Seoul National University
Dept. of Civil and Environmental Engineering

Instructor: Junho Song
junhosong@snu.ac.kr

M1586.002500 Information Engineering for CE Engineers

In-Class Material: Class 25

Unsupervised Learning (ISL Chapter 10)

1. Clustering Methods

(a) A very broad set of techniques for finding *subgroups* or *clusters* in a data set

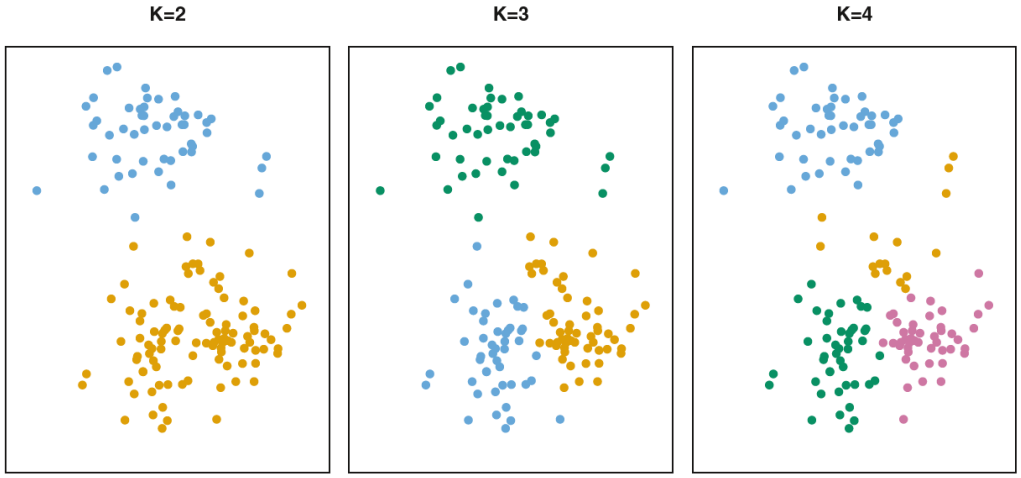
- Seek to partition the observations of a data set into distinct groups so that the observations within each group are quite *similar* to each other, while observations in different group are quite different from each other

(b) "What is different from PCA?"

- PCA: Find a low-dimensional representation of the observations that explain a good fraction of the variance
- Clustering: Find homogeneous subgroups among the observations (or features)

2. *K*-Means Clustering

(a) Basic concepts



- Observations: data points with dimension (p) belonging to *clusters*
- $C_1, C_2, \cdots, C_K$: sets containing the indices of the observations in each cluster

K-means clustering is a simple approach for partitioning a data set into K distinct, non-overlapping clusters

In other words, each observation belongs to *exactly one* of the K clusters, i.e. the clusters are non-overlapping

$$C_1 \cup C_2 \cup \cdots \cup C_K = \{1, \cdots, n\}$$

$$C_k \cap C_{k'} = \emptyset \quad for\ all\ k \neq k'$$

Instructor: Junho Song
junhosong@snu.ac.kr

- "Within-cluster" variation $W(C_k)$: Sum of the Euclidean distances between pairs in the *k*th cluster divided by the number of observations

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

- $|C_k|$: the number of observations in the kth cluster

- $\sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$: Squared Euclidean distance between two observations i and i'

K-means clustering aims to identify clusters minimizing within-cluster variations. The optimization problem is

$$\min_{C_1,\cdots,C_K} \left\{ \sum_{k=1}^{K} W(C_k) \right\} = \min_{C_1,\cdots,C_K} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}.$$

However, it is quite a difficult optimization problem to solve precisely, since there are almost K^n ways to partition n observations into K clusters. To find a local optimum efficiently, an alternative simple algorithm can be used.

(b) Local optimization algorithm

Algorithm 10.1 *K-Means Clustering*

1. Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.

2. Iterate until the cluster assignments stop changing:

   A. For each of the K clusters, compute the cluster centroid. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.

   B. Assign each observation to the cluster whose centroid is closest (where closest is defined using Euclidean distance).

Algorithm 10.1 *is guaranteed to decrease* the value of the objective ($\sum_{k=1}^{K} W(C_k)$) at each step. To understand why, the following identity is illuminating:

$$\frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = 2 \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2,$$

where $\bar{x}_{kj} = \frac{1}{|C_k|} \sum_{i \in C_k} x_{ij}$ is the mean for feature j in cluster C_k.

