L7-1

Cluster Analysis/Clustering

Seungchan Kim
Center for Computational Systems Biology
Electrical and Computer Engineering

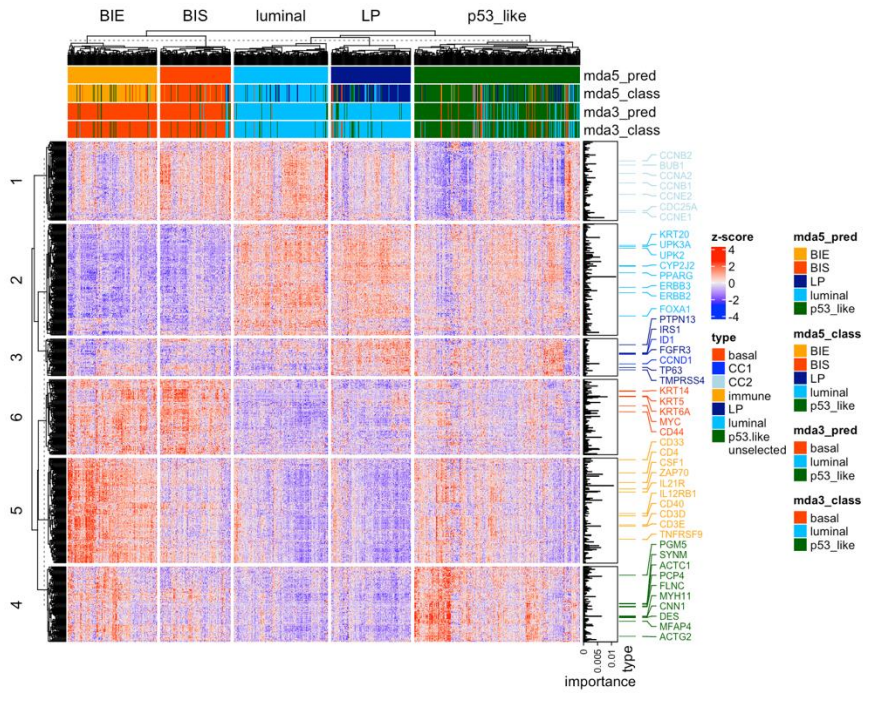
Cluster Analysis: Examples

- Cluster images to find categories
- Cluster patient data, i.e. RNAseq data, to find disease subtypes
- Cluster persons in social networks to detect communities
  - Amazon's suggesting items

