Machine Learning

Contents

12.1. Clustering

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

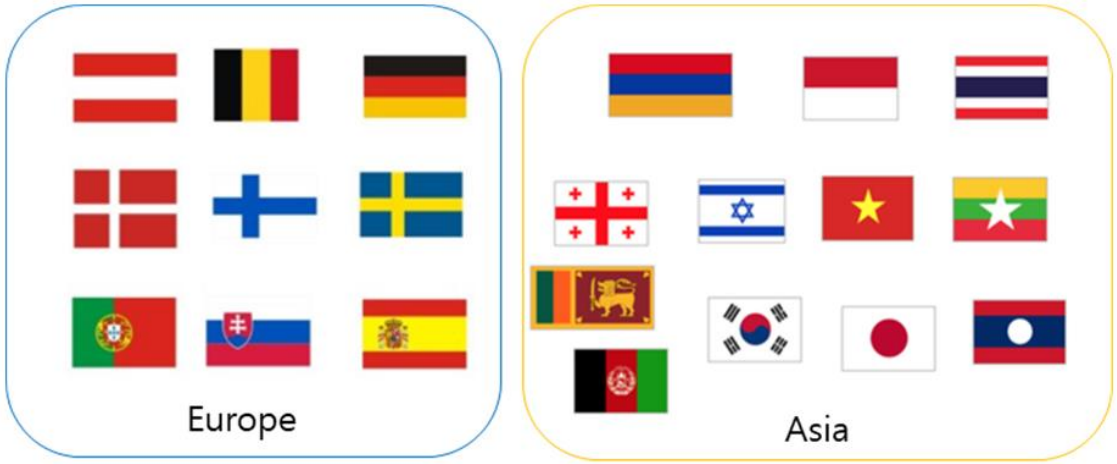


그림 12.1. 국기들의 대륙별 분류

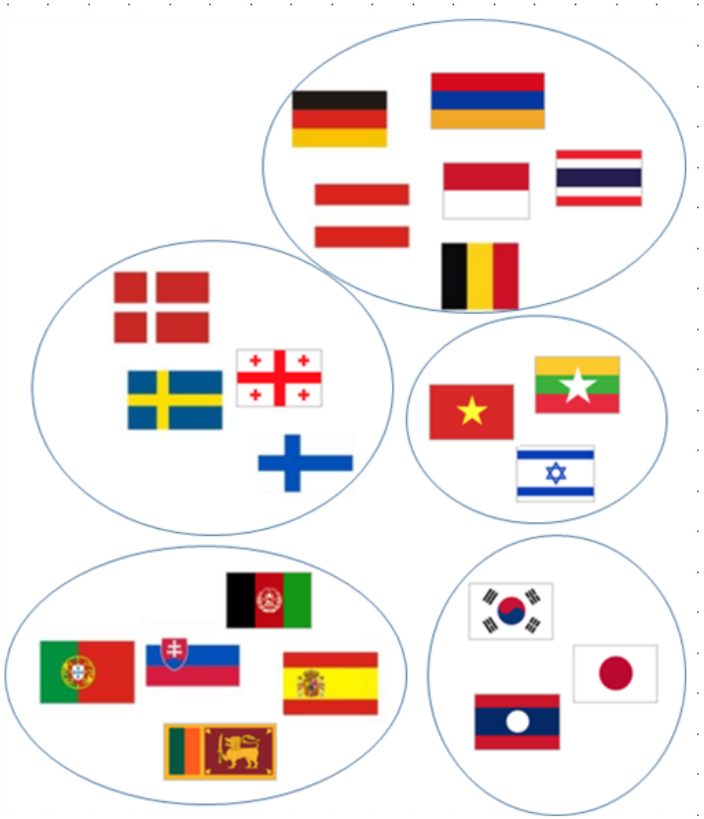


그림 12.2. 국기들의 클러스터

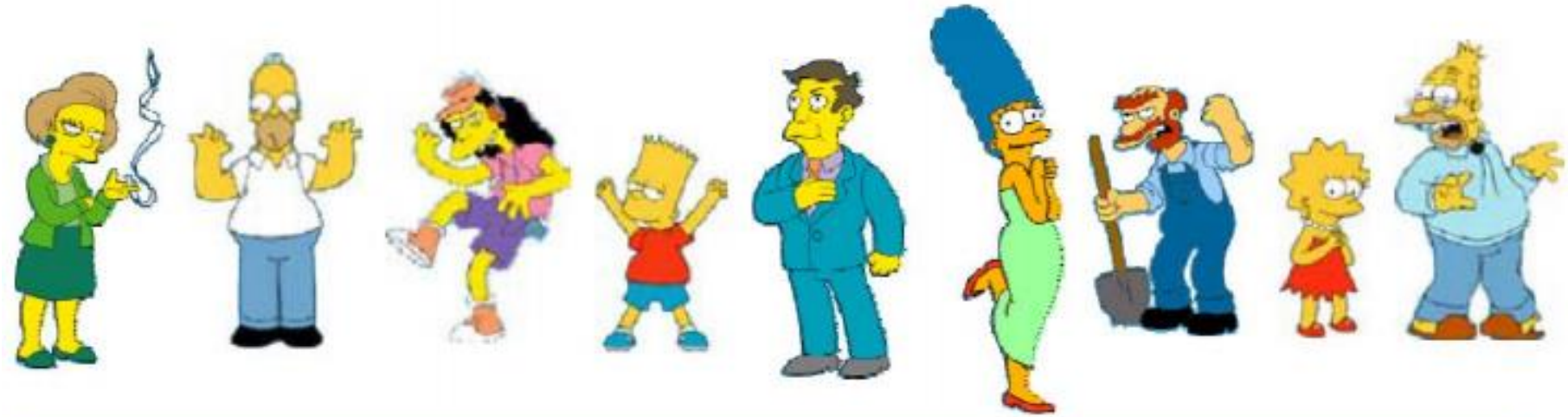
Clustering

- Clustering: the process of grouping a set of objects into classes of similar objects
  - high intra-class similarity
  - low inter-class similarity
  - It is the commonest form of unsupervised learning
- Unsupervised learning: learning from unlabelled data, as opposed to supervised data where a classification of examples is given
- A common and important task that finds many applications in science and engineering
  - Group genes that perform the same function
  - Group individuals that have similar political view
  - Categorize documents of similar topics
  - Identify similar objects from photos

