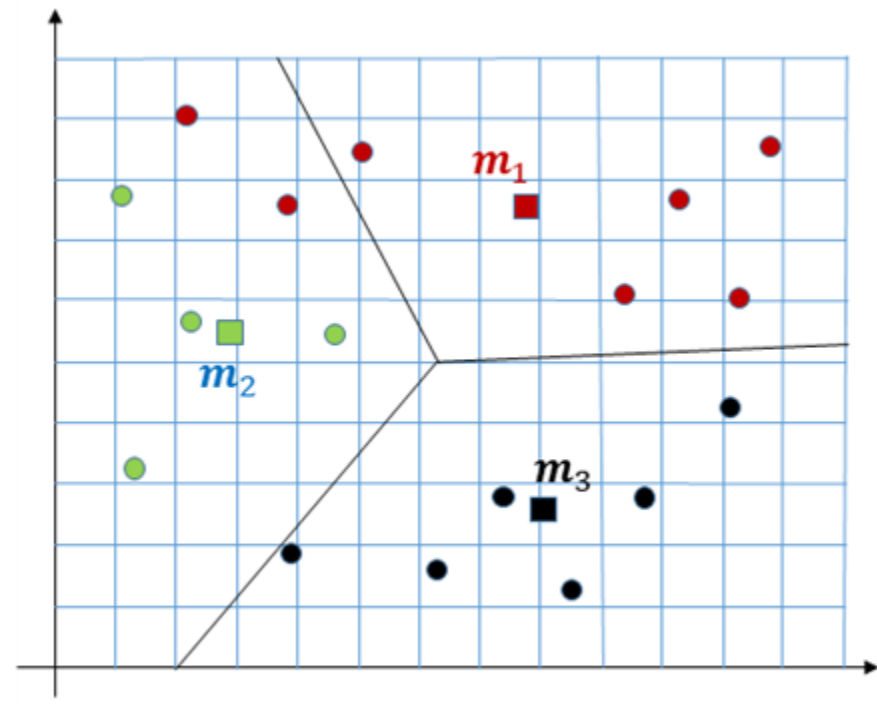
① Initialization of Centroids



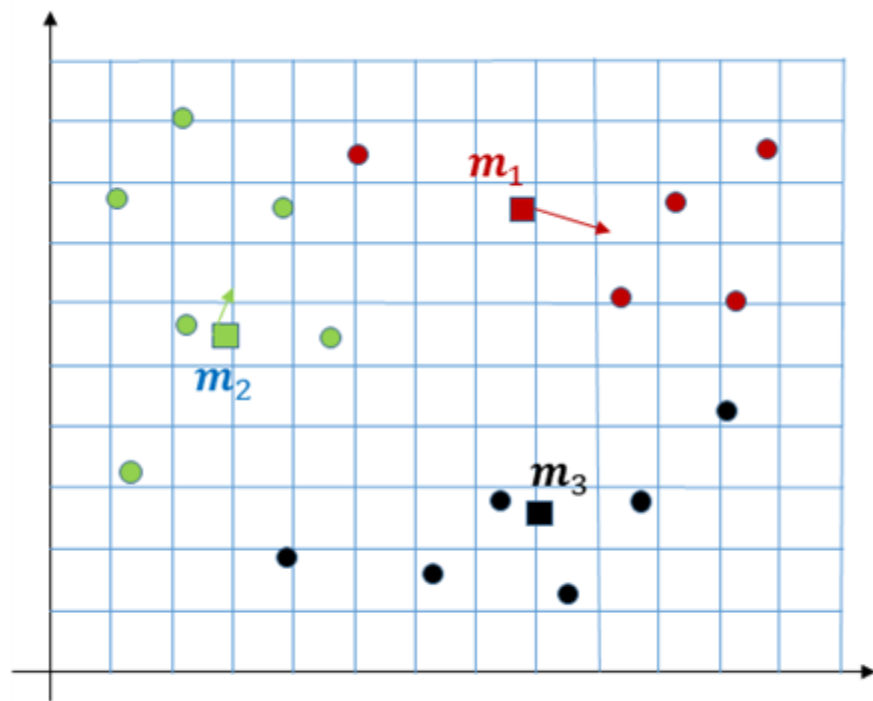
② Membership Assignment



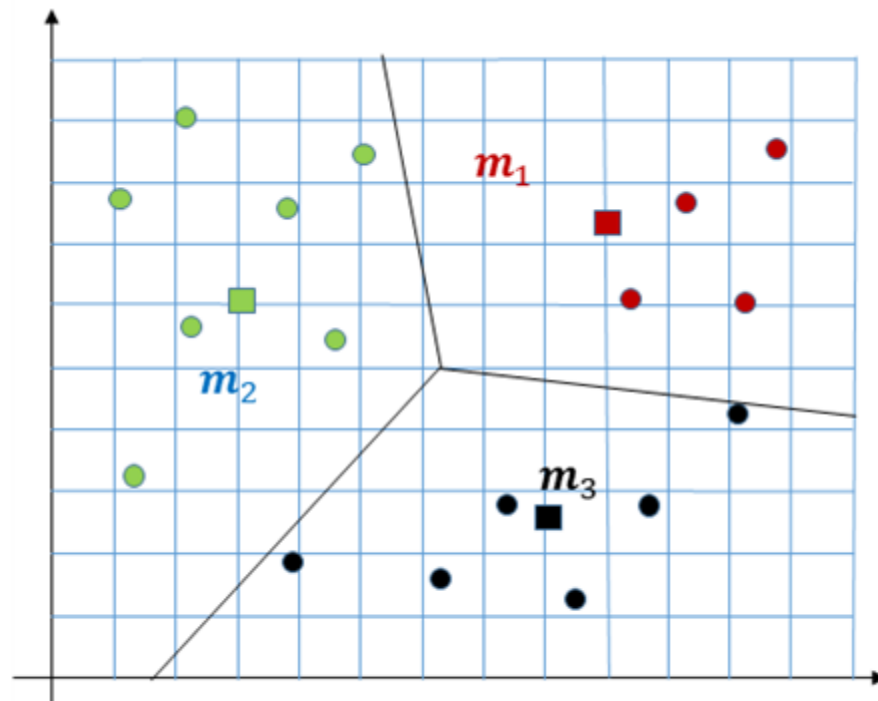
③ Updating Centroids



④ Membership Assignment



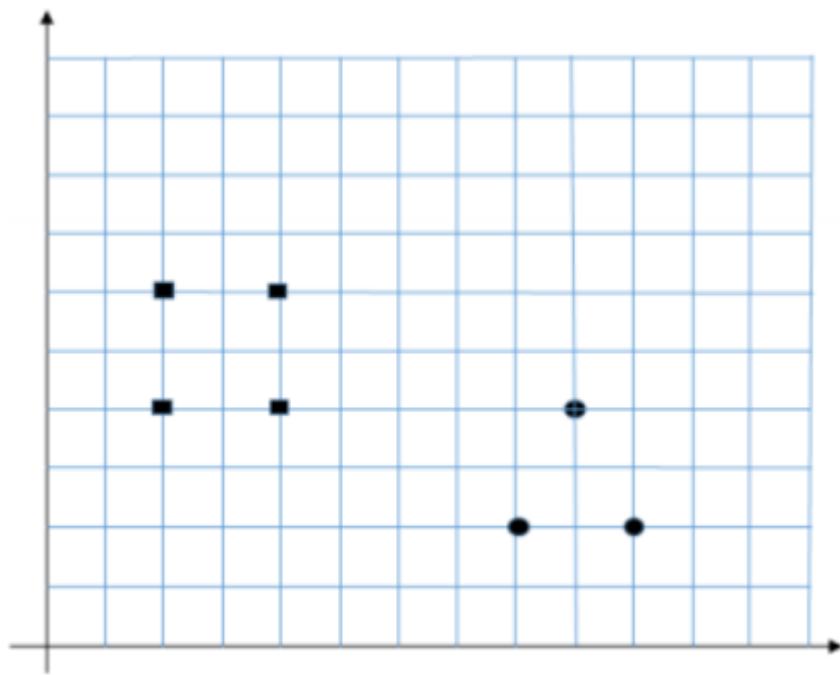
⑤ Updating Centroids



⑥ Membership Assignment

예제 12.5-1

그림과 같이 2차원 공간상에 점집합이 주어졌다. k -평균 알고리즘에 의해 두 개의 클러스터로 만들기 위하여 초기 기준벡터가 $(1,1)$ 과 $(10,10)$ 으로 주어진 경우에 할당단계와 갱신단계를 한번 씩 수행하여 보아라. 거리 기준은 Euclidean Distance를 사용하라.



12.6. Evaluation