Cluster Analysis: Examples

Clustering – PCA Plot

Clustering – Heatmap



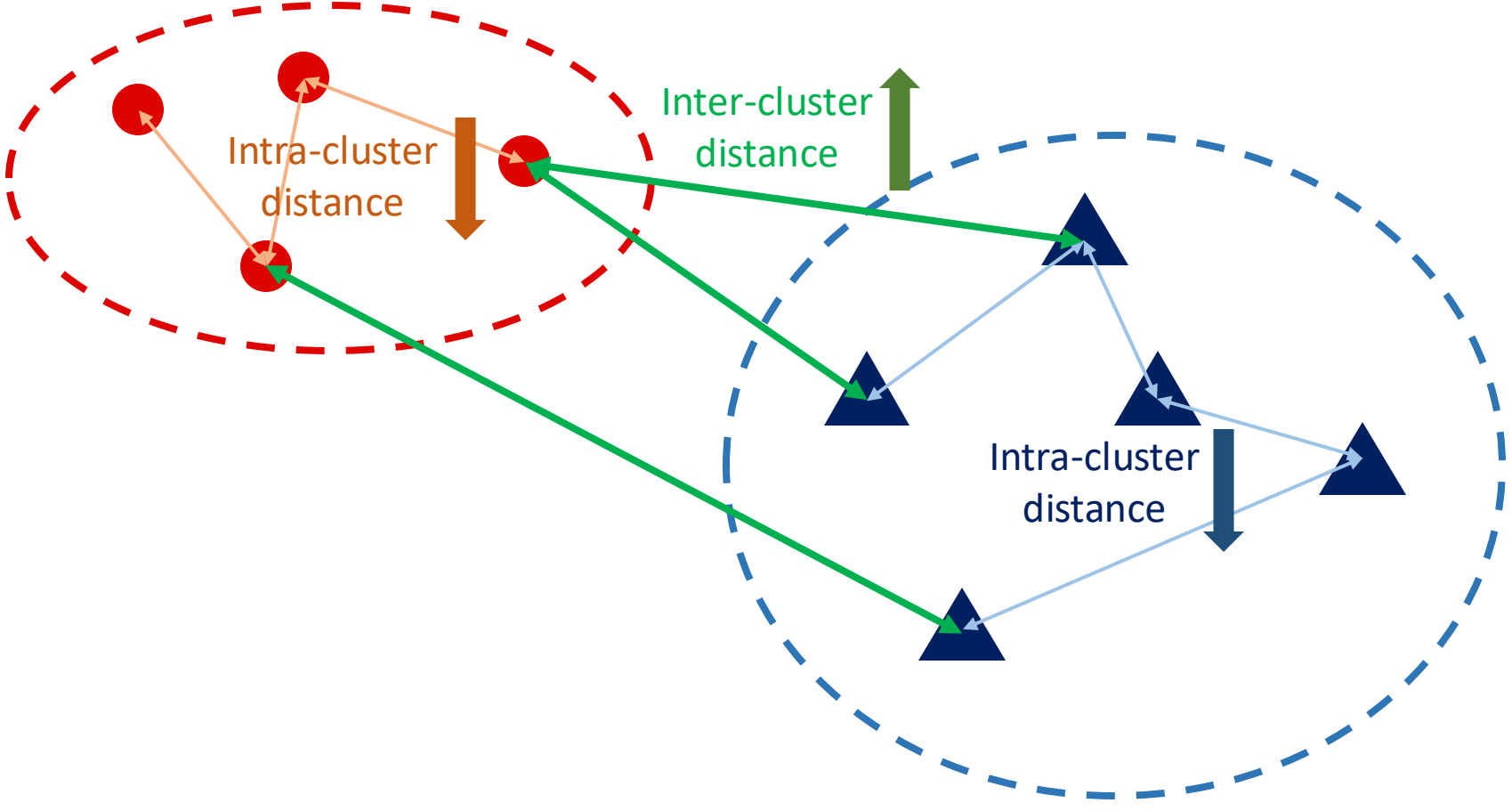
Cluster Analysis and Clusters

- Cluster analysis
  - Grouping a set of data objects into clusters

- Cluster: a collection of data objects
  - *Similar* to one another within the same cluster
  - *Dissimilar/different* from the objects in other clusters

  - High *Intra-cluster* similarity
  - Low *Inter-cluster* similarity

Clustering

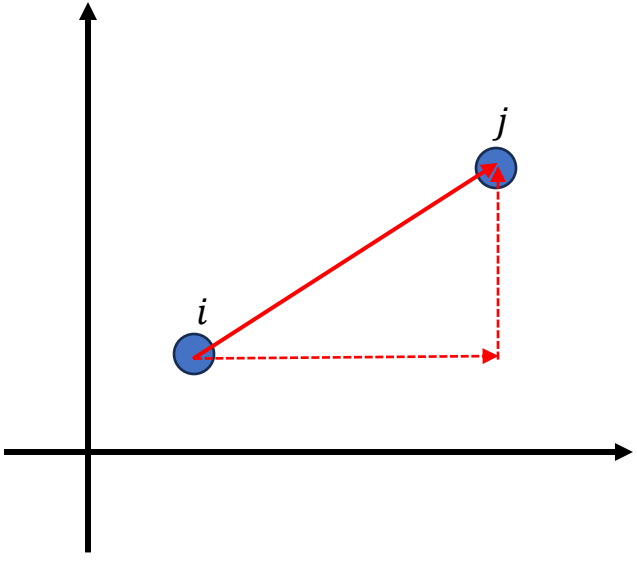

Intra-cluster distance
Inter-cluster distance
Intra-cluster distance

Distance/Similarity

- Dissimilarity/Similarity metric: Usually, points are in a high-dimensional space, and *similarity* is expressed in terms of a *distance function*, $d(i, j)$

- Euclidean distance:

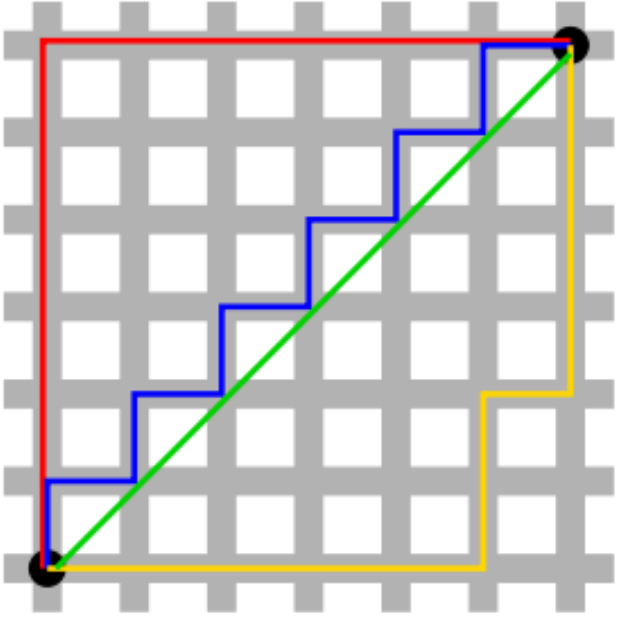
$$d(i,j) = \sqrt{\sum_k \left(x_{ik} - x_{jk}\right)^2}$$



Manhattan distance

- L_1 norm
- City block distance

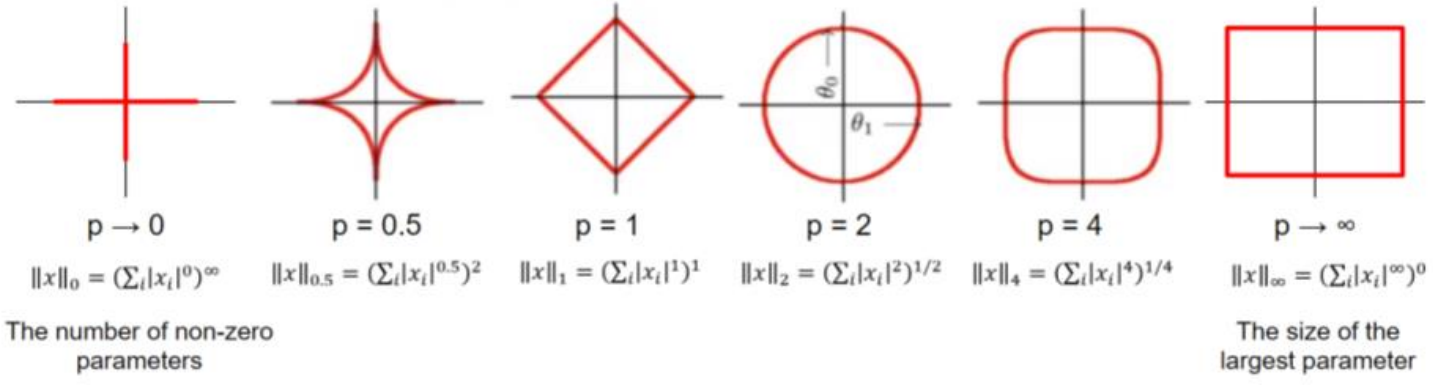
$$d(i,j) = \sum_k |x_{ik} - x_{jk}|$$



L_p Minkowski Distance

- A generalization of Euclidian distance

$$d(i,j) = \left[\sum_k |x_{ik} - x_{jk}|^p\right]^{1/p}$$



Metric Space: (*M*, *d*)