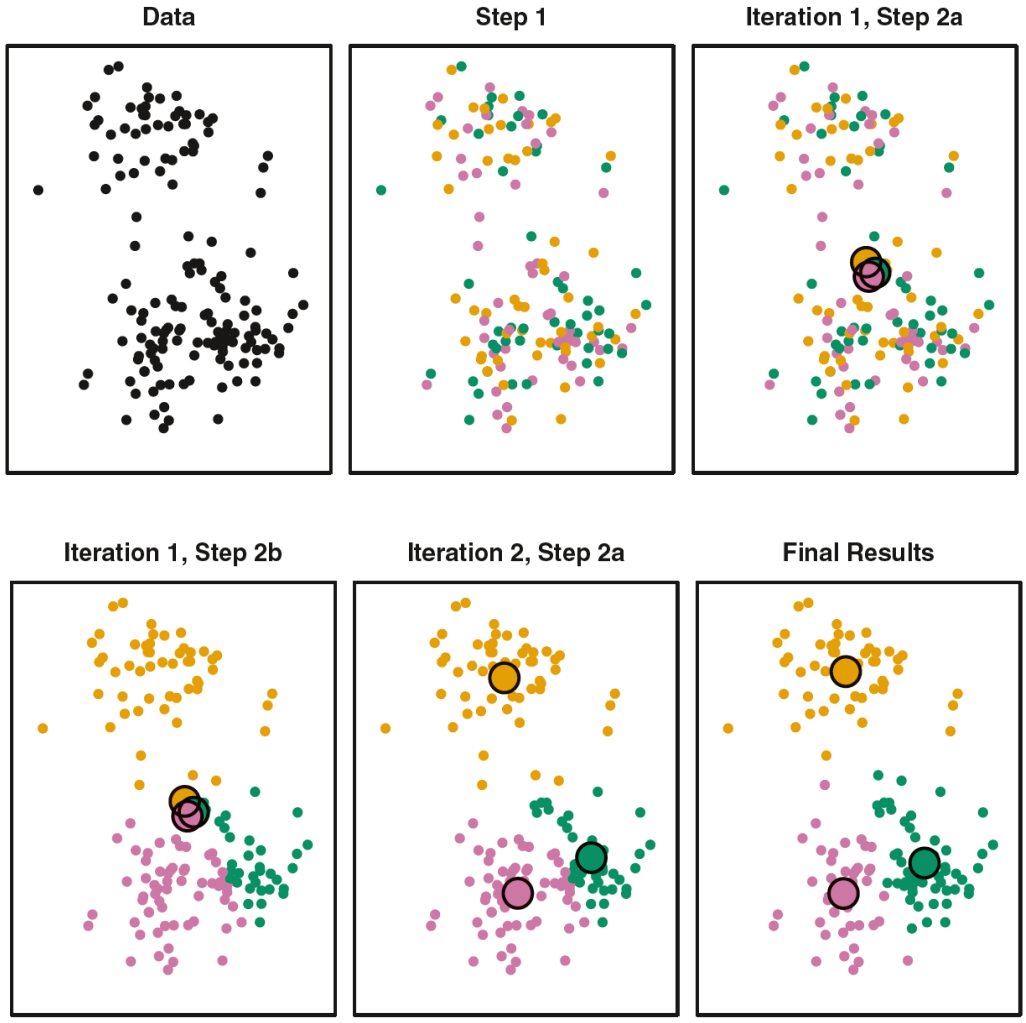
- Step 2A: the cluster means for each feature are the constants that would minimize the sum-of-squared deviations

- Step 2B: reallocating the observations can only improve $2 \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2$.

Instructor: Junho Song
junhosong@snu.ac.kr

This means that as the algorithm is run, the clustering obtained will continually improve until the result no longer changes; the objective $\sum_{k=1}^{K} W(C_k)$ will never increase (from the random initial cluster). When the result no longer changes, a *local optimum* has been reached.

- The figure below shows the progression of the *Algorithm 10.1*.