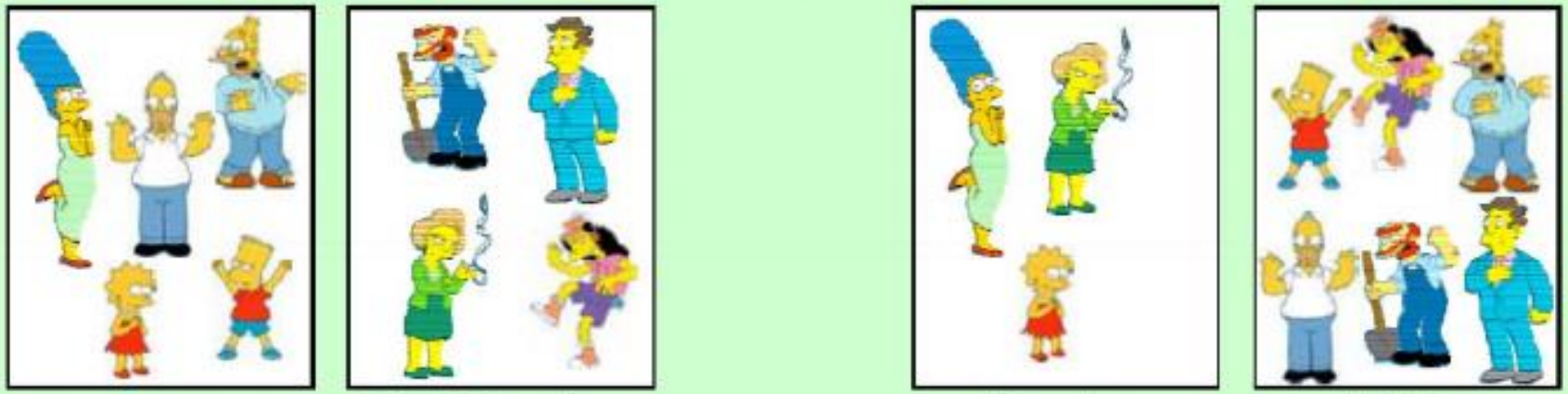
12.2. Issues for clustering

- “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?



Clustering is subjective



Groupness: What is a natural grouping among these objects?

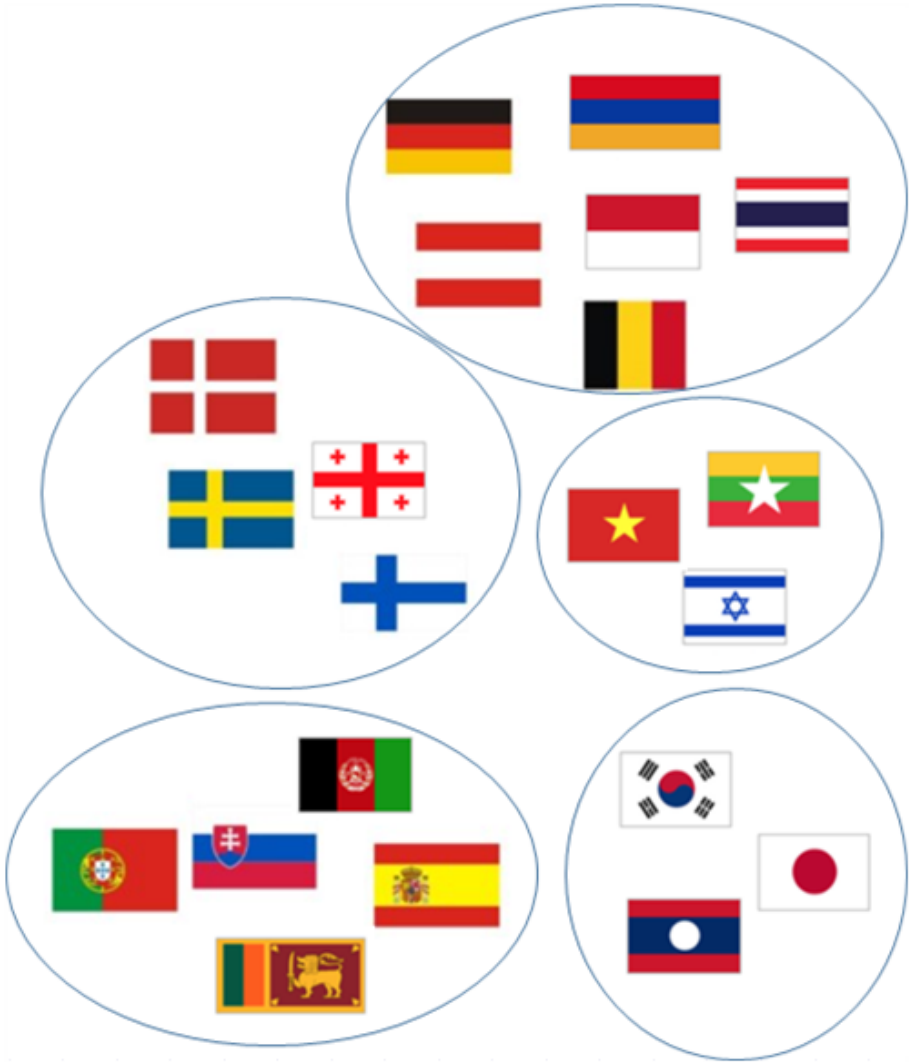


그림 12.2. 국기들의 클러스터

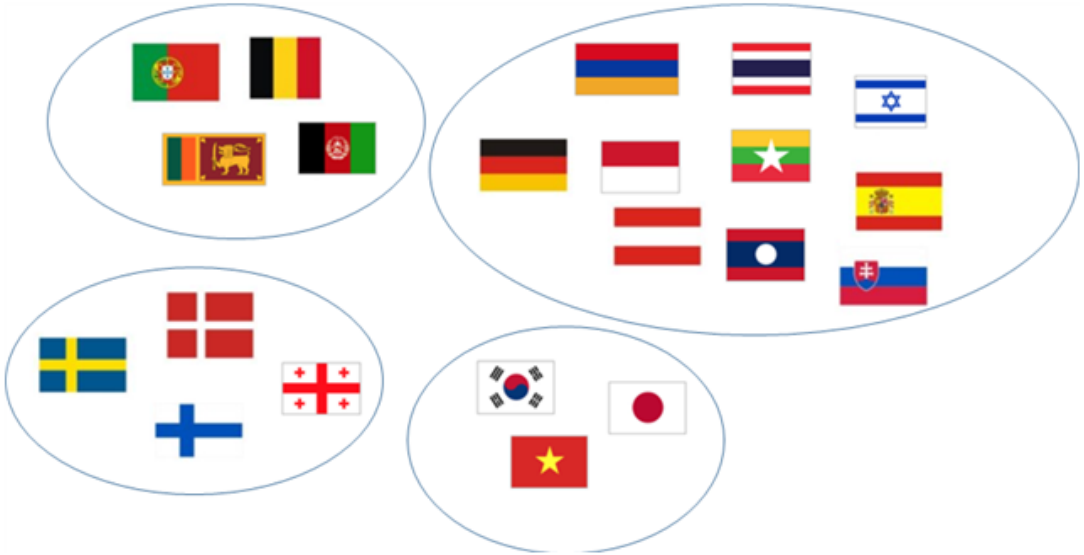


그림 12.3. 국기들의 클러스터2

How do we define similarity?

