연세대학교 통계 데이터 사이언스 학회 ESC 23-2 FALL WEEK6

Cluster Analysis

[ESC 정규세션 학술부] 김민주 오동윤

Contents

Part I.

Part II.

1. Introduction

Introduction

Cluster Analysis

-Proximity가 높은 object끼리 cluster로 묶는 다변량 기법

과정

1. Choose proximity measure

↓

2. Choose group-building algorithm

2.The proximity between objects

The Proximity Between Objects

- Data matrix $\mathcal{X}_{n\times p}$

- Proximity matrix(or dissimilarity matrix) $\mathcal{D}_{n\times n}$

$$\mathcal{D} = \begin{pmatrix} d_{11} & d_{12} & \dots\dots\dots & d_{1n} \\ \vdots & d_{22} & & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \dots\dots\dots & d_{nn} \end{pmatrix}.$$

where d_{ij} : dissimilarity measure(or proximity measure) Ex) L2-norm

The Proximity Between Objects

Similarity of Objects with Binary Structure

- Euclidean distance를 사용할 경우 x_{ik}가 0인 경우와 1인 경우를 동일하게 취급하므로, proximity measure를 사용

$$d_{ij} = \frac{a_1 + \delta a_4}{a_1 + \delta a_4 + \lambda(a_2 + a_3)}$$

where

$$a_1 = \sum_{k=1}^{p} \mathrm{I}(x_{ik} = x_{jk} = 1), \quad a_3 = \sum_{k=1}^{p} \mathrm{I}(x_{ik} = 1, x_{jk} = 0),$$

$$a_2 = \sum_{k=1}^{p} \mathrm{I}(x_{ik} = 0, x_{jk} = 1), \quad a_4 = \sum_{k=1}^{p} \mathrm{I}(x_{ik} = x_{jk} = 0).$$

Ex 13.1) Car Marks Data

X_1: A Economy,
X_2:B Service,
X_3: C Non-depreciation of value,
X_4: D Price, Mark 1 for very cheap cars
X_5: E Design,
X_6: F Sporty car,
X_7: G Safety, and
X_8: H Easy handling.

$X_i \in \{1,2,3,4,5,6\}$

$$y_{ik} = \begin{cases} 1 & \text{if } x_{ik} > \overline{x}_k, \\ 0 & \text{otherwise,} \end{cases}$$

$i = 1, \dots n, k - 1, \dots p$

Jacard($\delta = 0,\ \lambda = 1$)

$$\mathcal{D} = \begin{pmatrix} 1.000 & 0.000 & 0.400 \\ & 1.000 & 0.167 \\ & & 1.000 \end{pmatrix}$$

Tanimoto($\delta = 1,\ \lambda = 2$)

$$\mathcal{D} = \begin{pmatrix} 1.000 & 0.000 & 0.455 \\ & 1.000 & 0.231 \\ & & 1.000 \end{pmatrix}$$

Simple Matching($\delta = 1,\ \lambda = 1$)

$$\mathcal{D} = \begin{pmatrix} 1.000 & 0.000 & 0.625 \\ & 1.000 & 0.375 \\ & & 1.000 \end{pmatrix}$$

The Proximity Between Objects

Distance Measures for Continuous Variables

- Distance Measure: $L_r - norms$

$$d_{ij} = ||x_i - x_j||_r = \left\{ \sum_{k=1}^{p} |x_{ik} - x_{jk}|^r \right\}^{1/r}$$

$$d_{ij}^2 = \|x_i - x_j\|_{\mathcal{A}} = (x_i - x_j)^\top \mathcal{A}(x_i - x_j) \qquad d_{ij}^2 = \sum_{k=1}^{p} \frac{(x_{ik} - x_{jk})^2}{s_{X_k X_k}}$$

Ex 13.2) $x_1 = (0,0), x_2 = (1,0), x_3 = (5,5)$

-L1 norm

$$\mathcal{D}_1 = \begin{pmatrix} 0 & 1 & 10 \\ 1 & 0 & 9 \\ 10 & 9 & 0 \end{pmatrix}$$

-squared L2 norm

$$\mathcal{D}_2 = \begin{pmatrix} 0 & 1 & 50 \\ 1 & 0 & 41 \\ 50 & 41 & 0 \end{pmatrix}$$

-Contingency table χ에 대해, 각 행과 열은 $\frac{x_{ij}}{x_{i.}}$ 의conditional frequency distribution

- χ^2-metric: $$d^2(i_1, i_2) = \sum_{j=1}^{p} \frac{1}{\left(\frac{x_{\bullet j}}{x_{\bullet\bullet}}\right)} \left(\frac{x_{i_1 j}}{x_{i_1 \bullet}} - \frac{x_{i_2 j}}{x_{i_2 \bullet}} \right)^2$$

3.Cluster Algorithm

Cluster Algorithm

Traditional Clustering method

1. Non-hierarchical algorithm	vs	2. Hierarchical algorithm
-iteration에 따라 object의 그룹이 바뀜		-group이 정해지면 바뀌지 않음
-Data의 저장이 필요 없어 큰 data set에 적용가능		-비교적 큰 data set에 적용 불가능

Cluster Algorithm

Partitioning(nonhierarchical clustering) Algorithm

-Goal: 정해진 k에 대해 distance based objective function을 minimize

K-means Method

$$\hat{\mathcal{S}} = \underset{\mathcal{S}}{\operatorname{argmin}} \sum_{j=1}^{k} \sum_{i \in \mathcal{S}_j} \|x_i - \mu_j\|^2 \qquad S = \{S_1, \ldots, S_k\}$$

1. Initial partition set을 지정
2. Each object와 group centroid와의 거리를 계산하여 nearest group에 reassign
3. Repeat until convergence

Cluster Algorithm

K-means Method

Ex)

Item	Observations x_1	x_2
A	5	3
B	−1	1
C	1	−2
D	−3	−2

1. Initial Set: (AB) (CD)

Cluster	Coordinates of centroid $\bar{x}_1$	$\bar{x}_2$
(AB)	$\frac{5+(-1)}{2}=2$	$\frac{3+1}{2}=2$
(CD)	$\frac{1+(-3)}{2}=-1$	$\frac{-2+(-2)}{2}=-2$

2. Compute the distance

$$d^2(A,(AB)) = (5-2)^2 + (3-2)^2 = 10$$
$$d^2(A,(CD)) = (5+1)^2 + (3+2)^2 = 61$$
$$d^2(A,(B)) = (5+1)^2 + (3-1)^2 = 40$$
$$d^2(A,(ACD)) = (5-1)^2 + (3+.33)^2 = 27.09$$

$$d^2(B,(AB)) = (-1-2)^2 + (1-2)^2 = 10$$
$$d^2(B,(CD)) = (-1+1)^2 + (1+2)^2 = 9$$
$$d^2(B,(A))) = (-1-5)^2 + (1-3)^2 = 40$$
$$d^2(B,(BCD)) = (-1+1)^2 + (1+1)^2 = 4$$

$$d^2(C,(A)) = (1-5)^2 + (-2-3)^2 = 41$$
$$d^2(C,(BCD)) = (1+1)^2 + (-2+1)^2 = 5$$
$$d^2(C,(AC)) = (1-3)^2 + (-2-.5)^2 = 10.25$$
$$d^2(C,(BD)) = (1+2)^2 + (-2+.5)^2 = 11.25$$

Cluster A: 0
Cluster (BCD): $4+5+5=14$

$$\min E = \sum d^2_{i,\,c(i)}$$

- update set: (A) (BCD) => converge

- Successive computation 사용

$$\bar{x}_{i,\,new} = \frac{n\bar{x}_i + x_{ji}}{n+1} \quad \text{if the jth item is } \textit{added} \text{ to a group}$$

$$\bar{x}_{i,\,new} = \frac{n\bar{x}_i - x_{ji}}{n-1} \quad \text{if the jth item is } \textit{removed} \text{ from a group}$$