- Metric space is an ordered pair (*M*, *d*) where *M* is a set and *d* is a metric (distance) on *M*, i.e. a function:

$$d: M \times M \to \mathbb{R}$$

satisfying the following axioms:

$$d(x, x) = 0$$

$$d(x, y) > 0 \text{ if } x \neq y \text{ [positivity]}$$

$$d(x, y) = d(y, x) \text{ [symmetry]}$$

$$d(x, z) \leq d(x, y) + d(y, z) \text{ [triangular inequality]}$$

Other distance/similarity

- Correlation coefficients
  - Pearson
  - Spearman's rank correlation

- Jaccard Index - $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$

Other distance/similarity

- Entropy-based
  - *Kullback-Leibler* divergence, a.k.a. *relative entropy* - how one probability distribution, $P(x)$, is different from a second, $Q(x)$, reference probability distribution

$$D_{KL}(P||Q) = \sum_{x \in \mathrm{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right) = -\sum_{x \in \mathrm{X}} P(x)[\log Q(x) - \log P(x)]$$

  - *Jensen-Shannon* divergence - the similarity between two probability distributions; a symmetrized and smoothed version of KL divergence

$$JSD(P||Q) = \frac{1}{2} D(P||M) + \frac{1}{2} D(Q||M)$$

where $M = \frac{1}{2}(P + Q)$

Clustering Approaches

- **Partitioning algorithms**: Construct various partitions and then evaluate them by some criterion
- **Hierarchy algorithms**: Create a hierarchical decomposition of the set of data (or objects) using some criterion
- **Density-based**: based on connectivity and density functions
- **Model-based**: A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other
- **Grid-based**: based on a multiple-level granularity structure

k-means Clustering

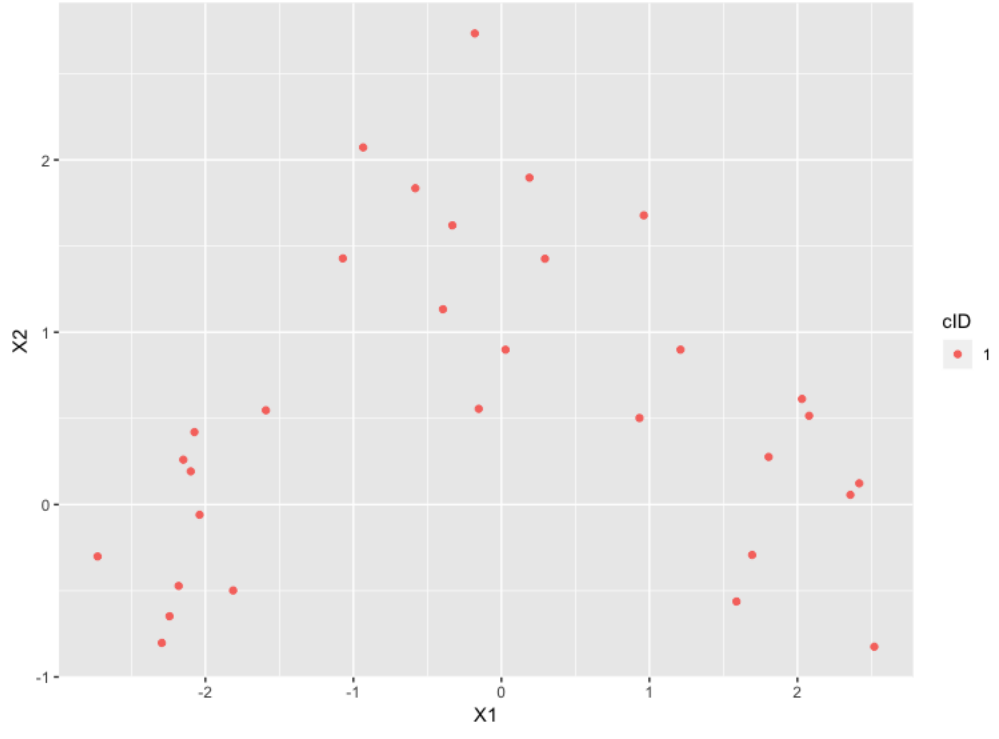
- Start by
  - picking *k*, the number of clusters, and
  - initializing clusters by
    1. picking *k* random centroids, or
    2. randomly assigning each sample to one of *k* clusters and computing centroids
- Reassign samples to the closest centroids
- Repeat computing centroids and reassignment of samples, until convergence.