(c) Global optimization

Because the K-means algorithm finds a local rather than a global optimum, it is important to run the algorithm multiple times from different random initial configurations.

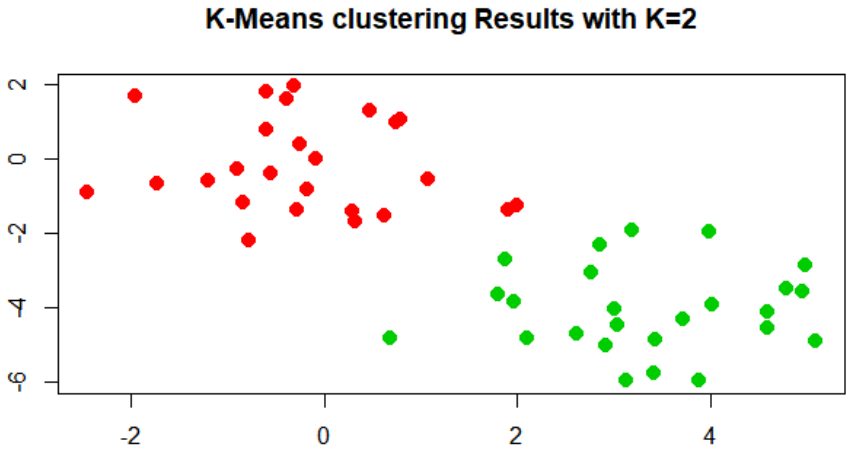
- Global optimization: the best solution with the smallest objective $\sum_{k=1}^{K} W(C_k)$ in the multiple local optima

The figure below shows the local optima obtained by running K-menas algorithm six times using six different initial cluster assignment (Step 1. of the Algorithm 10.1). In this case, the best clustering is the one with an objective value of 235.8.

Instructor: Junho Song
junhosong@snu.ac.kr

- Weakness of the K-means clustering: the problem of selecting K

(d) A simple example



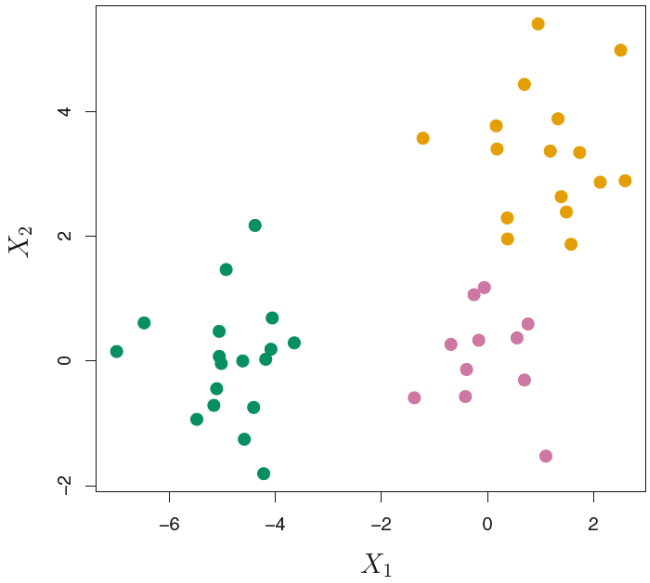
Instructor: Junho Song
junhosong@snu.ac.kr

```
set.seed(2)
x=matrix(rnorm(50*2), ncol=2)
x[1:25,1]=x[1:25,1]+3
x[1:25,2]=x[1:25,2]-4
km.out=kmeans(x,2,nstart=20)  # K=2
plot(x, col=(km.out$cluster+1), main="K-Means clustering Results with
   K=2", xlab="", ylab="", pch=20, cex=2)

set.seed(4)
km.out=kmeans(x,3,nstart=20)  # K=3
plot(x, col=(km.out$cluster+1), main="K-Means clustering Results with
   K=3", xlab="", ylab="", pch=20, cex=2)

set.seed(3)
km.out=kmeans(x,3,nstart=1)
km.out$tot.withinss  # the total within-cluster sum of squares
km.out=kmeans(x,3,nstart=20)
km.out$tot.withinss
```

3. Hierarchical Clustering

- An alternative approach of K-means clustering that requires a particular choice of *K*