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

- The real meaning of similarity is a philosophical question. We will take a more pragmatic approach
- Depends on representation and algorithm. For many rep./alg., easier to think in terms of a distance (rather than similarity) between vectors.

What properties should a distance measure have?

- Symmetry
- Self-similarity
- Separation
- Triangular inequality

Distance measures

- Suppose two objects x and y both have p features

$$x = (x_1, x_2, \ldots, x_p) \tag{1}$$

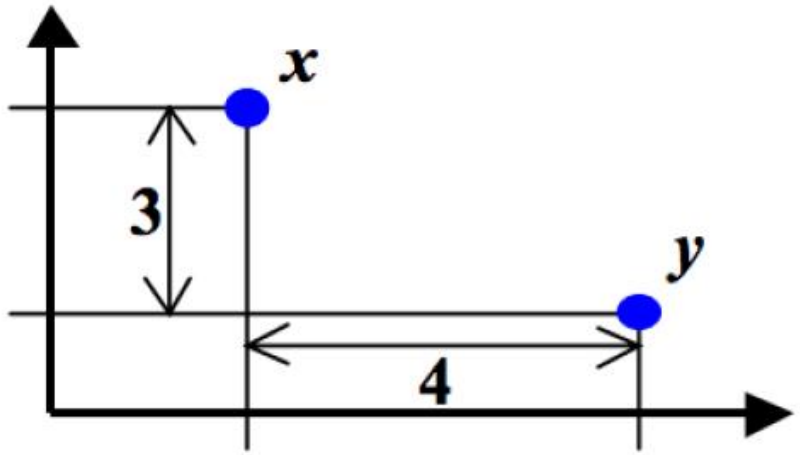
$$y = (y_1, y_2, \ldots, y_p) \tag{2}$$

- The Minkowski metric is defined by

$$d(x, y) = \left(\sum_{i=1}^{p} |x_i - y_i|^r \right)^{1/r} \tag{3}$$

- Common Minkowski metrics
  - Euclidean distance: $r =$?
  - Manhattan distance: $r =$?
  - $r = \infty$ ("sup" distance)

An example



1 : Euclidean distance : $\sqrt[2]{4^2+3^2}=5.$

2 : Manhattan distance : $4+3=7.$

3 : "sup" distance : $\max\{4,3\}=4.$

두 대상 $\boldsymbol{x}$와 $\boldsymbol{y}$의 비슷함을 측정하는 기준으로는 내적(Inner Product)

$$< \boldsymbol{x}, \boldsymbol{y} > = \|\boldsymbol{x}\| \ \|\boldsymbol{y}\| \ \cos(\theta) = \sum_{i=1}^{p} x_i y_i \tag{12.2.7}$$

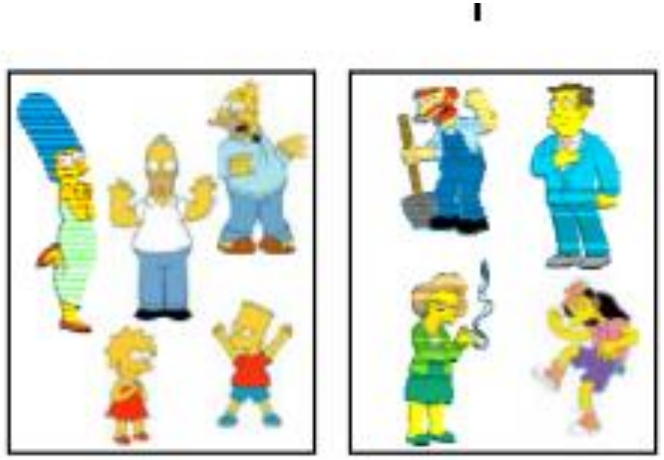
을 생각할 수 있다. 이를 정규화 시킨 Cosine Similarity는

$$\cos(\theta) = \frac{< \boldsymbol{x}, \boldsymbol{y} >}{\|\boldsymbol{x}\| \ \|\boldsymbol{y}\|} = \frac{\sum_{i=1}^{p} x_i y_i}{\sqrt{\sum_{i=1}^{p} x_i^2} \sqrt{\sum_{i=1}^{p} y_i^2}} \tag{12.2.8}$$

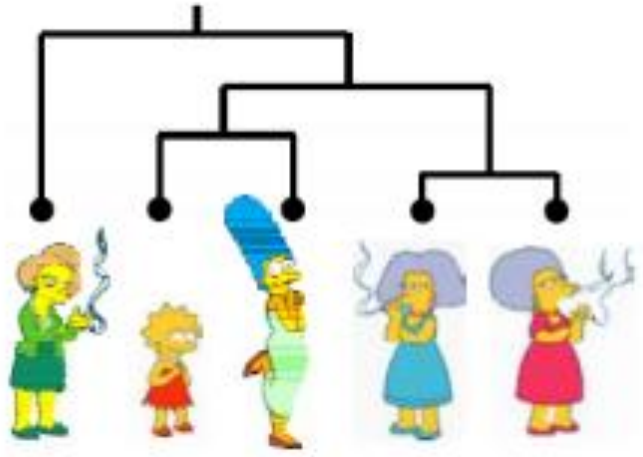
로 정의된다.

Clustering algorithms

- Partitional algorithms
  - Usually start with a random (partial) partitioning
  - Refine it iteratively
    - K means clustering
    - Mixture-Model based clustering