Cluster Algorithm

K-means Method

Ex) Cluster 개수 K를 설정하는 기준 예시: Table12.4의 Public Utility Data 22개

maximize the between-cluster variability relative to the within-cluster variability

$K = 4$

Cluster	Number of firms	Firms
1	5	Idaho Power Co. (8), Nevada Power Co. (11), Puget Sound Power & Light Co. (16), Virginia Electric & Power Co. (22), Kentucky Utilities Co. (9).
2	6	Central Louisiana Electric Co. (3), Oklahoma Gas & Electric Co. (14), The Southern Co. (18), Texas Utilities Co. (19), Arizona Public Service (1), Florida Power & Light Co. (6).
3	5	New England Electric Co. (12), Pacific Gas & Electric Co. (15), San Diego Gas & Electric Co. (17), United Illuminating Co. (21), Hawaiian Electric Co. (7).
4	6	Consolidated Edison Co. (N.Y.) (5), Boston Edison Co. (2), Madison Gas & Electric Co. (10), Northern States Power Co. (13), Wisconsin Electric Power Co. (20), Commonwealth Edison Co. (4).

Distances between Cluster Centers

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}
\begin{array}{c}
\begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\
\begin{bmatrix} 0 & & & \\ 3.08 & 0 & & \\ 3.29 & 3.56 & 0 & \\ 3.05 & 2.84 & 3.18 & 0 \end{bmatrix}
\end{array}$$

$K = 5$

Cluster	Number of firms	Firms
1	5	Nevada Power Co. (11), Puget Sound Power & Light Co. (16), Idaho Power Co. (8), Virginia Electric & Power Co. (22), Kentucky Utilities Co. (9).
2	6	Central Louisiana Electric Co. (3), Texas Utilities Co. (19), Oklahoma Gas & Electric Co. (14), The Southern Co. (18), Arizona Public Service (1), Florida Power & Light Co. (6).
3	5	New England Electric Co. (12), Pacific Gas & Electric Co. (15), San Diego Gas & Electric Co. (17), United Illuminating Co. (21), Hawaiian Electric Co. (7).
4	2	Consolidated Edison Co. (N.Y.) (5), Boston Edison Co. (2).
5	4	Commonwealth Edison Co. (4), Madison Gas & Electric Co. (10), Northern States Power Co. (13), Wisconsin Electric Power Co. (20).

Distances between Cluster Centers

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}
\begin{array}{c}
\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{bmatrix} 0 & & & & \\ 3.08 & 0 & & & \\ 3.29 & 3.56 & 0 & & \\ 3.63 & 3.46 & 2.63 & 0 & \\ 3.18 & 2.99 & 3.81 & 2.89 & 0 \end{bmatrix}
\end{array}$$

MANOVA Table for Comparing Population Mean Vectors

Source of variation	Matrix of sum of squares and cross products (SSP)	Degrees of freedom (d.f.)
Treatment	$\mathbf{B} = \sum_{\ell=1}^{g} n_\ell(\bar{\mathbf{x}}_\ell - \bar{\mathbf{x}})(\bar{\mathbf{x}}_\ell - \bar{\mathbf{x}})'$	$g - 1$
Residual (Error)	$\mathbf{W} = \sum_{\ell=1}^{g} \sum_{j=1}^{n_\ell} (\mathbf{x}_{\ell j} - \bar{\mathbf{x}}_\ell)(\mathbf{x}_{\ell j} - \bar{\mathbf{x}}_\ell)'$	$\sum_{\ell=1}^{g} n_\ell - g$
Total (corrected for the mean)	$\mathbf{B} + \mathbf{W} = \sum_{\ell=1}^{g} \sum_{j=1}^{n_\ell} (\mathbf{x}_{\ell j} - \bar{\mathbf{x}})(\mathbf{x}_{\ell j} - \bar{\mathbf{x}})'$	$\sum_{\ell=1}^{g} n_\ell - 1$

$$F_{\text{nuc}} = \frac{\text{mean square percent nuclear between clusters}}{\text{mean square percent nuclear within clusters}} = \frac{3.335}{.255} = 13.1$$

$\frac{|W|}{|B+W|}$, $tr(W^{-1}B)$ 등을 기준으로 사용

Cluster Algorithm

K-means Method

장점

1. Simple and easy
2. Fast: Computational cost $O(tkn) \approx O(n)$
3. Scalability
4. Flexibility

단점

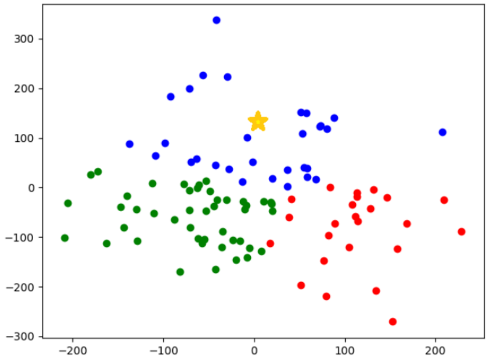
1. Sensitive to initial set
2. Sensitive to outlier

→ local minimum에 도달할 수도

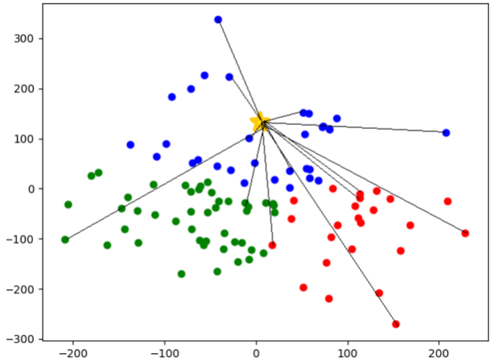
Cluster Algorithm

K-means++ Method

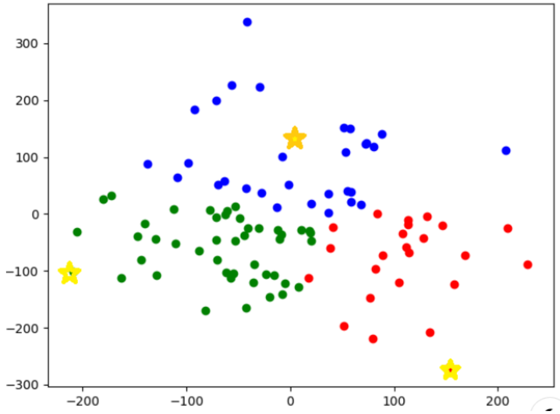
1. K개의 centroid를 initialize하지 않고,
   1개의 point를 centroid로 지정



2. Centroid부터 나머지 point까지의 거리 계산



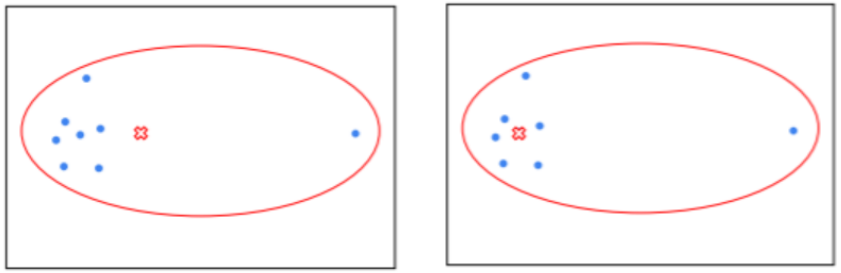
3. Centroid로부터 가장 먼 곳 data point를
   centroid로 지정해 k개 initial centroid