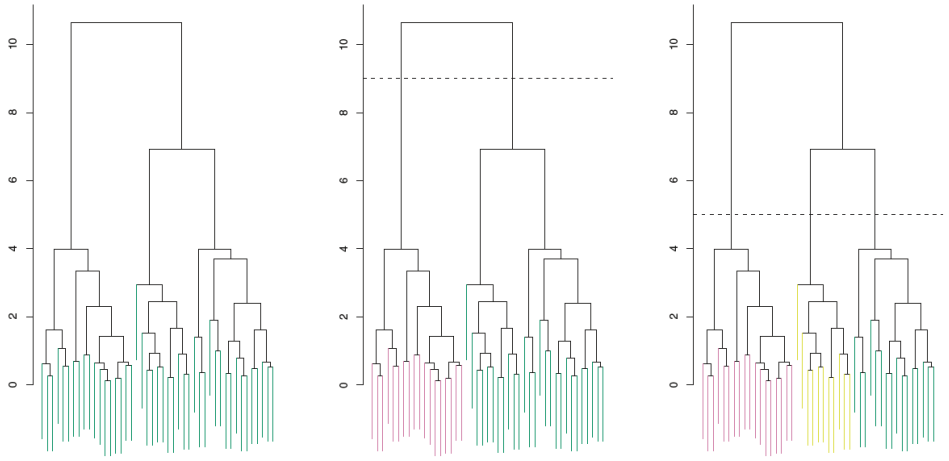
(a) Interpreting a *dendrogram*



The data were generated from a three-class model to explain a dendrogram concept; the true class labels for each observation are shown in distinct colors.

Suppose that the data were observed without the class labels, hierarchical clustering yields the result shown in the dendrograms below.

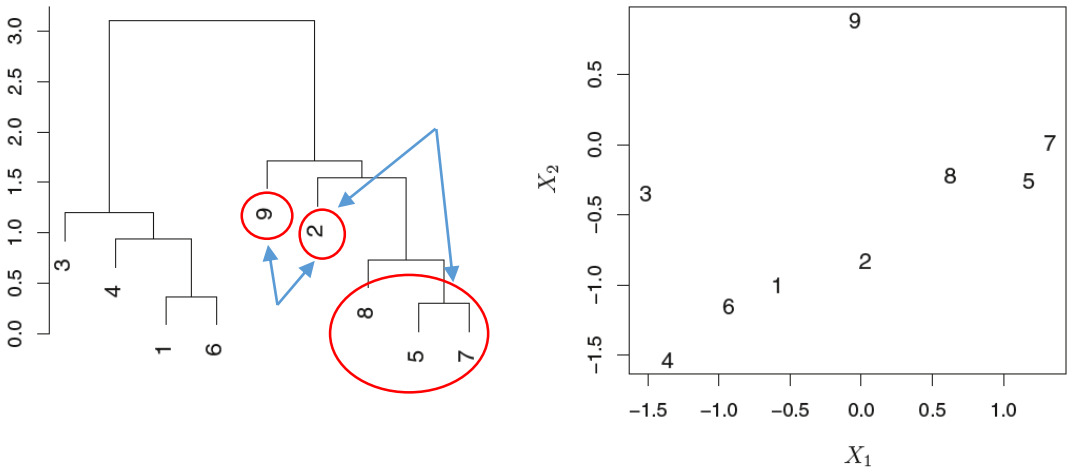
Instructor: Junho Song
junhosong@snu.ac.kr



- Dendrogram: a tree-based representation of the observations
- Leaf: a lowest branch in a dendrogram which means each observation
- Fusion: some *leaves* begin to *fuse* into *branches*. These correspond to observations that are similar to each other. At the higher part of the tree, *branches* themselves *fuse*, either with *leaves* or *other branches*.

For any two observations, one can locate the point of first fusion in the branches of the dendrogram. The height of this fusion (on the vertical axis) indicates how different the two observations are.

Note: We cannot draw conclusions about the similarity of two observations based on their proximity along the horizontal axis.



Horizontal cuts are shown in the center and right-hand panels of figure (top). The height of the cut in the dendrogram serves the same role as the number of clusters (K) obtained. One can select the number of clusters by looking at the dendrogram.