Evaluation (or "validation") of clustering results is as difficult as the clustering itself. Popular approaches involve "**internal**" evaluation, where the clustering is summarized to a single quality score, "**external**" evaluation, where the clustering is compared to an existing "ground truth" classification, "**manual**" evaluation by a human expert, and "**indirect**" evaluation by evaluating the utility of the clustering in its intended application.

Internal evaluation

Davies–Bouldin index

$$DB = \frac{1}{k} \sum_{i=1}^k \max_{j \neq i} \left(\frac{\mu_i + \mu_j}{d(\mathbf{m}_i, \mathbf{m}_j)} \right)$$

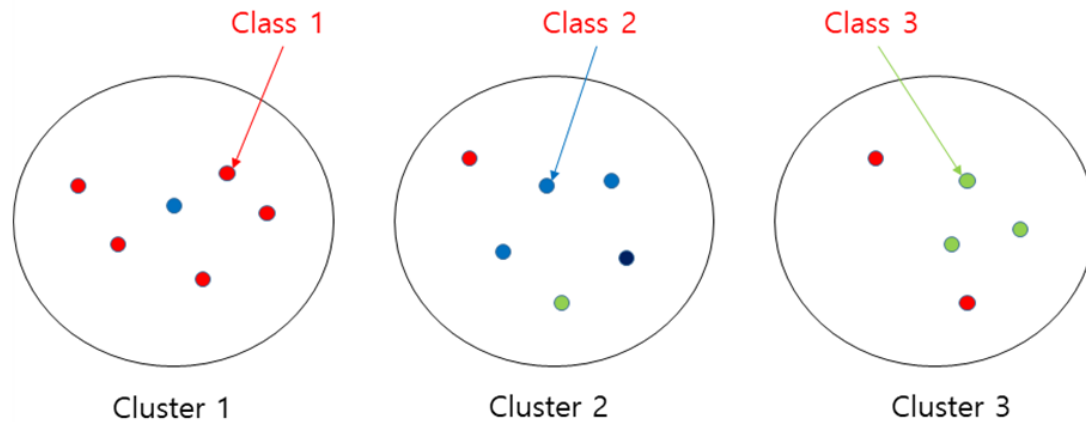
Dunn index

$$D = \frac{\min_{1 \leq i < j \leq k} d(i, j)}{\max_{1 \leq i \leq k} d'(i)}$$

External evaluation

Purity

$$Purity(S_i) = \frac{1}{n_i} \max_{j \in C} (n_{ij})$$



$$Purity(S_1) = \frac{1}{6} \max(5, 1, 0) = \frac{5}{6}$$
$$Purity(S_2) = \frac{1}{6} \max(1, 4, 1) = \frac{4}{6}$$
$$Purity(S_3) = \frac{1}{5} \max(2, 0, 3) = \frac{3}{5}$$