- Hierarchical algorithms
  - Bottom-up, agglomerative
  - Top-down, divisive



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

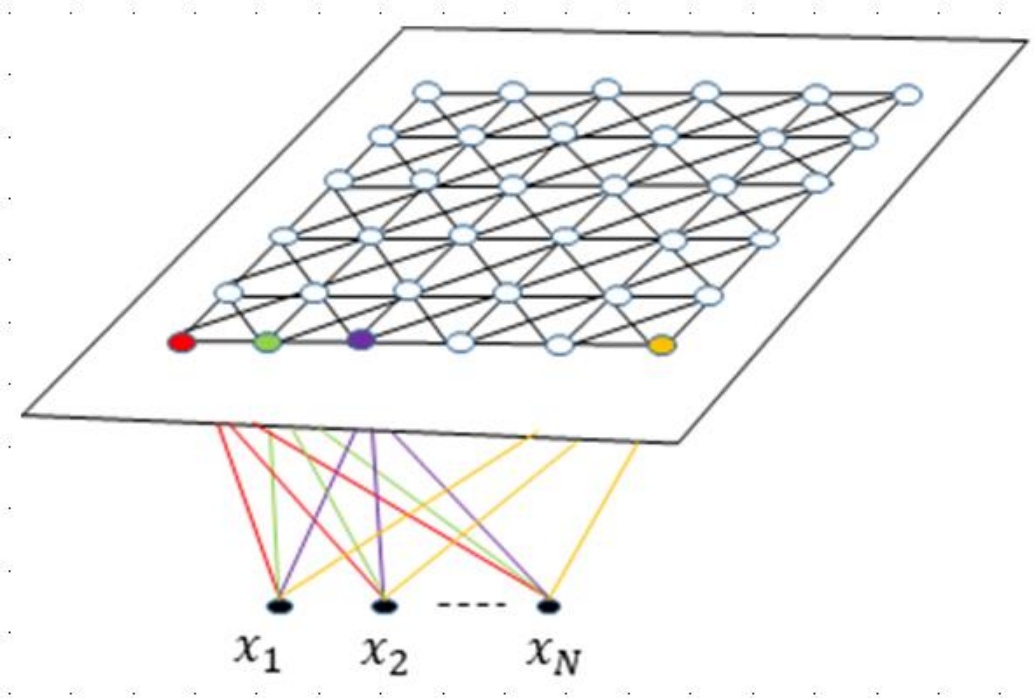
② 한 클러스터를 두 개의 클러스터로 나누는 가능한 방법 중에서 최상의 결과를 택하며

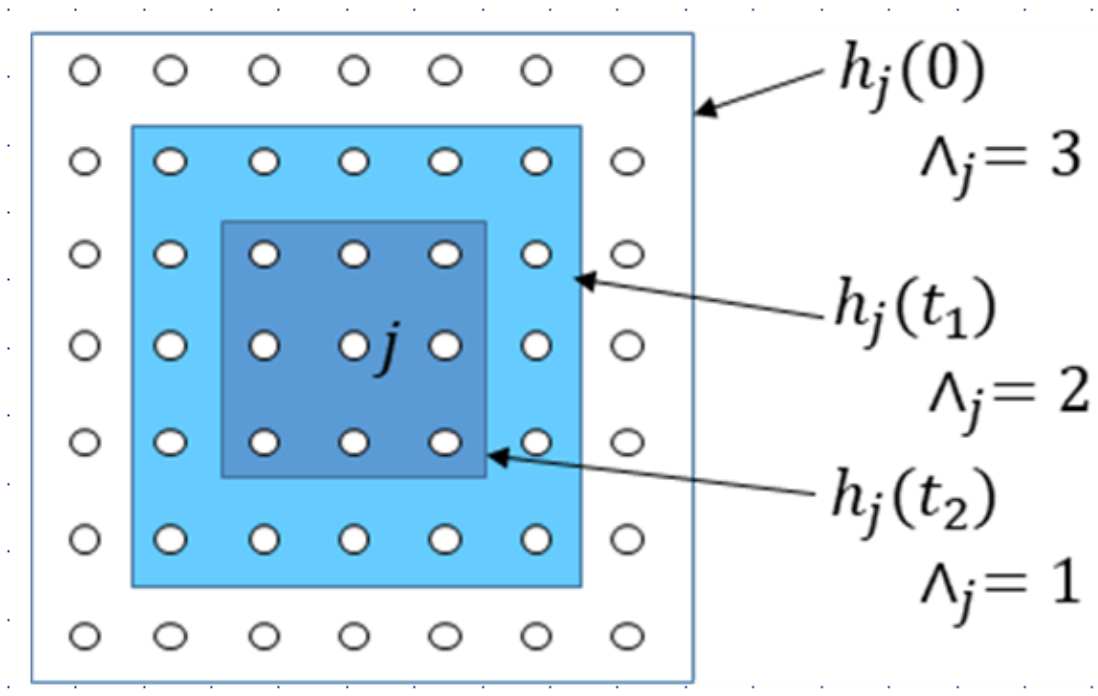
③ 이 과정을 모든 클러스터를 대상으로 반복함.

Distance between clusters

- We can define the distance between two clusters as
  - Single-link nearest neighbor: their closest members
  - Complete-link farthest neighbor: their farthest members
  - Centroid: the centrois (centers of gravity) of the two clusters
  - Average: 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_j를 설정한다.

② 입력 부여

새로운 입력벡터 $\boldsymbol{x}$를 입력노드에 부여한다.

③ 모든 출력노드의 거리 계산

모든 출력노드가 지닌 가중치 벡터와 입력벡터 사이의 거리

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

를 계산한다. 여기서, t는 시간 인덱스이다.