Cluster Algorithm

K-medoids Method

-K-mean Method의 outlier에 민감함을 보완

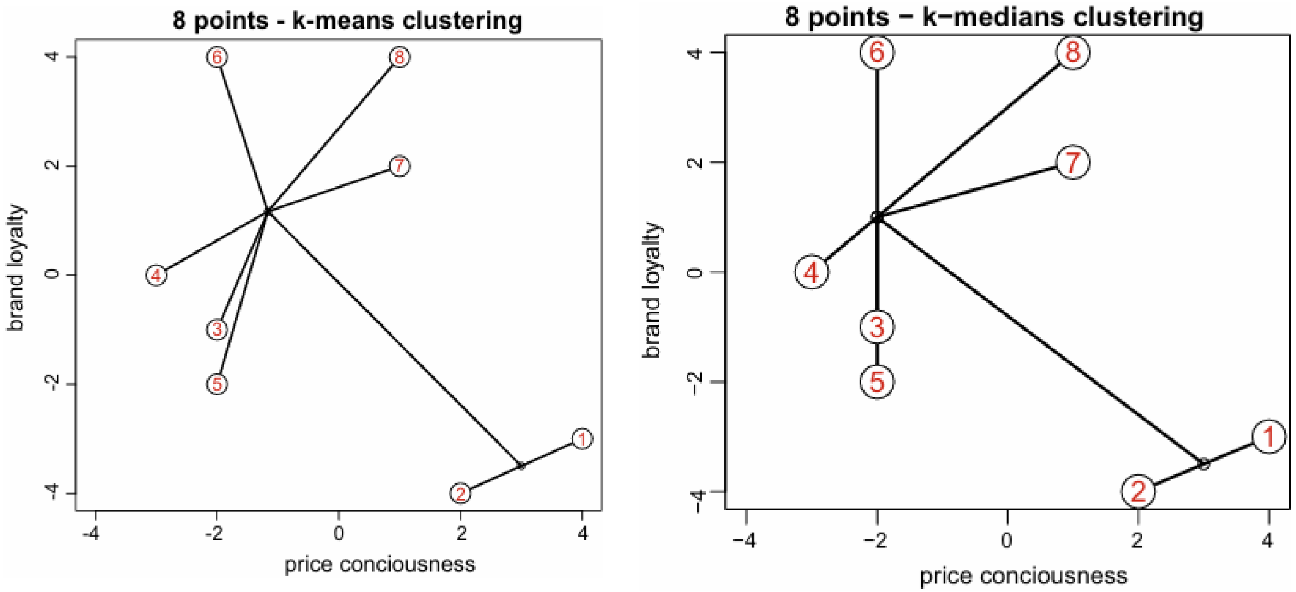


단점

-느림: Computational cost $O(k*(n-k)^2)$

K-median Method

$$\hat{\mathcal{S}} = \underset{\mathcal{S}}{\operatorname{argmin}} \sum_{j=1}^{k} \sum_{i \in \mathcal{S}_j} |x_i - med_j|$$



Cluster Algorithm

Fuzzy k-means Method

$$\hat{\mathcal{S}} = \underset{\mathcal{S}}{\operatorname{argmin}} \sum_{j=1}^{k} \sum_{i \in \mathcal{S}_j} u_{i,j} \|x_i - \mu_j\|^2$$

-각 data point가 특정 cluster에 속할 가능성을 weight로

- w_{ij}: object i가 cluster j에 속할 확률

Problem

$min_{\mathcal{S}} \sum_{j=1}^{k} \sum_{i=1}^{n} w_{ij}{}^{p} \|x_i - \mu_j\|$

Subject to $\sum_{j=1}^{k} w_{ij}$, $0 < \sum_{i=1}^{n} w_{ik} < n$

-$\hat{\mathcal{S}} = argmin_{\mathcal{S}} \sum_{j=1}^{k} \sum_{i=1}^{n} w_{ij}{}^{p} d(x_i, \mu_j)^2$

-$\mu_j = \frac{\sum_{i=1}^{n} w^p{}_{ik} x_i}{\sum_{i=1}^{n} w^p{}_{ik}}$, j=1,⋯,K

-$w_{ik} = \frac{\left\{\frac{1}{d(x_i,\mu_k)^2}\right\}^{\frac{1}{p-1}}}{\sum_{j=1}^{K}\left\{\frac{1}{d(x_i,\mu_j)^2}\right\}^{\frac{1}{p-1}}}$, j=1,⋯,k

-p가 커질수록 fuzzy해지므로 일반적으로 p=2 사용

-$w_{ik} = \frac{1}{\sum_{j=1}^{K}\left\{\frac{d(x_i,\mu_k)^2}{d(x_i,\mu_j)^2}\right\}}$, j=1,⋯,k

Cluster Algorithm

Hierarchical Algorithm

- Agglomerative algorithm
- Splitting algorithm

Agglomerative Algorithm

1. N개의 cluster로 초기값 설정, $\mathcal{D}_{n\times n} = \{d_{ik}\}$
2. 가장 가까운 두 개의 cluster를 하나로 병합
3. $\mathcal{D}_{(n-1)\times(n-1)} = \{d_{ik}\}$ 업데이트
4. 2-3을 n-1번 반복

$$d_{(UV)W} = \delta_1 d_{UW} + \delta_2 d_{VW} + \delta_3 d_{UV} + \delta_4 |d_{UW} - d_{VW}|$$

Cluster Algorithm

Single Linkage(Nearest Neighbor algorithm)

$d_{(UV)W} = \delta_1 d_{UW} + \delta_2 d_{VW} + \delta_3 d_{UV} + \delta_4 |d_{UW} - d_{VW}|$ where $\delta_1 = \frac{1}{2},\ \delta_2 = \frac{1}{2}, \delta_3 = 0,\ \delta_4 = -\frac{1}{2},$

$d_{(UV)W} = \min\{d_{UW}, d_{VW}\}$

Ex)

$$\mathbf{D} = \{d_{ik}\} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & & \\ 9 & 0 & & & \\ 3 & 7 & 0 & & \\ 6 & 5 & 9 & 0 & \\ 11 & 10 & 2 & 8 & 0 \end{bmatrix} \end{array}$$

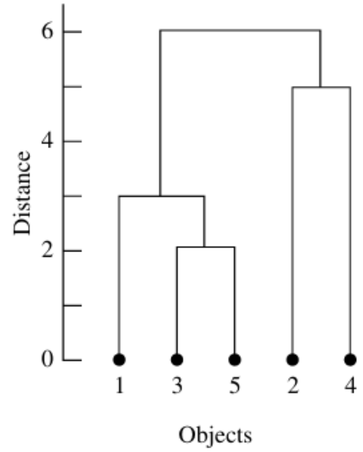
$$\min_{i,k}(d_{ik}) = d_{53} = 2$$

$$d_{(35)1} = \min\{d_{31}, d_{51}\} = \min\{3, 11\} = 3$$
$$d_{(35)2} = \min\{d_{32}, d_{52}\} = \min\{7, 10\} = 7$$
$$d_{(35)4} = \min\{d_{34}, d_{54}\} = \min\{9, 8\} = 8$$

$$\begin{array}{c} \\ (35) \\ 1 \\ 2 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} (35) & 1 & 2 & 4 \end{array} \\ \begin{bmatrix} 0 & & & \\ 3 & 0 & & \\ 7 & 9 & 0 & \\ 8 & 6 & 5 & 0 \end{bmatrix} \end{array}$$

$$d_{(135)2} = \min\{d_{(35)2}, d_{12}\} = \min\{7, 9\} = 7$$
$$d_{(135)4} = \min\{d_{(35)4}, d_{14}\} = \min\{8, 6\} = 6$$

$$\begin{array}{c} \\ (135) \\ (24) \end{array} \begin{array}{c} \begin{array}{cc} (135) & (24) \end{array} \\ \begin{bmatrix} 0 & \\ 6 & 0 \end{bmatrix} \end{array}$$