k-means Clustering

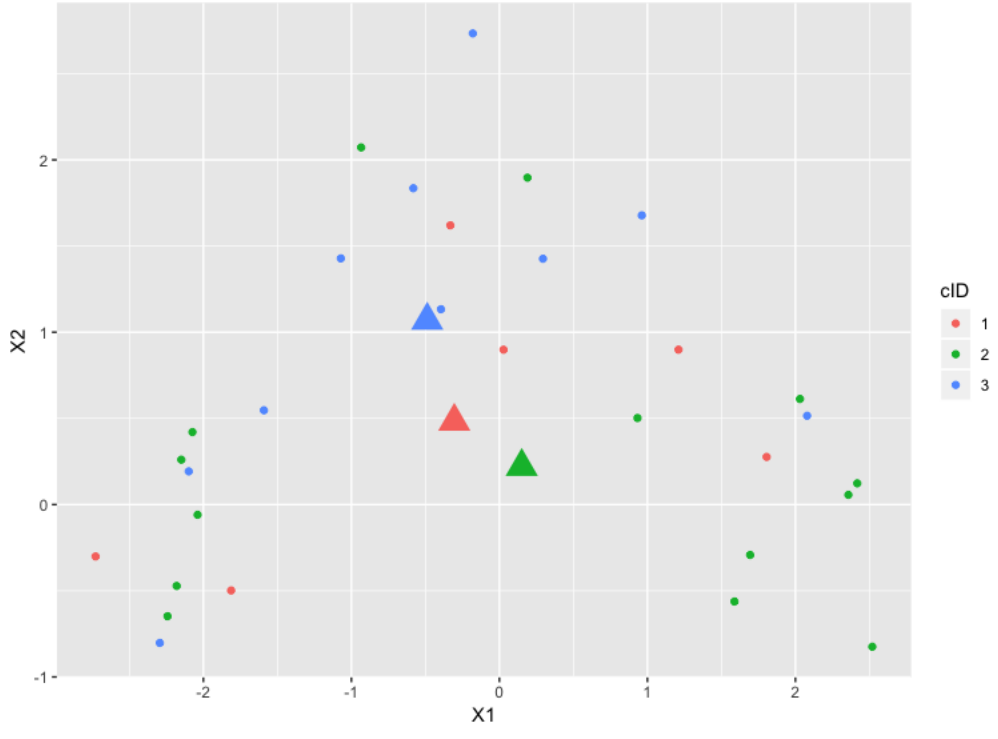
- **Algorithm**

for $k = 1, \ldots, K$, let $\mathbf{r}(k)$ be a randomly chosen point from D;
while changes in clusters C_k happen do
form clusters:
for $k = 1, \ldots, K$ do
$C_k = \{x \in D | d(\mathbf{r}_k, x) \leq d(\mathbf{r}_j, x)\}$ for all $j = 1, \ldots, K, j \neq K\}$
end;
compute new cluster centers;
for $k = 1, \ldots, K$ do
$\mathbf{r}_k =$ the vector mean of the points in C_k
end;
end;