④ **최소거리 출력노드 선정**

모든 출력노드들의 거리 중에서 가장 거리가 작은 출력노드를

$$j^* = \arg\min_j d_j \tag{12.3.2}$$

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

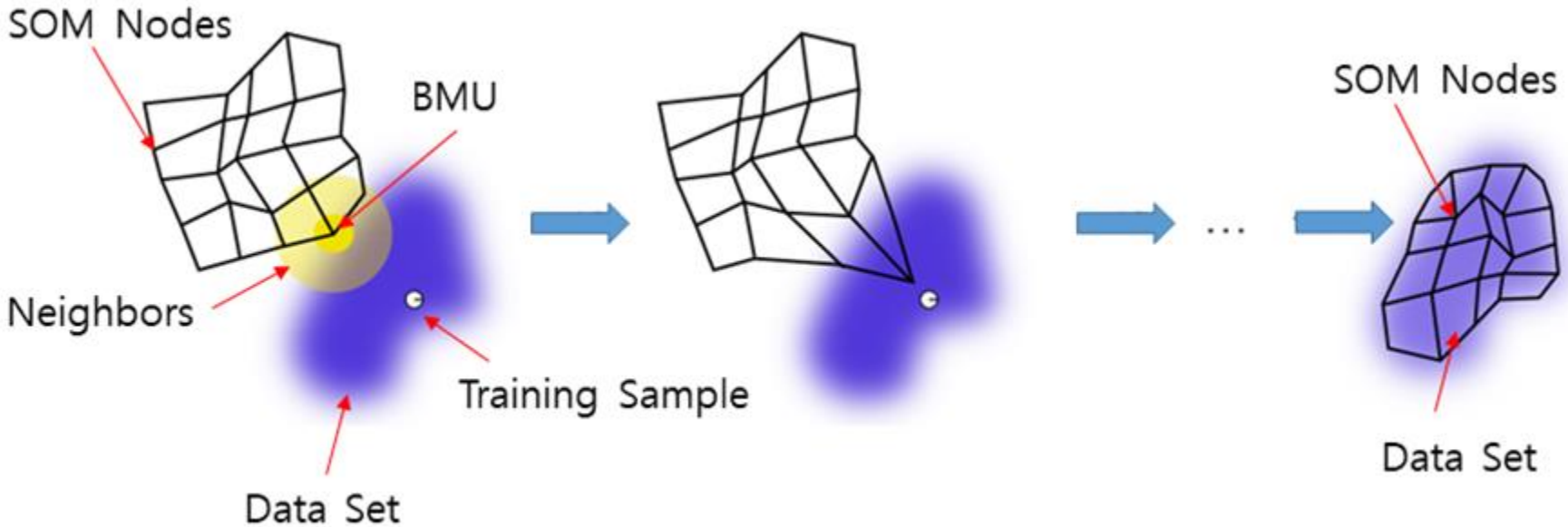
⑤ **BMU와 이웃 노드의 가중치 변경**

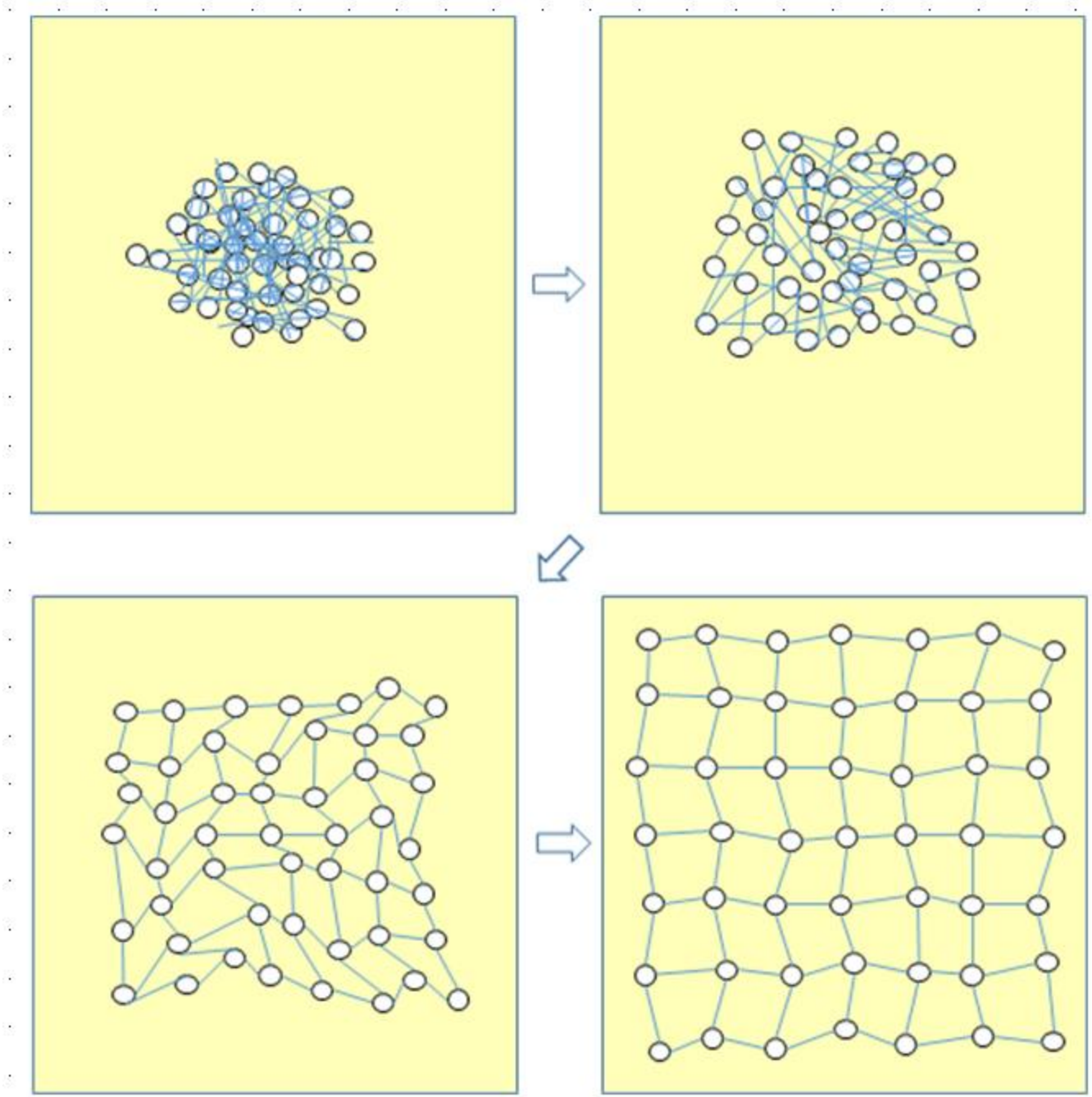
BMU y_{j^*}와 $h_{j^*}(t)$에 의해 결정된 이웃 출력노드들의 가중치를

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

에 따라 변경시킨다. 여기서, $\eta(t)$는 학습률이다.

⑥ ②부터 반복한다.


SOM Nodes
BMU
Neighbors
Training Sample
Data Set
...
SOM Nodes
Data Set



12.4. Conscience Learning

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

$$p_j(t+1) = p_j(t) + B(y_j(t) - p_j(t)) \tag{12.4.2}$$

$$b_j = C(1/M - p_j) \tag{12.4.3}$$

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

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

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

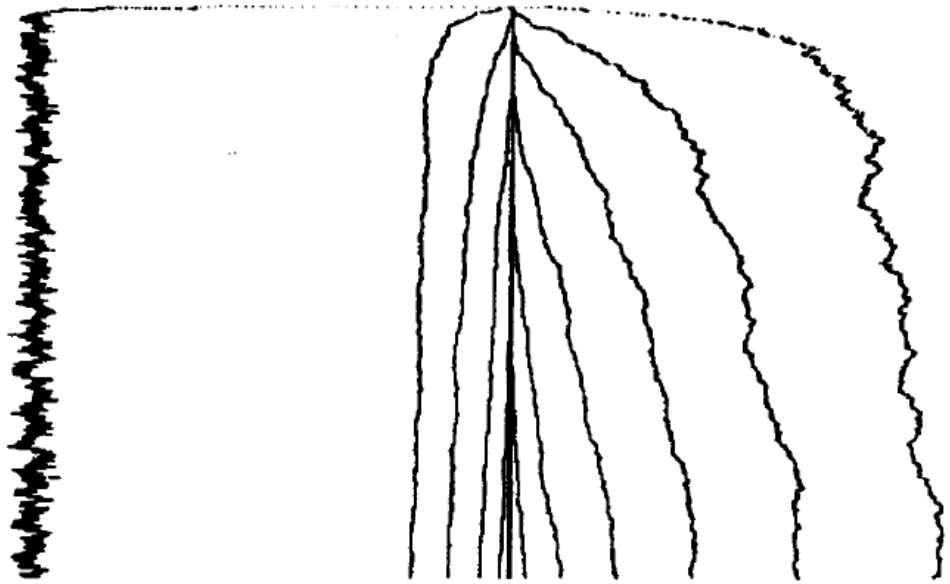


Figure 9. Kohonen learning. A = 0.03 for 16000 iterations.