Cluster Algorithm

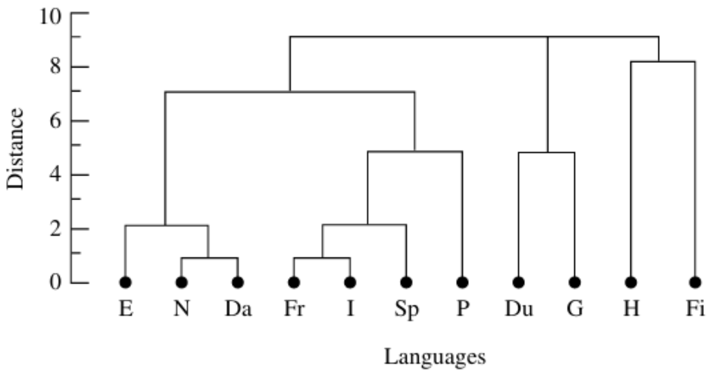
Single Linkage(Nearest Neighbor algorithm)

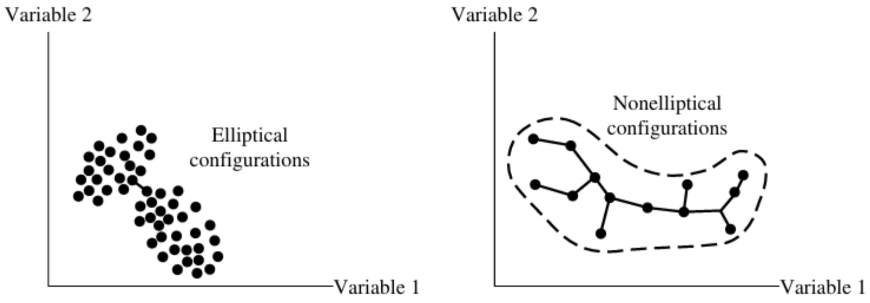
$d_{(UV)W} = \min\{d_{UW}, d_{VW}\}$

Ex) Single linkage clustering of 11 languages

	E	N	Da	Du	G	Fr	Sp	I	P	H	Fi
E	0										
N	2	0									
Da	2	1	0								
Du	7	5	6	0							
G	6	4	5	5	0						
Fr	6	6	6	9	7	0					
Sp	6	6	5	9	7	2	0				
I	6	6	5	9	7	1	1	0			
P	7	7	6	10	8	5	3	4	0		
H	9	8	8	8	9	10	10	10	10	0	
Fi	9	9	9	9	9	9	9	9	9	8	0

$d_{32} = 1; \quad d_{86} = 1; \quad \text{and } d_{87} = 1$





(a) Single linkage confused by near overlap

(b) Chaining effect

Figure 12.5 Single linkage clusters.

Cluster Algorithm

Complete Linkage(Farthest Neighbor algorithm)

$d_{(UV)W} = \delta_1 d_{UW} + \delta_2 d_{VW} + \delta_3 d_{UV} + \delta_4 |d_{UW} - d_{VW}|$ where $\delta_1 = \frac{1}{2},\ \delta_2 = \frac{1}{2}, \delta_3 = 0,\ \delta_{4} = -\frac{1}{2},$

$$d_{(UV)W} = \max\{d_{UW,}\, d_{VW}\}$$

Ex)

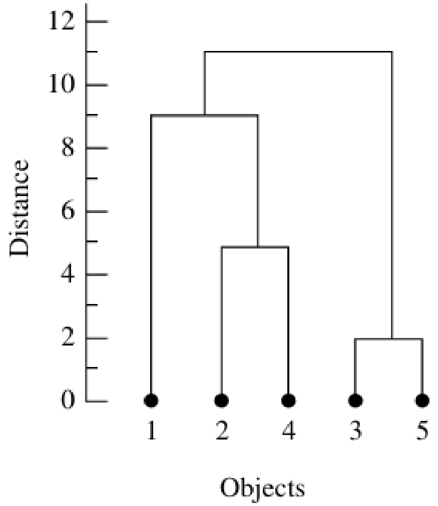
$$\mathbf{D} = \{d_{ik}\} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & & \\ 9 & 0 & & & \\ 3 & 7 & 0 & & \\ 6 & 5 & 9 & 0 & \\ 11 & 10 & 2 & 8 & 0 \end{bmatrix} \end{array}$$

$$d_{(35)1} = \max\{d_{31}, d_{51}\} = \max\{3, 11\} = 11$$
$$d_{(35)2} = \max\{d_{32}, d_{52}\} = 10$$
$$d_{(35)4} = \max\{d_{34}, d_{54}\} = 9$$

$$\begin{array}{c} \\ (35) \\ 1 \\ 2 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} (35) & 1 & 2 & 4 \end{array} \\ \begin{bmatrix} 0 & & & \\ 11 & 0 & & \\ 10 & 9 & 0 & \\ 9 & 6 & 5 & 0 \end{bmatrix} \end{array}$$

$$d_{(24)(35)} = \max\{d_{2(35)}, d_{4(35)}\} = \max\{10, 9\} = 10$$
$$d_{(24)1} = \max\{d_{21}, d_{41}\} = 9$$

$$\begin{array}{c} \\ (35) \\ (24) \\ 1 \end{array} \begin{array}{c} \begin{array}{ccc} (35) & (24) & 1 \end{array} \\ \begin{bmatrix} 0 & & \\ 10 & 0 & \\ 11 & 9 & 0 \end{bmatrix} \end{array}$$



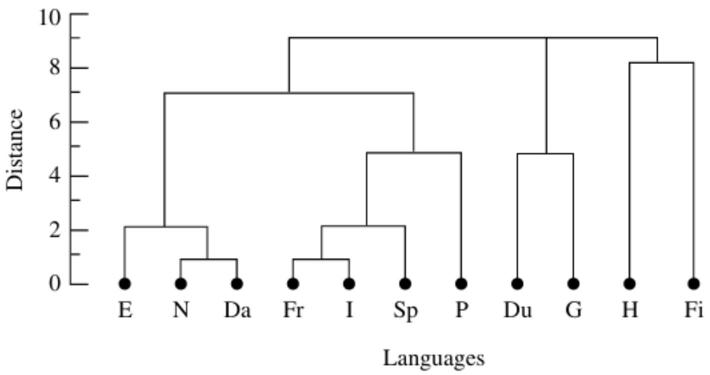
Cluster Algorithm

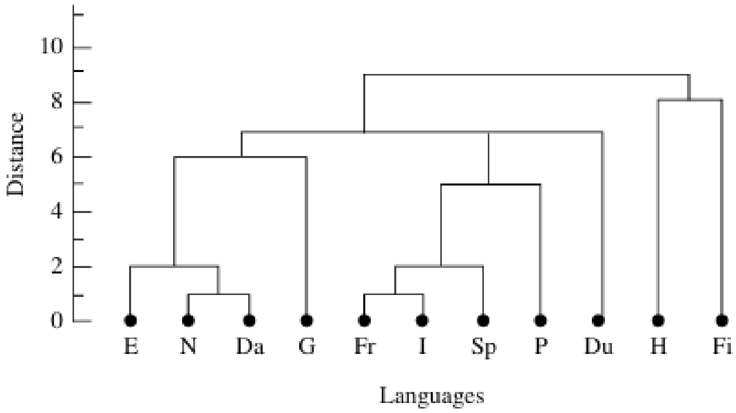
Average Linkage algorithm

$d_{(UV)W} = \delta_1 d_{UW} + \delta_2 d_{VW} + \delta_3 d_{UV} + \delta_4 |d_{UW} - d_{VW}|$ where $\delta_1 = \frac{N_U}{N_U + N_V}$, $\delta_2 = \frac{N_V}{N_U + N_V}$, $\delta_3 = 0$, $\delta_4 = 0$,