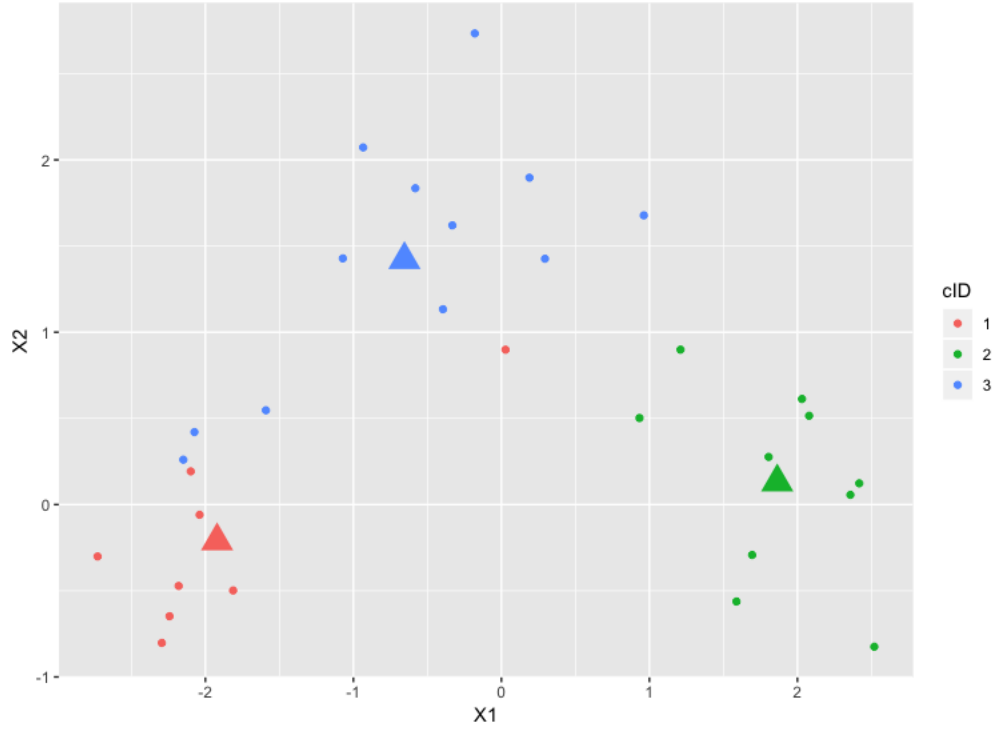
k-means Clustering: example



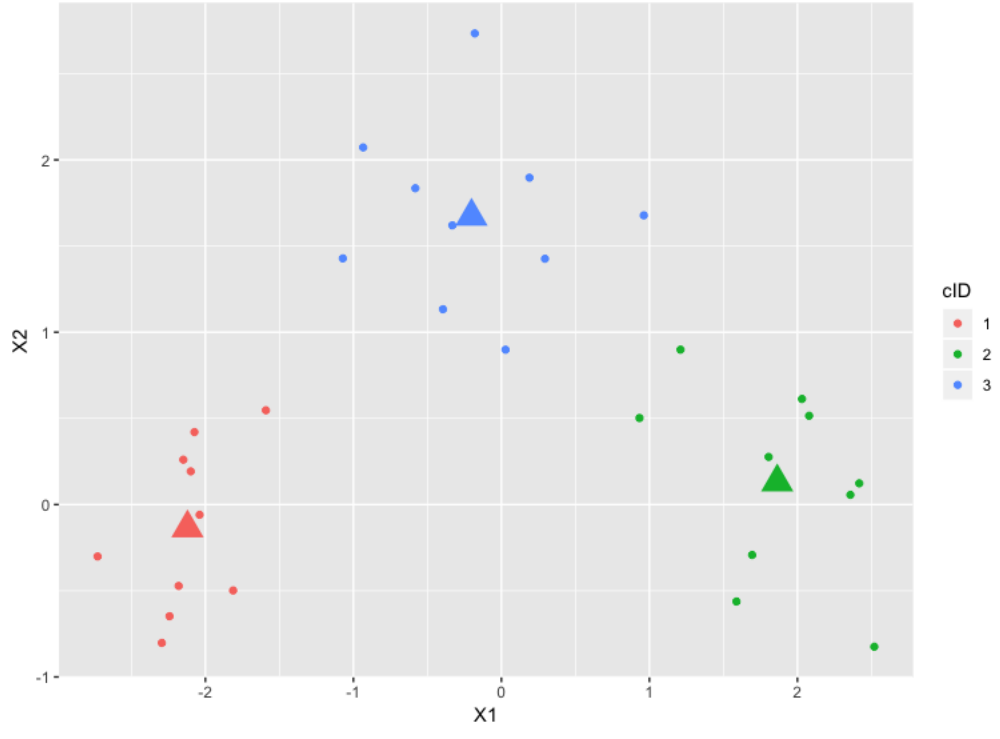
k-means Clustering: example



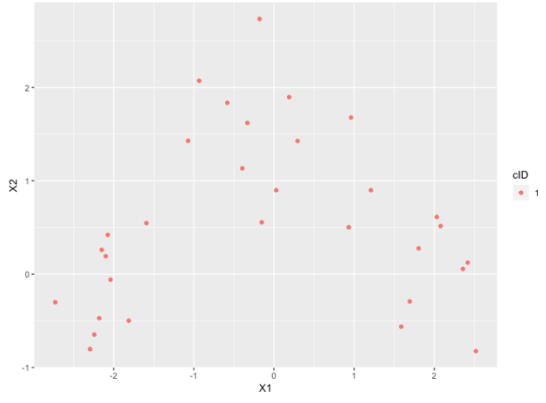
k-means Clustering: example

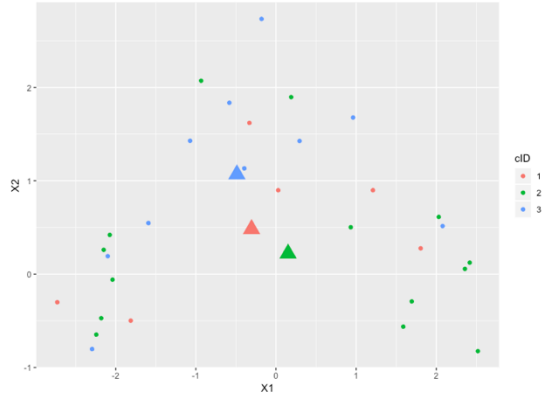


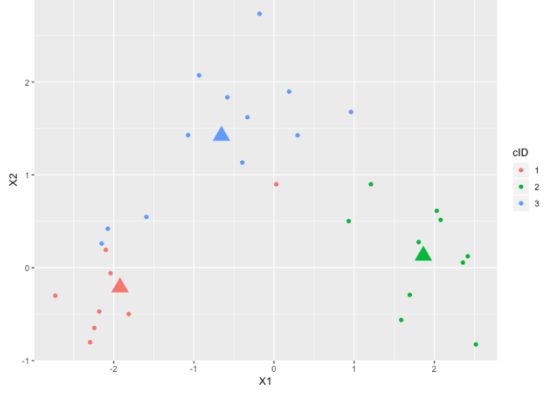
k-means Clustering: example

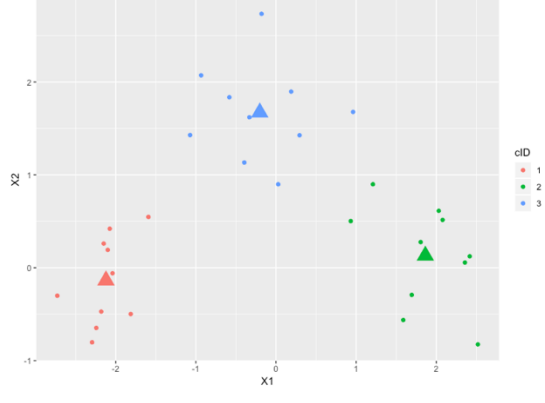


k-means Clustering: example









Comments on K-means Clustering

- **Strength**: *Relatively efficient*: O(*tkn*), where *n* is # objects, *k* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.
- **Comment**: Often terminates at a *local optimum*. The global optimum may be found using techniques such as: *deterministic annealing, genetic algorithms* or *resampling*
- **Weakness**
  - Applicable only when *mean* can be defined, then what about categorical data?
  - Need to specify *k*, the number of clusters, in advance
  - Unable to handle noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex* shapes