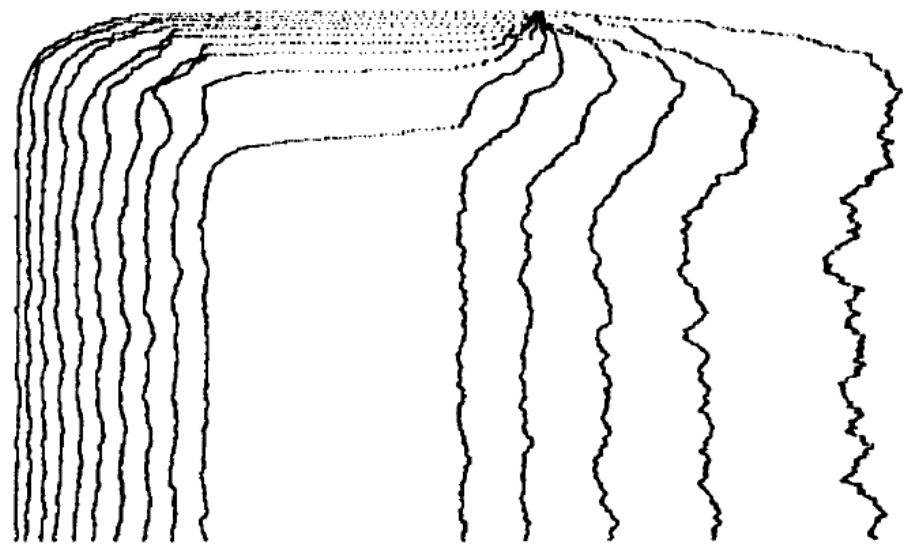


Figure 10. Conscience learning. A = 0.03 for 16000 iterations.

12.5. K-means Clustering

☐ 24 bits/pixel (16M colors) ⇒ 8bits/pixel (256 colors)

- We could quantize uniformly (e.g., color [0, 65535] → color 0), but wastes the color map
- Find k reference vectors (prototypes/codebook vectors/codewords) which best represent the data

☐ Given $\mathcal{X} = \{\mathbf{x}^t\}$, find k reference vectors: $\mathbf{m}_j, j = 1, \ldots, k$

- In color quantization example, $\mathcal{X}$ is the colors found in the image, and $\mathbf{m}_j, j = 1, \ldots, 256$ are the color map entries

☐ Will be using nearest (most similar) reference:

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

☐ Reconstruction error defined as:

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

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

☐ Iterative algorithm:

Initialize $\boldsymbol{m}_i, i = 1, \ldots, k$, for example, to k random $\boldsymbol{x}^t$

Repeat

For all $\boldsymbol{x}^t \in \mathcal{X}$

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

For all $\boldsymbol{m}_i, i = 1, \ldots, k$

$$\boldsymbol{m}_i \leftarrow \sum_t b_i^t \boldsymbol{x}^t / \sum_t b_i^t$$

Until $\boldsymbol{m}_i$ converge

Estimate the labels (determine which m_i to use)

∂ Err/∂ m = 0

☐ Assuming Euclidean distance and fixed b_i^t:

- $Err(\{\mathbf{m}_i\}|\mathcal{X}) = \sum_t \sum_i b_i^t \|\mathbf{x}^t - \mathbf{m}_i\|^2 = \sum_t \sum_i b_i^t \sum_j (x_j^t - m_{ij})^2$
- ∂ Err / ∂ m_{ij} = 0 yields: $m_{ij} = \sum_t b_i^t x_j^t / \sum_t b_i^t$

$$b_i^t \leftarrow \begin{cases} 1 & \text{if } \|\boldsymbol{x}^t - \boldsymbol{m}_i\| = \min_j \|\boldsymbol{x}^t - \boldsymbol{m}_j\| \\ 0 & \text{otherwise} \end{cases} \tag{12.5.4}$$

k-평균 알고리즘

① 파라미터 설정

k 값을 결정한다.

$$\boldsymbol{m}_i \leftarrow \frac{\sum_t b_i^t \boldsymbol{x}^t}{\sum_i b_i^t} \tag{12.5.6}$$

② 초기화

k 클러스터의 중심 기준벡터 $\boldsymbol{m}_j (j = 1,2,...,k)$를 임의로 초기화 한다.

③ 할당 단계

모든 $\boldsymbol{x}^t \in X$에 대하여 b_i^t를 식 (12.5.4)에 따라 할당한다.

④ 갱신 단계

모든 $\boldsymbol{m}_i (i = 1,2,...,k)$에 대하여 식 (12.5.6)에 따라 $\boldsymbol{m}_i$를 갱신한다.

⑤ 종료/갱신

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

참조: 오차함수 최적화와 $\boldsymbol{m}_i$ 갱신

식 (12.5.6)과 같이 $\boldsymbol{m}_i$를 변경하는 것이 식 (12.5.2)로 주어진 오차함수를 최적화 시키는 지에 대하여 알아보자. 식 (12.5.2)로 주어진 오차함수를 다시 적으면

$$E(\{\boldsymbol{m}_i\}|X) = \sum_t \sum_i b_i^t \|\boldsymbol{x}^t - \boldsymbol{m}_i\|^2 = \sum_t \sum_i b_i^t \sum_j (x_j^t - m_{ij})^2 \qquad (12.5.7)$$