$$\Rightarrow d_{(UV)W} = \frac{\sum_i \sum_k d_{ik}}{N_{(UV)} N_W}$$

Ex) (Complete linkage) vs (Average linkage) clustering of 11 languages



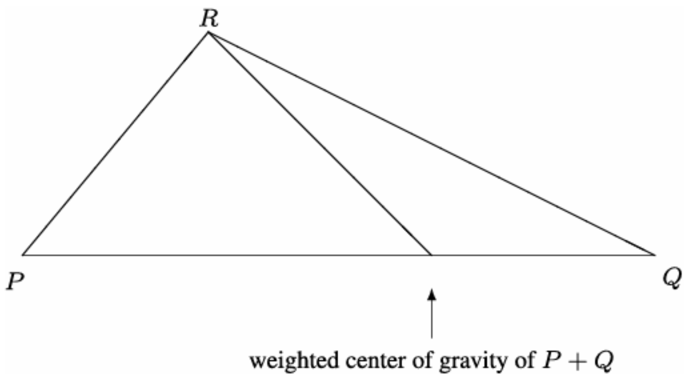


Cluster Algorithm

Centroid algorithm

$$d_{(UV)W} = \delta_1 d_{UW} + \delta_2 d_{VW} + \delta_3 d_{UV} + \delta_4 |d_{UW} - d_{VW}| \quad \text{where } \delta_1 = \frac{N_U}{N_U+N_V},\ \delta_2 = \frac{N_V}{N_U+N_V}, \delta_3 = -\frac{N_U N_V}{(N_U+N_V)^2},\ \delta_4 = 0,$$

Ex)



Cluster Algorithm

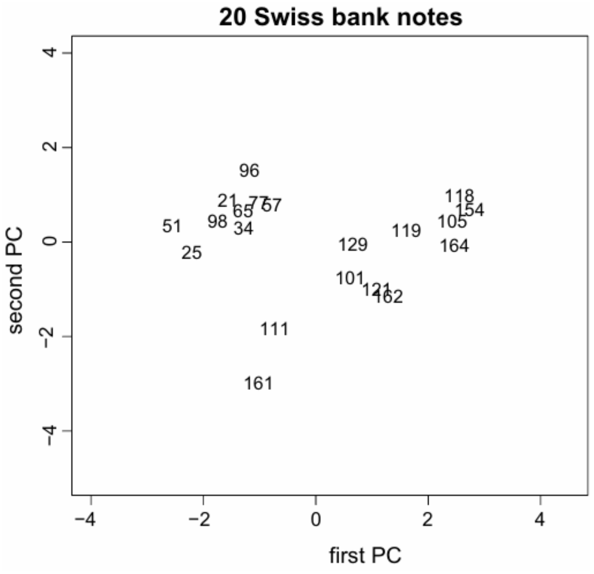
Ward algorithm

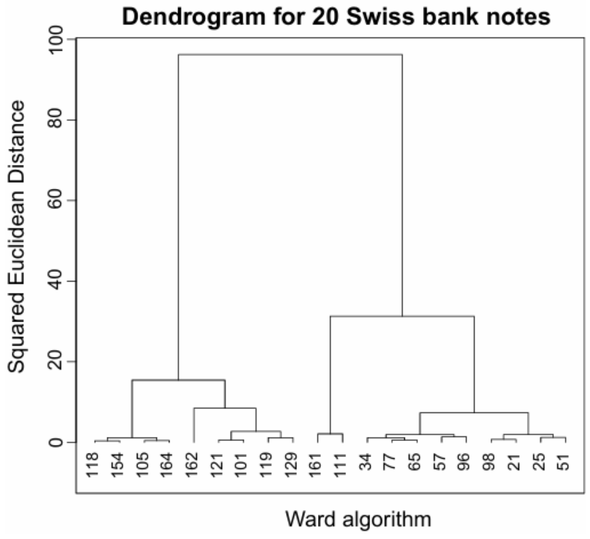
$d_{(UV)W} = \delta_1 d_{UW} + \delta_2 d_{VW} + \delta_3 d_{UV} + \delta_4 |d_{UW} - d_{VW}|$ where $\delta_1 = \frac{N_W + N_U}{N_U + N_V + N_W}$, $\delta_2 = \frac{N_W + N_V}{N_U + N_V + N_W}$, $\delta_3 = -\frac{N_W}{N_U + N_V + N_W}$, $\delta_4 = 0$,

$$I_R = \frac{1}{n_R} \sum_{i=1}^{n_R} d^2(x_i, \overline{x}_R)$$

$$\Delta(P, Q) = \frac{n_P n_Q}{n_P + n_Q} d^2(P, Q)$$

Ex) 20 Swiss bank notes

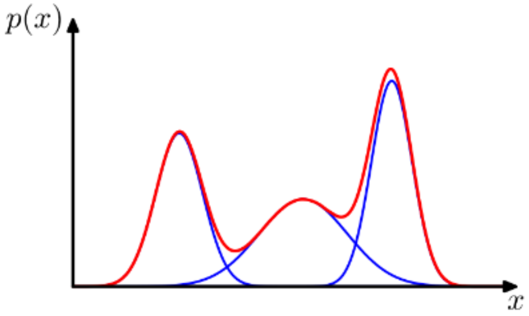




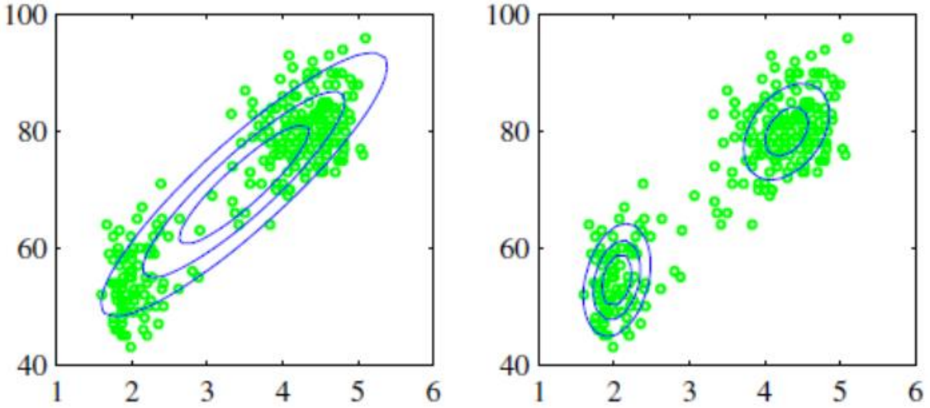
Cluster Algorithm

Clustering based on Statistical Models

- data가 특정한 분포를 따르는 데이터일 때의 clustering



ex) 3개의 정규분포가 결합된 혼합분포



$$f_{Mix}(\mathbf{x}) = \sum_{k=1}^{K} p_k\, f_k(\mathbf{x})$$

=> p_k의 확률로 f_k의 분포를 따른다! : Mixing distribution

Cluster Algorithm

Clustering based on Statistical Models

$$f_{Mix}(\mathbf{x}) = \sum_{k=1}^{K} p_k f_k(\mathbf{x}) \qquad f_{Mix}(\mathbf{x} \mid \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K)$$

$$= \sum_{k=1}^{K} p_k \frac{1}{(2\pi)^{p/2} \mid \boldsymbol{\Sigma}_k \mid^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)' \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)\right)$$

$$L(p_1, \ldots, p_K, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_K) = \prod_{j=1}^{N} f_{Mix}(\mathbf{x}_j \mid \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K)$$

$$= \prod_{j=1}^{N} \left(\sum_{k=1}^{K} p_k \frac{1}{(2\pi)^{p/2} \mid \boldsymbol{\Sigma}_k \mid^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}_j - \boldsymbol{\mu}_k)' \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_j - \boldsymbol{\mu}_k)\right)\right)$$

$$L_{\max} = L(\hat{p}_1, \ldots, \hat{p}_K, \hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\Sigma}}_1, \ldots, \hat{\boldsymbol{\mu}}_K, \hat{\boldsymbol{\Sigma}}_K)$$