K-medoids Clustering

- Instead of the mean of points to define the center of a cluster, k-medoids uses actual data points—called medoids—as the cluster centers.
- **PAM** (Partitioning Around Medoids, 1987)
  - starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clusters
  - PAM works effectively for small data sets, but does not scale well for large data sets
- **CLARA** (Kaufmann & Rousseeuw, 1990)
- **CLARANS** (Ng & Han, 1994): Randomized sampling

PAM (Partitioning Around Medoids)

Use real object to represent the cluster

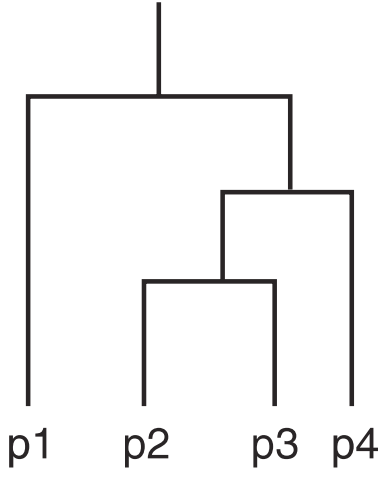
1. Select *k* representative objects arbitrarily
2. For each pair of non-selected object *h* and selected object *i*, calculate the total swapping cost TC_{ih}
3. For each pair of *i* and *h*,
   - If $TC_{ih} < 0$, *i* is replaced by *h*
   - Then assign each non-selected object to the most similar representative object
4. repeat steps 2-3 until there is no change

Hierarchical Clustering

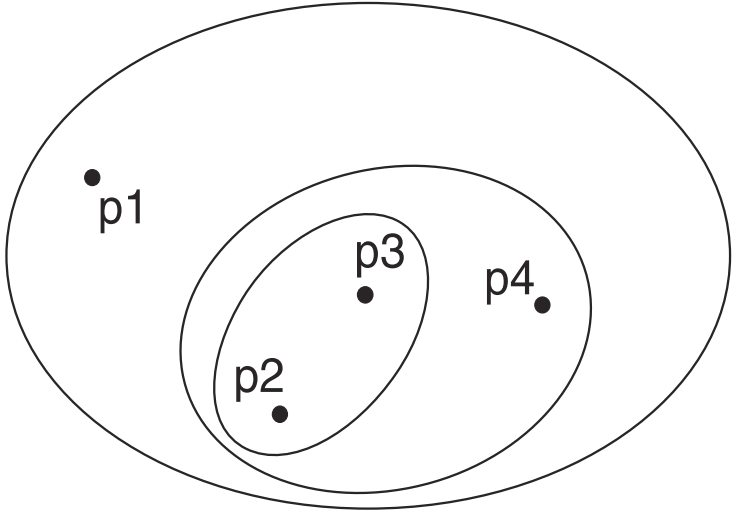
- **Agglomerative**: Start with the points as individual clusters and, at each step, merge the closest pair of clusters. This requires defining a notion of cluster proximity – **the most common hierarchical clustering**
- **Divisive**: Start with one, all-inclusive cluster and, at each step, split a cluster until only singleton clusters of individual points remain. In this case, we need to decide which cluster to split at each step and how to do the splitting.

- **Dendrogram**: A tree-like diagram that displays both the cluster-subcluster relationships and the order in which the clusters were merged (agglomerative view)

Dendrogram



(a) Dendrogram.



(b) Nested cluster diagram.

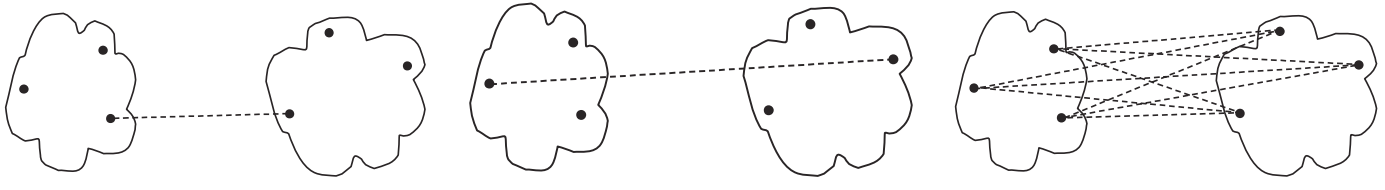
Agglomerative Hierarchical Clustering

Algorithm

1. Let each data point/sample be a cluster C_i
2. Compute the distance/similarity matrix, $D = [d_{ij}]$, if necessary:
$$d_{ij} = d(C_i, C_j)$$
3. **repeat**
    1. Merge the closest two clusters and form a new cluster, and remove the two clusters
    2. Update the proximity matrix to reflect the proximity between the new cluster and the original clusters
4. **until** only **one** cluster remains.

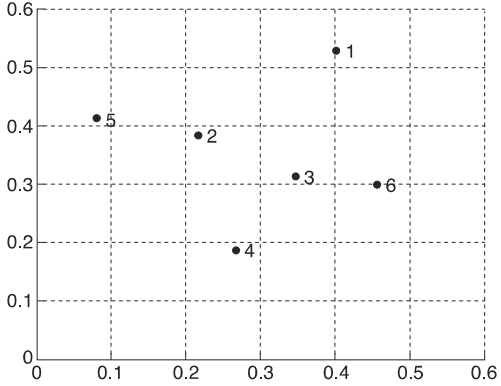
Distance between Clusters

- **Single linkage**: shortest distance
- **Complete linkage**: longest distance
- **Average linkage**: average distance across all members



(a) MIN (single link.) (b) MAX (complete link.) (c) Group average.