- In the center panel: cutting at a height of 9, and resulting in two clusters
- In the right-hand panel: cutting at a height of 5, and resulting in three clusters.

Seoul National University
Dept. of Civil and Environmental Engineering

Instructor: Junho Song
junhosong@snu.ac.kr

M1586.002500 Information Engineering for CE Engineers

In-Class Material: Class 26

Unsupervised Learning (ISL Chapter 10)

3. Hierarchical Clustering (contd.)

(b) The Hierarchical Clustering Algorithm

- Extremely simple algorithm: fuse the two clusters that are most "similar" to each other to construct the dendrogram (See the figure on Page 3)

- To quantifying similarity, dissimilarity measure (e.g. Euclidean distance) is used

Algorithm 10.2 *Hierarchical Clustering*

1. Begin with n observations and a measure (such as Euclidean distance) of all the $\binom{n}{2} = n(n-1)/2$ pairwise dissimilarities. Treat each observation as its own cluster.

2. For $i = n, n-1, \cdots, 2$:

   A. Examine all pairwise inter-cluster dissimilarities among the i clusters and identify the pair of clusters that are least dissimilar (that is, most similar). Fuse these two clusters. The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion should be placed.

   B. Compute the new pairwise inter-cluster dissimilarities among the $(i-1)$ remaining clusters.

- Step 2A & 2B: the two clusters that are most similar to each other are then fused so that there now are $n-1$ clusters. The algorithm proceeds in this fashion until all of the observation belong to one single cluster.

To address the problem defining the dissimilarity between two clusters if one or both of the clusters contains multiple observations, the notion of ***linkage*** is developed.

- Linkage: defines the dissimilarity between two *groups* of observations.

The below table is a summary of the four most commonly-used types of linkage in hierarchical clustering.

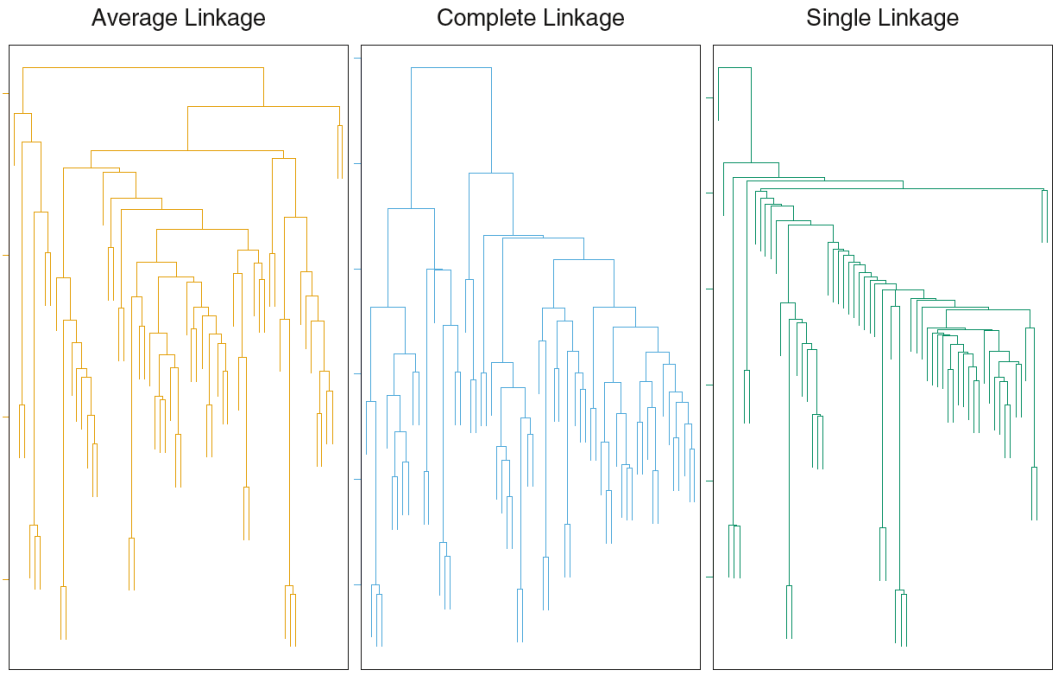
Linkage	Description
Complete	Maximal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *largest* of these dissimilarities.
Single	Minimal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *smallest* of these dissimilarities. Single linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time.