Cluster Algorithm

Clustering based on Statistical Models

$$L(p_1, \ldots, p_K, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_K) = \prod_{j=1}^{N} f_{Mix}(\mathbf{x}_j \,|\, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K)$$

$$= \prod_{j=1}^{N}\left(\sum_{k=1}^{K} p_k \frac{1}{(2\pi)^{p/2} \,|\, \boldsymbol{\Sigma}_k \,|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}_j - \boldsymbol{\mu}_k)' \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_j - \boldsymbol{\mu}_k)\right)\right)$$

$$\text{AIC} = 2 \ln L_{\max} - 2N\left(K\frac{1}{2}(p+1)(p+2) - 1\right)$$

$$\text{BIC} = 2 \ln L_{\max} - 2 \ln(N)\left(K\frac{1}{2}(p+1)(p+2) - 1\right)$$

Assumed form for $\boldsymbol{\Sigma}_k$	Total number of parameters	BIC
$\boldsymbol{\Sigma}_k = \eta \mathbf{I}$	$K(p+1)$	$\ln L_{\max} - 2\ln(N)K(p+1)$
$\boldsymbol{\Sigma}_k = \eta_k \mathbf{I}$	$K(p+2) - 1$	$\ln L_{\max} - 2\ln(N)(K(p+2) - 1)$
$\boldsymbol{\Sigma}_k = \eta_k \, Diag(\lambda_1, \lambda_2, \ldots, \lambda_p)$	$K(p+2) + p - 1$	$\ln L_{\max} - 2\ln(N)(K(p+2) + p - 1)$

Cluster Algorithm

Clustering based on Statistical Models

Ex) A model based clustering of the iris data

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 5.01 \\ 3.43 \\ 1.46 \\ 0.25 \end{bmatrix}, \quad \boldsymbol{\mu}_2 = \begin{bmatrix} 5.90 \\ 2.75 \\ 4.40 \\ 1.43 \end{bmatrix}, \quad \boldsymbol{\mu}_3 = \begin{bmatrix} 6.85 \\ 3.07 \\ 5.73 \\ 2.07 \end{bmatrix}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 5.01 \\ 3.43 \\ 1.46 \\ 0.25 \end{bmatrix}, \quad \boldsymbol{\mu}_2 = \begin{bmatrix} 6.26 \\ 2.87 \\ 4.91 \\ 1.68 \end{bmatrix}$$

$$\hat{\boldsymbol{\Sigma}}_1 = \begin{bmatrix} .1218 & .0972 & .0160 & .0101 \\ .0972 & .1408 & .0115 & .0091 \\ .0160 & .0115 & .0296 & .0059 \\ .0101 & .0091 & .0059 & .0109 \end{bmatrix} \qquad \hat{\boldsymbol{\Sigma}}_2 = \begin{bmatrix} .4530 & .1209 & .4489 & .1655 \\ .1209 & .1096 & .1414 & .0792 \\ .4489 & .1414 & .6748 & .2858 \\ .1655 & .0792 & .2858 & .1786 \end{bmatrix}$$

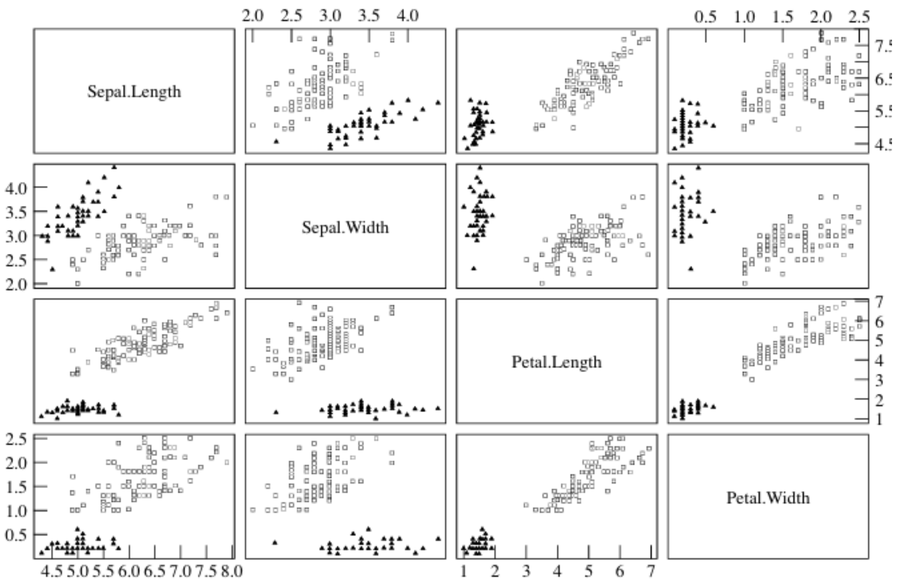


Figure 12.13 Multiple scatter plots of $K = 3$ clusters for Iris data

4. Adaptive Weights Clustering (AWC)

Adaptive Weights Clustering (AWC)

Notation

주어진 데이터: $X_1 \sim X_n \subset \mathbb{R}^p$ (p가 매우 큰 경우도 가능)

X_i와 X_j의 거리는 $d(X_i,\ X_j)$로 표현

가중치 행렬 $W = (w_{ij}), \quad i, j = 1...n$ (w_{ij}는 binary)

$w_{ij} = 1$은 X_i와 X_j가 같은 군집에 속한다는 의미

$w_{ij} = 0$은 X_i와 X_j가 다른 군집에 속한다는 의미

C_i는 고정된 i에 대해서 w_{ij}가 양수인 j로 이루어진 cluster

Adaptive Weights Clustering (AWC)

Overview

AWC 알고리즘은 순차적으로 w_{ij}를 새롭게 계산하면서 clustering을 진행

처음$(k=0)$에는 초기값 $w_{ij}^{(0)}$를 통해서 $C_i^{(0)}$을 구성

$k \geq 1$ 단계에서는 $C_i^{(k-1)}$과 $C_j^{(k-1)}$ 사이에 “**no gab test**”를 진행해 $w_{ij}^{(k)}$를 업데이트

(이때, $d(X_i, X_j) \leq h_k$인 $X_i,\ X_j$에 대해서만 진행한다.)

이 과정을 $k=K$까지 반복해주고 완성된 W를 통해서 clustering

Adaptive Weights Clustering (AWC)

Sequence of radii

각 단계마다 기준치가 되는 **반경**

$h_1 \le h_2 \le \ldots \le h_K$

h_k는 다음과 같은 조건을 만족하도록 설정

$n(X_i, h_{k+1}) \le a \cdot n(X_i, h_k), \quad h_{k+1} \le b \cdot h_k$

$(a = \sqrt{2},\ b = 1.95)$

Adaptive Weights Clustering (AWC)

Initialization of weights

초기 단계에서는 각 point를 n_0개의 가까운 이웃들만 가중치 부여 $(n_0 = 2p + 2)$

$$w_{ij}^{(0)} = I[\ d(X_i, X_j) \leq max\{h_0(X_i),\ h_0(X_j)\}\ ]$$

$h_0(X_i)$는 X_i와 n_o번째로 가까운 데이터 사이의 거리

Adaptive Weights Clustering (AWC)

Updates at step k

$k-1$번째 단계에서의 결과는 주어져 있다고 가정

각각의 X_i에 대해서 가중치 $\{w_{ij}^{(k-1)}, j = 1, ..., n\}$를 가지고 있음

이때, $w_{ij} = 1$은 X_j가 다음을 만족한다는 의미

$B(X_i, h_{k-1}) = \{x : d(X_i, x) \leq h_{k-1}\}$ or $d(X_i, X_j) \leq h_{k-1}$

$d(X_i, X_j) \leq h_k$ 를 만족하는 point에 대해서만 w_{ij} 업데이트

Adaptive Weights Clustering (AWC)