Example



Point	x Coordinate	y Coordinate
p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Single Linkage

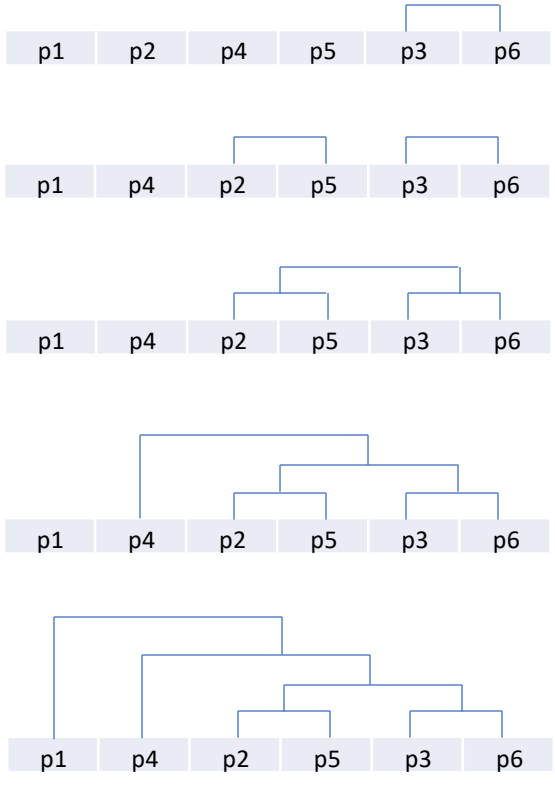
	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	**0.11**
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

	p1	p2	p4	p5	p36
p1	0.00	0.24	0.37	0.34	0.22
p2	0.24	0.00	0.20	**0.14**	0.15
p4	0.37	0.20	0.00	0.29	0.15
p5	0.34	0.14	0.29	0.00	0.28
p36	0.22	0.15	0.15	0.28	0.00

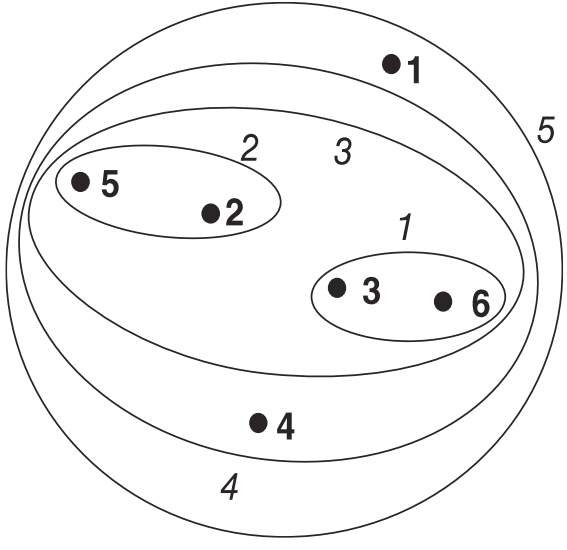
	p1	p4	p36	p25
p1	0.00	0.37	0.22	0.24
p4	0.37	0.00	0.15	0.20
p36	0.22	0.15	0.00	**0.15**
p25	0.24	0.20	0.15	0.00

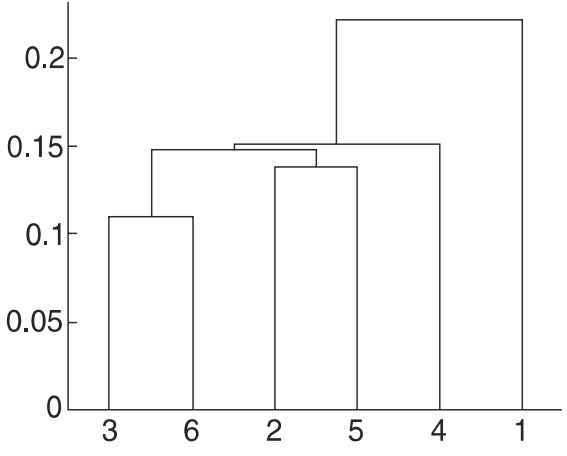
	p1	p4	p25-36
p1	0.00	0.37	0.22
p4	0.37	0.00	**0.15**
p25-36	0.22	0.15	0.00