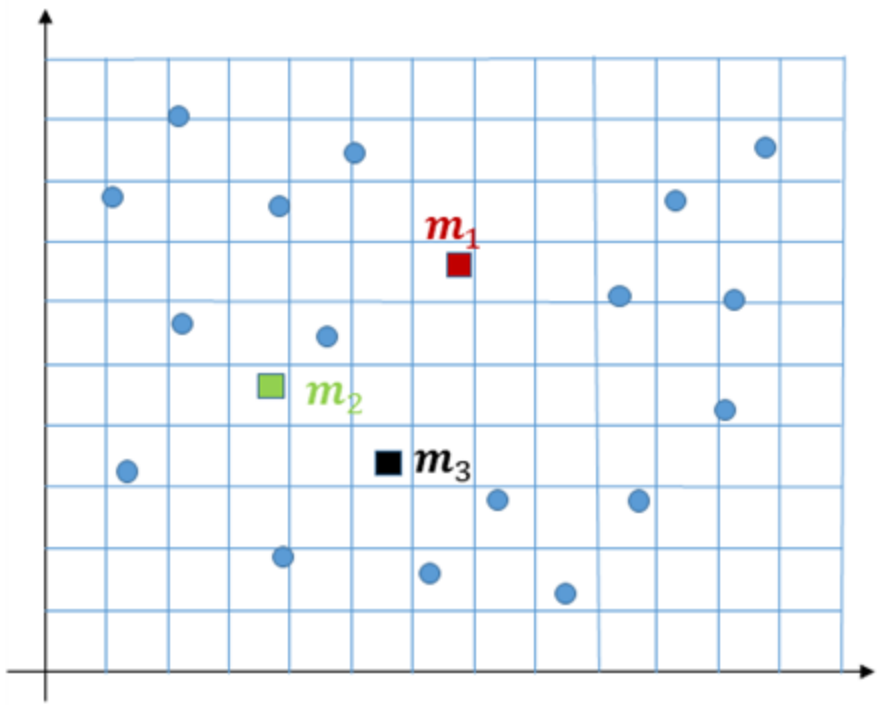
이 된다. 위 식을 m_{ij}에 대하여 편미분을 하면

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

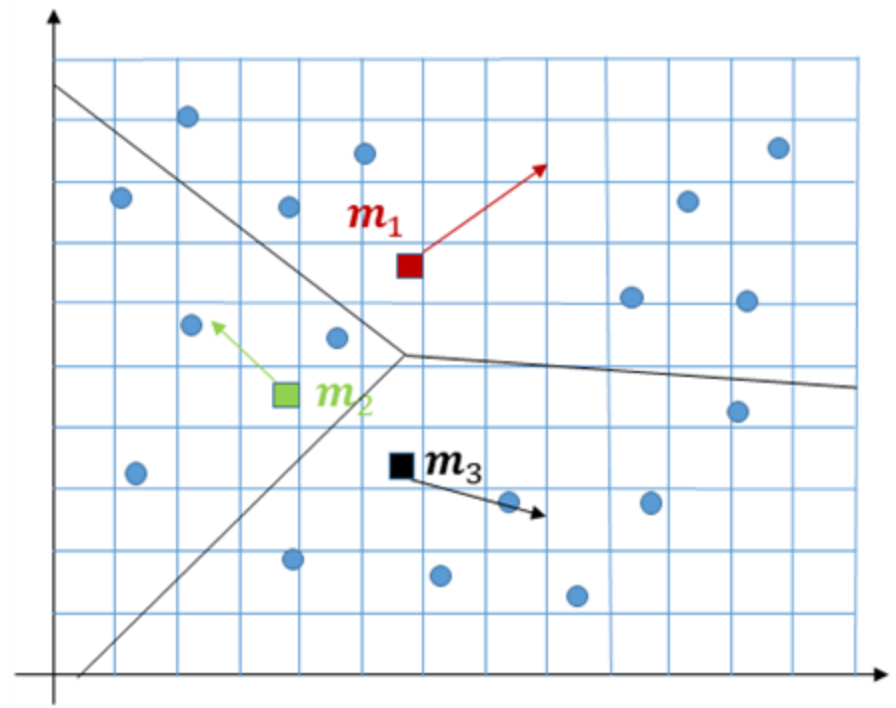
이 된다. 이 편미분이 0이 되는 조건에 의해

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

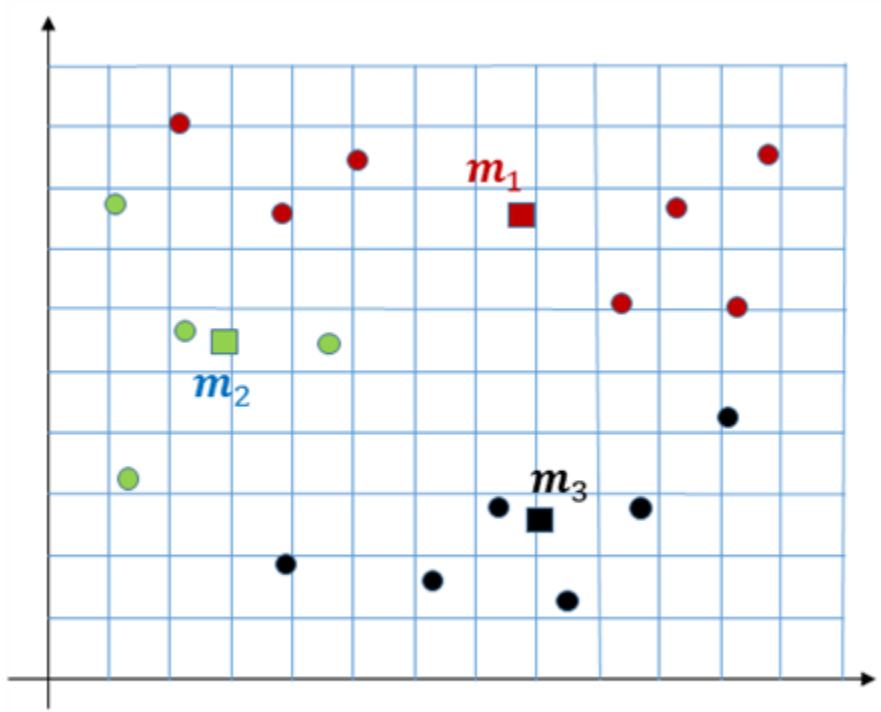
을 얻게 되며, 이를 벡터로 표현하면 식 (12.5.6)이 된다.