Updates at step k

$$N_{i\wedge j}^{(k)} = \sum_{l\neq i,j} w_{il}^{(k-1)} w_{jl}^{(k-1)}.$$

$$N_{i\triangle j}^{(k)} = \sum_{l\neq i,j} \Big\{ w_{il}^{(k-1)} \mathrm{I}(X_l \notin B(X_j, h_{k-1})) + w_{jl}^{(k-1)} \mathrm{I}(X_l \notin B(X_i, h_{k-1})) \Big\}.$$

$$N_{i\vee j}^{(k)} = N_{i\wedge j}^{(k)} + N_{i\triangle j}^{(k)}$$

$$\tilde{\theta}_{ij}^{(k)} = N_{i\wedge j}^{(k)} \,/\, N_{i\vee j}^{(k)} .$$

Adaptive Weights Clustering (AWC)

Updates at step k

$\tilde{\theta}_{ij}^{(k)}$ 는 $B(X_i, h_k)$와 $B(X_j, h_k)$의 교집합과 합집합의 비율의 추정치

$$\tilde{\theta}_{ij}^{(k)} \approx q_{ij}^{(k)} = \frac{V_{\cap}(d_{ij}, h_{k-1})}{2V(h_{k-1}) - V_{\cap}(d_{ij}, h_{k-1})}$$

만약 $\tilde{\theta}_{ij}^{(k)}$가 $q_{ij}^{(k)}$ 충분히 작다면, 두 군집간의 gap이 크다는 것을 의미

두 군집 간의 gap이 크다면 두 군집은 합치기X

두 군집 간의 gap이 작다면 두 군집은 합치기O

$\tilde{\theta}_{ij}^{(k)} > q_{ij}^{(k)}$ vs $\tilde{\theta}_{ij}^{(k)} \leq q_{ij}^{(k)}$

Adaptive Weights Clustering (AWC)

Updates at step k

$$T_{ij}^{(k)} = N_{i \vee j}^{(k)} \; KL\big(\tilde{\theta}_{ij}^{(k)}, q_{ij}^{(k)}\big) \big\{\mathrm{I}(\tilde{\theta}_{ij}^{(k)} \le q_{ij}^{(k)}) - \mathrm{I}(\tilde{\theta}_{ij}^{(k)} > q_{ij}^{(k)})\big\}.$$

$KL(\theta, \eta)$ 는 Kullback-Leibler(KL) divergence로 주로 두 분포 간에 차이를 볼 때 사용

$$KL(\theta, \eta) = \theta \log \frac{\theta}{\eta} + (1 - \theta) \log \frac{1-\theta}{1-\eta}$$

0보다 크거나 같은 값을 가짐

Adaptive Weights Clustering (AWC)

Updates at step k

$d(X_i, X_j) \leq h_k$를 만족하는 $X_i,\ X_j$에 대해서 다음과 같이 w_{ij} 업데이트

$$w_{ij}^{(k)} = \mathrm{I}\left(T_{ij}^{(k)} \leq \lambda\right)$$

λ는 tuning parameter로 clustering에 큰 영향을 끼침

만약 λ가 크다면 적은 수의 통합된 군집이 생성되고 λ가 작다면 많은 수의 개별적인 군집 생성

Adaptive Weights Clustering (AWC)

Choose lambda

$$S(\lambda) = \sum_{i,j=1}^{n} w_{ij}^{K}(\lambda).$$

λ 값을 변화시켜가며 $S(\lambda)$를 계산 (λ가 크다면 $S(\lambda)$가 크고, λ가 작다면 $S(\lambda)$가 작음)

$S(\lambda)$가 급격하게 변할 경우, 직전의 λ 선택

만약, $S(\lambda)$ 변하는 구간이 여러 개인 경우 λ를 비교해가며 선택

Adaptive Weights Clustering (AWC)

AWC Algorithm

Algorithm 13.5 AWC

1: **Fix** a sequence of radii $h_1 \le h_2 \le \ldots \le h_K$
2: **Initialization of weights**: $w_{ij}^{(0)} = \mathrm{I}\left(d(X_i, X_j) \le \max(h_0(X_i), h_0(X_j))\right)$
3: **Updates at step** k :
4: Compute $T_{ij}^{(k)}$ using 13.27
5: $w_{ij}^{(k)} = \mathrm{I}\left(d(X_i, X_j) \le h_k\right) \mathrm{I}\left(T_{ij}^{(k)} \le \lambda\right)$
6: **Repeat** until $k = K$.

5. Spectral Clustering

Spectral Clustering

Notation

$G = (V, E)$

weighted adjacency matrix

degree matrix

$$d_i = \sum_{j=1}^{n} w_{ij}.$$

Spectral Clustering

Notation

$A \subset V$ 일 때, $V \setminus A$ 는 $\bar{A}$로 정의

indicator vector $1_A = (f_1, ..., f_n)'$, 만약 $v_i \in A$라면 $f_i = 1$, 아니면 $f_i = 0$

$$W(A, B) := \sum_{i \in A, j \in B} w_{ij}.$$

$$|A| := \text{ the number of vertices in } A$$

$$\mathrm{vol}(A) := \sum_{i \in A} d_i.$$

Spectral Clustering

How to make Similarity graph

데이터 x_1 ~ x_n가 주어졌을 때, x_i와 x_j간의 유사도를 나타내는 s_{ij} 또는 d_{ij}를 활용하여

Similarity Graph 생성

The ϵ-neighborhood graph

데이터 간의 거리가 ϵ보다 작은 경우에만 이어줌

일반적으로 unweighted graph로 간주

k-nearest neighbor graphs

v_j가 v_i의 k번째 가까운 노드에 속하면 연결

연결 후 노드의 유사도에 따라 엣지에 가중치 부여

The fully connected graph

양의 유사도를 가진 데이터끼리 연결

Gaussian 함수를 이용하여 s_{ij}를 만들고 엣지에 s_{ij}를 가중치로 부여

$s(x_i, x_j) = exp(-\|x_i - x_j\|^2/(2\sigma^2))$

Spectral Clustering

Laplacian Matrix

G는 undirected, weighted graph로 가정

$$L = D - W$$

1. *For every vector* $f \in \mathbb{R}^n$ *we have*

$$f'Lf = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2.$$

2. L *is symmetric and positive semi-definite.*
3. *The smallest eigenvalue of* L *is 0, the corresponding eigenvector is the constant one vector* $\mathbb{1}$.
4. L *has* n *non-negative, real-valued eigenvalues* $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.

Spectral Clustering

Laplacian Matrix

앞선 성질에 대한 증명

$$f'Lf = f'Df - f'Wf = \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} f_i f_j w_{ij}$$

$$= \frac{1}{2}\left(\sum_{i=1}^{n} d_i f_i^2 - 2\sum_{i,j=1}^{n} f_i f_j w_{ij} + \sum_{j=1}^{n} d_j f_j^2\right) = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

W와 D가 symmetry이고 $f'Lf \geq 0$ for all $f \in \mathbb{R}^n$이므로 positive semi definite

Spectral Clustering

Algorithm

Unnormalized spectral clustering

Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct.

- Construct a similarity graph by one of the ways described in Section 2. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L.
- **Compute the first k eigenvectors $u_1, \dots, u_k$ of L.**
- Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \dots, u_k$ as columns.
- For $i = 1, \dots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U.
- Cluster the points $(y_i)_{i=1,\dots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $C_1, \dots, C_k$.

Output: Clusters $A_1, \dots, A_k$ with $A_i = \{j \mid y_j \in C_i\}$.