	p1	p25-36-4
p1	0.00	0.22
p25-36-4	0.22	0.00

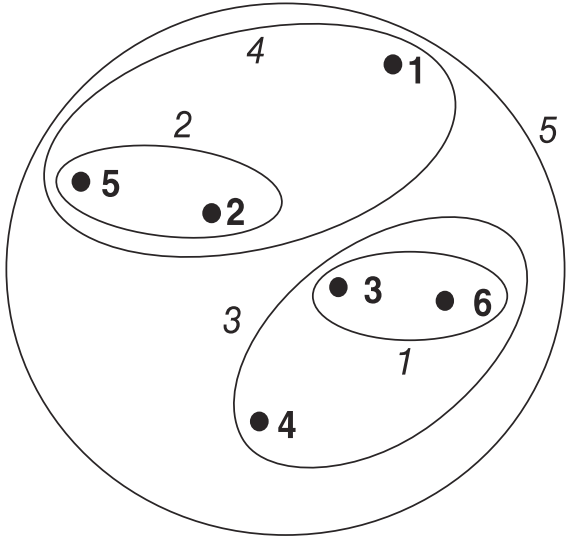


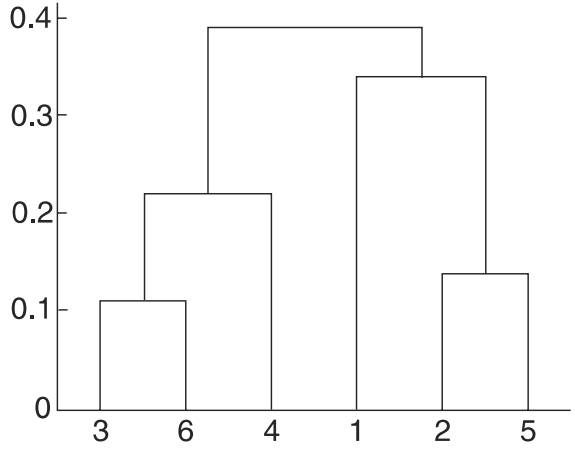
Single Linkage



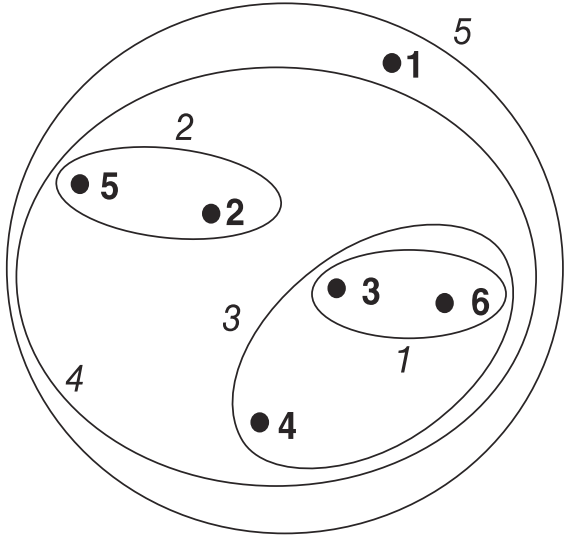


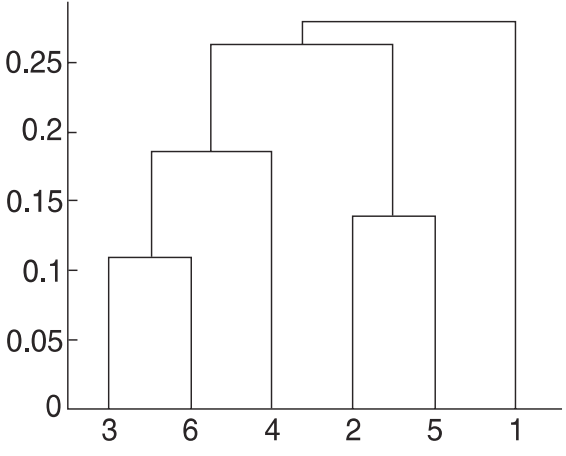
Complete Linkage





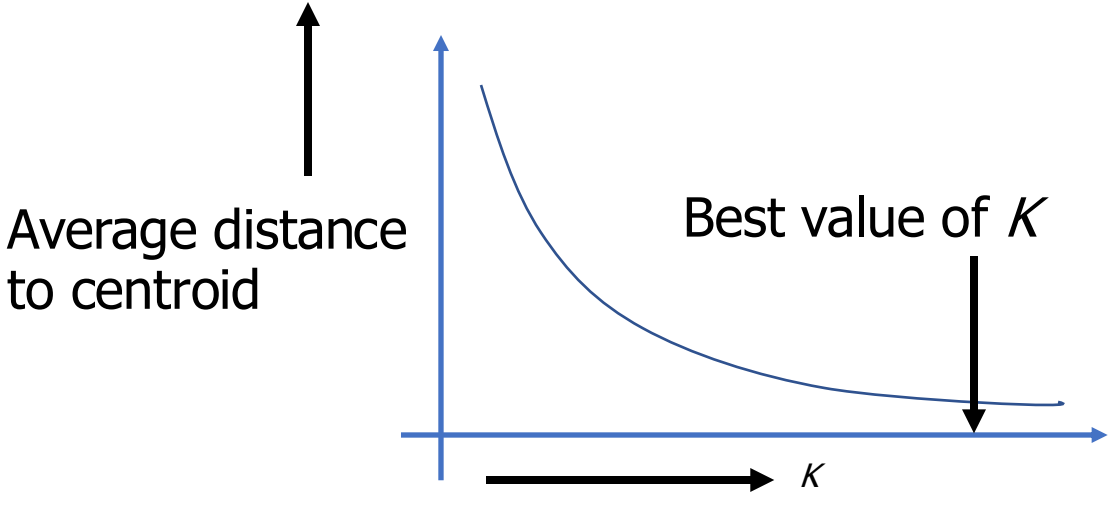
Average Linkage





Getting right *K*

- Try different *K*, looking at the change in the average distance to centroid, as *K* increases.
- Average falls rapidly until right *K*, then changes little.



Final Thoughts & Takeaways

- Clustering is a powerful tool for uncovering hidden patterns in data.
- Hierarchical methods offer flexibility and intuitive visualizations through dendrograms.
- Linkage choices (single, complete, average) impact cluster shape and interpretation.
- Always consider the context and goals of your analysis when choosing clustering methods.

Reading

- **R for Data Science**
  - http://r4ds.had.co.nz/index.html
- **Introduction to Statistical Learning with Applications in R**
  - http://www-bcf.usc.edu/~gareth/ISL/
- **Cluster Analysis: Basic Concepts and Algorithms:**
  - https://www-users.cs.umn.edu/~kumar001/dmbook/ch8.pdf
- Monti, S., Tamayo, P., Mesirov, J., Golub, T. (2003) **Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data**. *Machine Learning*, 52, 91–118.
  - https://link.springer.com/article/10.1023/A:1023949509487
- **ConsensusClusterPlus**:
  - https://bioconductor.org/packages/release/bioc/html/ConsensusClusterPlus.html