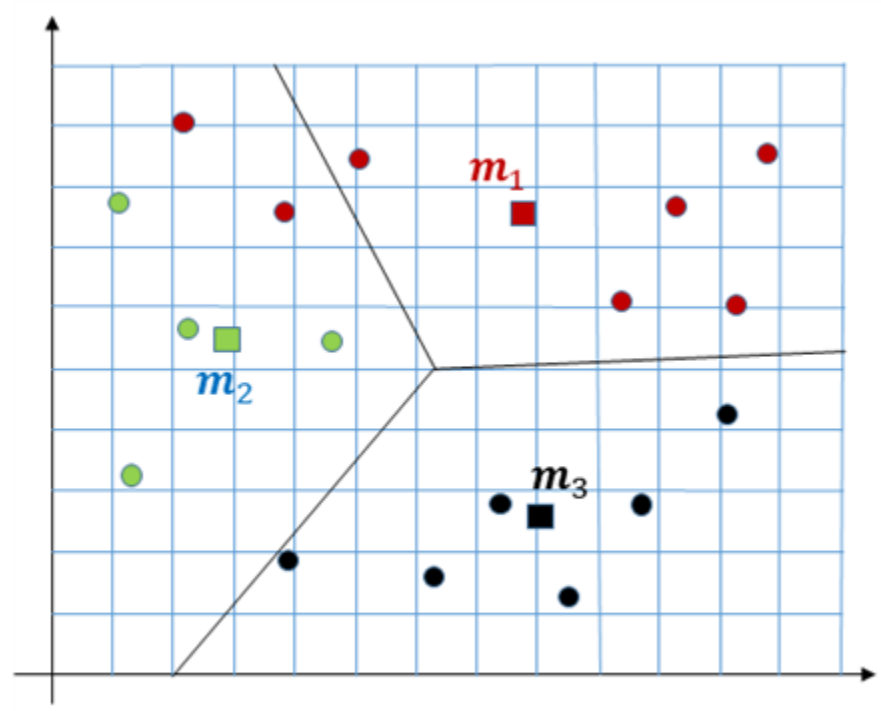
①Initialization of Centroids



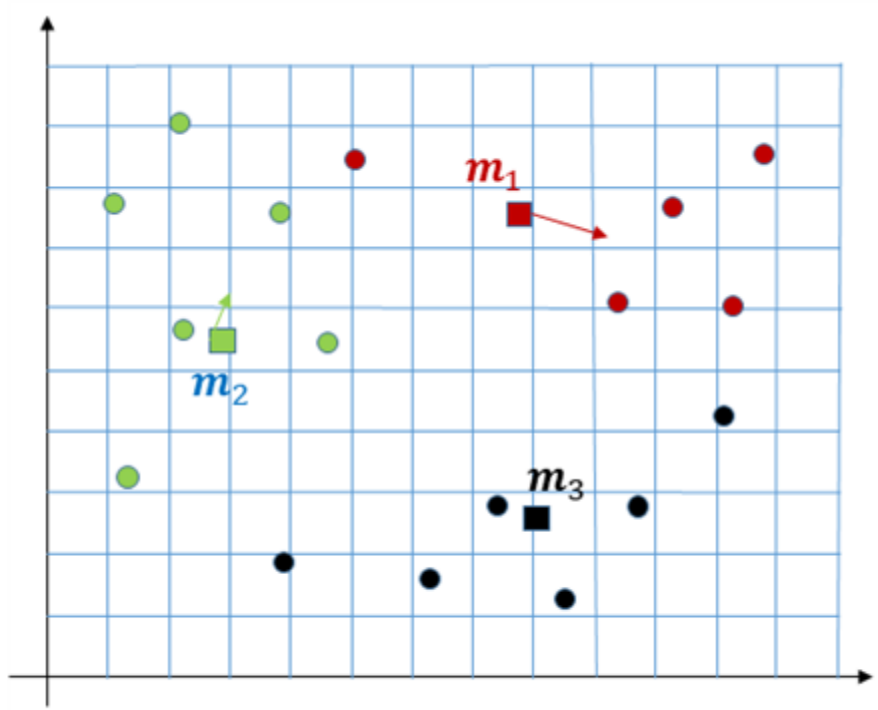
②Membership Assignment



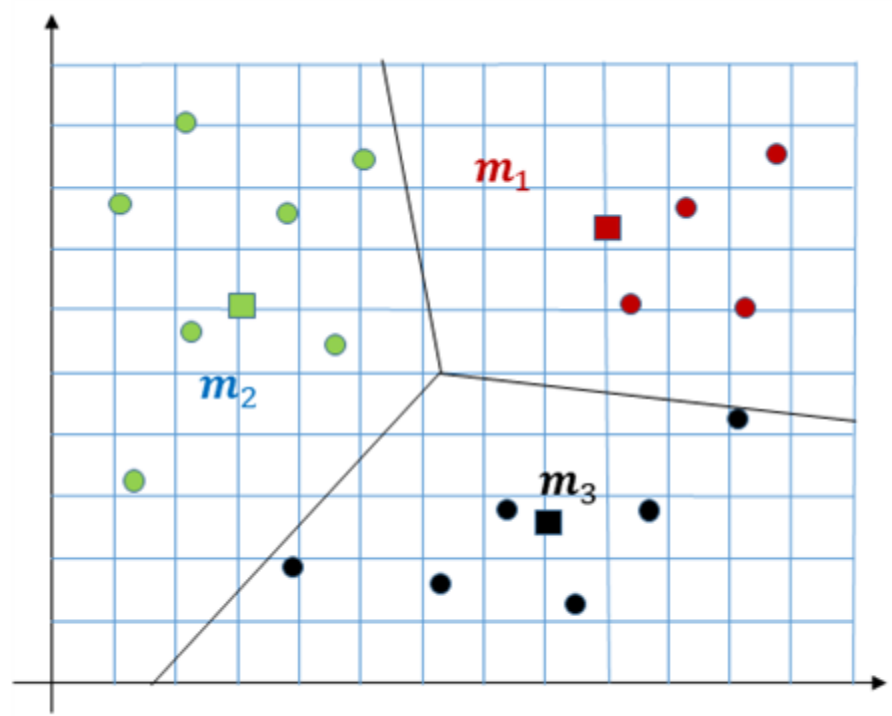
③Updating Centroids



④Membership Assignment



⑤Updating Centroids



⑥Membership Assignment

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

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

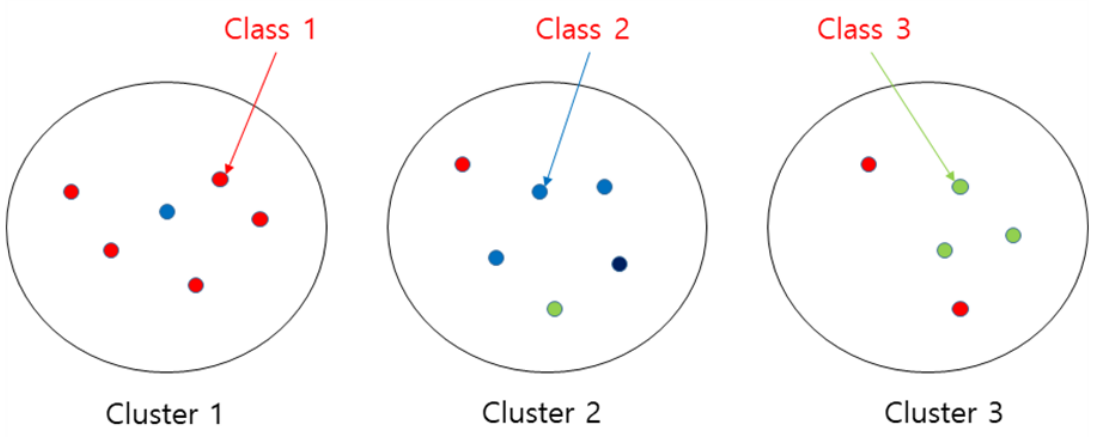
Dunn index

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

External evaluation

Purity

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



$$Purity(S_1) = \frac{1}{6} max(5,1,0) = \frac{5}{6}$$

$$Purity(S_2) = \frac{1}{6} max(1,4,1) = \frac{4}{6}$$

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