Spectral Clustering

Graph cut point of view

그래프가 주어졌을 때, 서로 다른 그룹 사이의 엣지는 낮은 가중치를 갖도록, 같은 그룹내에서의 엣지는 높은 가중치를 갖도록 나누고 싶음

$$\mathrm{cut}(A_1, \ldots, A_k) := \frac{1}{2} \sum_{i=1}^{k} W(A_i, \overline{A}_i).$$

k=2인 경우

$$cut(A, \bar{A}) := \frac{1}{2} \cdot W(A, \bar{A})$$

Spectral Clustering

Graph cut point of view

그룹의 크기를 고려하는 RatioCut

$$\mathrm{RatioCut}(A_1, \ldots, A_k) := \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \overline{A}_i)}{|A_i|} = \sum_{i=1}^{k} \frac{\mathrm{cut}(A_i, \overline{A}_i)}{|A_i|}$$

k=2인 경우

$$RatioCut(A, \bar{A}) = cut(A, \bar{A}) \times \left(\frac{1}{|A|} + \frac{1}{|\bar{A}|}\right)$$

Spectral Clustering

Approximating RatioCut for k=2

우리는 주어진 데이터를 그래프로 바꿀 수 있음

그래프에 대해서 서로 다른 그룹 사이의 엣지는 낮은 가중치를 갖도록, 같은 그룹내에서의 엣지는 높은 가중치를 갖도록 나누고 싶음

그래프를 RatioCut을 가장 작게 하는 k개의 cluster로 나누면 됨 (k=2인 경우)

다음과 같은 목적함수를 갖는 최적화 문제를 풀면 됨

$$\min_{A \subset V} \operatorname{RatioCut}(A, \overline{A})$$

Spectral Clustering

Approximating RatioCut for k=2

벡터 $f = (f_1, \ldots, f_n)' \in \mathbb{R}^n$ 의 entry f_i를 다음과 같이 설정

$$f_i = \begin{cases} \sqrt{|\overline{A}|/|A|} & \text{if } v_i \in A \\ -\sqrt{|A|/|\overline{A}|} & \text{if } v_i \in \overline{A}. \end{cases}$$

$$\begin{aligned} f'Lf &= \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2 \\ &= \frac{1}{2}\sum_{i\in A, j\in\overline{A}} w_{ij}\left(\sqrt{\frac{|\overline{A}|}{|A|}} + \sqrt{\frac{|A|}{|\overline{A}|}}\right)^2 + \frac{1}{2}\sum_{i\in\overline{A}, j\in A} w_{ij}\left(-\sqrt{\frac{|\overline{A}|}{|A|}} - \sqrt{\frac{|A|}{|\overline{A}|}}\right)^2 \\ &= \text{cut}(A,\overline{A})\left(\frac{|\overline{A}|}{|A|} + \frac{|A|}{|\overline{A}|} + 2\right) \\ &= \text{cut}(A,\overline{A})\left(\frac{|A| + |\overline{A}|}{|A|} + \frac{|A| + |\overline{A}|}{|\overline{A}|}\right) \\ &= |V| \cdot \text{RatioCut}(A,\overline{A}). \end{aligned}$$

$$\sum_{i=1}^{n} f_i = \sum_{i\in A}\sqrt{\frac{|\overline{A}|}{|A|}} - \sum_{i\in\overline{A}}\sqrt{\frac{|A|}{|\overline{A}|}} = |A|\sqrt{\frac{|\overline{A}|}{|A|}} - |\overline{A}|\sqrt{\frac{|A|}{|\overline{A}|}} = 0.$$

$$\|f\|^2 = \sum_{i=1}^{n} f_i^2 = |A|\frac{|\overline{A}|}{|A|} + |\overline{A}|\frac{|A|}{|\overline{A}|} = |\overline{A}| + |A| = n.$$

Spectral Clustering

Approximating RatioCut for k=2

$$\min_{A \subset V} \operatorname{RatioCut}(A, \overline{A})$$

$$\min_{A \subset V} f' L f \text{ subject to } f \perp \mathbb{1}, \ \|f\| = \sqrt{n}.$$

$$\min_{f \in \mathbb{R}^n} f' L f \text{ subject to } f \perp \mathbb{1}, \ \|f\| = \sqrt{n}.$$

최적해: 벡터 f는 L 행렬의 2번째로 작은 고유값에 대응하는 고유벡터

$$\begin{cases} v_i \in A & \text{if } f_i \geq 0 \\ v_i \in \overline{A} & \text{if } f_i < 0. \end{cases}$$

Spectral Clustering

Approximating RatioCut for arbitrary k

주어진 V를 $A_1, \ldots, A_k$로 나눌 때, indicatort vector $h_j = (h_{1,j}, \ldots, h_{n,j})$의 entry를 다음과 같이 설정

$$h_{i,j} = \begin{cases} 1/\sqrt{|A_j|} & \text{if } v_i \in A_j \\ 0 & \text{otherwise} \end{cases}$$

$$h_i' L h_i = \frac{\mathrm{cut}(A_i, \overline{A}_i)}{|A_i|}. \qquad h_i' L h_i = (H' L H)_{ii}.$$

$$\mathrm{RatioCut}(A_1, \ldots, A_k) = \sum_{i=1}^{k} h_i' L h_i = \sum_{i=1}^{k} (H' L H)_{ii} = \mathrm{Tr}(H' L H),$$

Spectral Clustering

Approximating RatioCut for arbitrary k

$$\min_{A_1,\ldots,A_k} RatioCut(A_1, \ldots, A_k)$$

$$\min_{A_1,\ldots,A_k} \operatorname{Tr}(H'LH) \text{ subject to } H'H = I$$

$$\min_{H\in\mathbb{R}^{n\times k}} \operatorname{Tr}(H'LH) \text{ subject to } H'H = I.$$

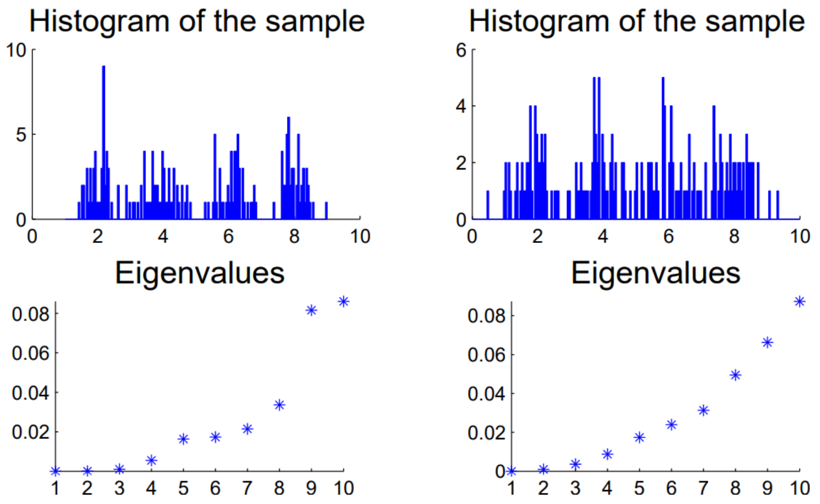
최적해: H행렬은 L행렬의 k개의 고유값(작은 순서대로)에 대응하는 고유벡터 k개가 열로 이루어짐

Spectral Clustering

How to choose k

eigengap heuristic 사용

L의 $\lambda_1, \ldots, \lambda_k$는 작은데 λ_{k+1}이 상대적으로 커지게 되는 k 선택



Spectral Clustering

Algorithm

Unnormalized spectral clustering

Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct.

- Construct a similarity graph by one of the ways described in Section 2. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L.
- **Compute the first k eigenvectors $u_1, \dots, u_k$ of L.**
- Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \dots, u_k$ as columns.
- For $i = 1, \dots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U.
- Cluster the points $(y_i)_{i=1,\dots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $C_1, \dots, C_k$.

Output: Clusters $A_1, \dots, A_k$ with $A_i = \{j \mid y_j \in C_i\}$.

END