Instructor: Junho Song
junhosong@snu.ac.kr

Average	Mean intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *average* of these dissimilarities.
Centroid	Dissimilarity between the centroid for cluster A (a mean vector of length *p*) and the centroid for cluster B. Centroid linkage can result in undesirable *inversions*. → often used in genomics

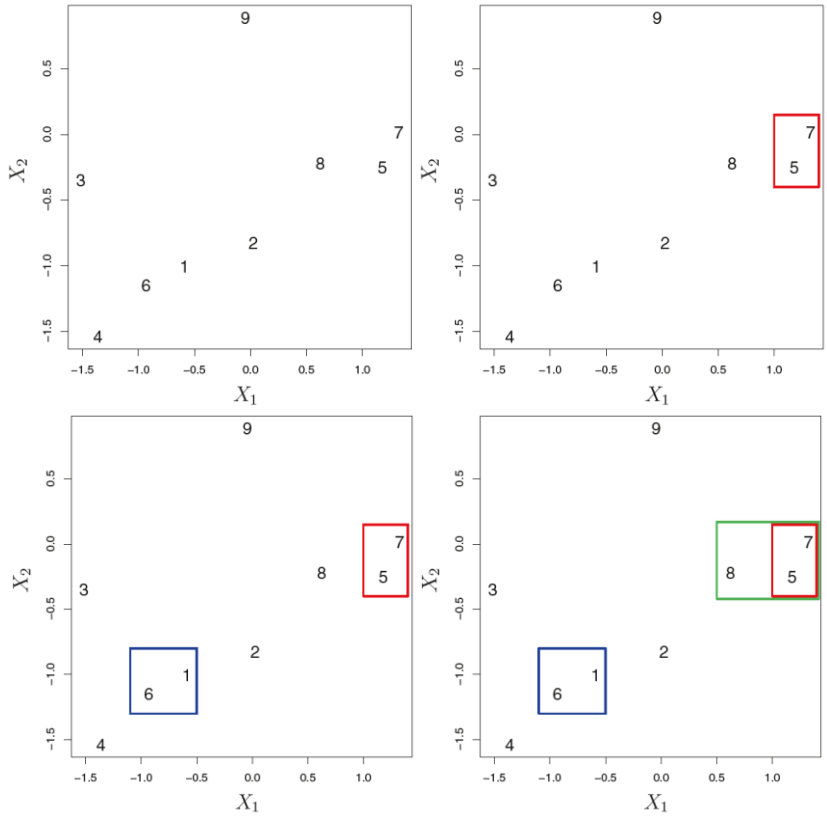
Average, complete, and single linkage are most popular among statisticians. In addition, average and complete linkage are generally preferred over single linkage, as they tend to yield more balanced dendrograms.

- Average, complete, and single linkage applied to an example data set.



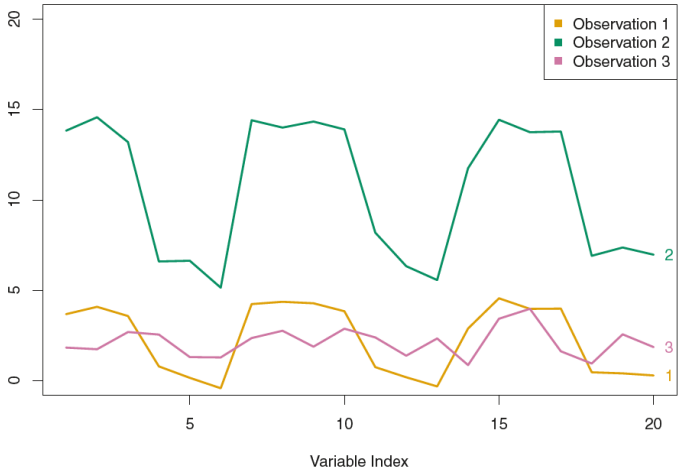
Instructor: Junho Song
junhosong@snu.ac.kr

- Example: hierarchical clustering with complete linkage



- Top Left: initially, there are nine distinct clusters, {1}, {2}, . . . , {9}.
- Top Right: the two clusters that are closest together, {5} and {7}, are fused into a single cluster.
- Bottom Left: the two clusters that are closest together, {6} and {1}, are fused into a single cluster.
- Bottom Right: the two clusters that are closest together using complete linkage, {8} and the cluster {5, 7}, are fused into a single cluster.

(c) Choice of Dissimilarity Measure: any alternative?



Instructor: Junho Song
junhosong@snu.ac.kr

- Correlation-based distance: the measure which focuses on the *shapes* of observation profiles rather than their magnitude

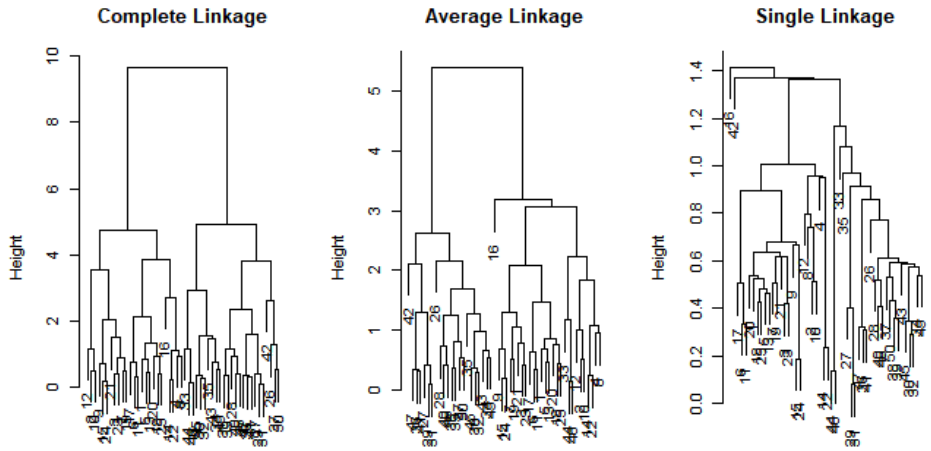
In the above figure, three observations with measurements on 20 variables are shown. Observations 1 and 3 have similar values for each variable and so there is a small Euclidean distance between them. But they are very weakly correlated, so they have a large correlation-based distance.

On the other hand, observations 1 and 2 have quite different values for each variable, and so there is a large Euclidean distance between them. But they are highly correlated, so there is a small correlation-based distance between them.

See textbook (pp. 396-399) regarding when we should use Euclidean distance or correlation-based distance.

(d) R example

```
hc.complete=hclust(dist(x), method="complete")  # same x with CM25's
hc.average=hclust(dist(x), method="average")
hc.single=hclust(dist(x), method="single")
par(mfrow=c(1,3))
plot(hc.complete, main="Complete Linkage", xlab="", sub="", cex=.9)
plot(hc.average, main="Average Linkage", xlab="", sub="", cex=.9)
plot(hc.single, main="Single Linkage", xlab="", sub="", cex=.9)

cutree(hc.complete, 2)  # To determine the cluster labels
cutree(hc.average, 2)
cutree(hc.single, 2)
cutree(hc.single, 4)

x=matrix(rnorm(30*3), ncol=3)  # correlation-based distance
dd=as.dist(1-cor(t(x)))
par(mfrow=c(1,1))
plot(hclust(dd, method="complete"), main="Complete Linkage with
   Correlation-Based Distance", xlab="", sub="")
```



Instructor: Junho Song
junhosong@snu.ac.kr

4. Practical Issues in Clustering

(a) Small decisions with big consequences

- Standardization of the observations or features

  e.g. Centered to have mean zero and scaled to have standard deviation one (`scale`() in R does both tasks)

- *K*-Means clustering

  The number of clusters → How many?

- Hierarchical clustering

  Dissimilarity measure → What measure?

  Linkage → What type?

  Cut the dendrogram → Where?

(b) Validating the clusters obtained

- Whether the clusters that have been found represent true subgroups in the data, or whether they are simply a result of clustering the noise.
- A number of techniques for assigning a p-value to a cluster in order to assess

  → But, no consensus on a single best approach.

(c) Other considerations in clustering

- *K*-means and hierarchical clustering force every observation into a cluster, the clusters found may be heavily distorted.

  → Because of the presence of outliers that do not belong to any cluster.

- Clustering methods generally are not very robust to perturbations to the data.