CS699
Lecture 10
Clustering

What is Cluster Analysis?

- Cluster: A collection of data objects
  - similar (or related) to one another within the same group
  - dissimilar (or unrelated) to the objects in other groups
- Cluster analysis (or *clustering*, *data segmentation, …*)
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- Unsupervised learning: no predefined classes (i.e., *learning by observations* vs. learning by examples: supervised)
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

Clustering for Data Understanding and Applications

- Biology: taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- Information retrieval: document clustering
- Land use: Identification of areas of similar land use in an earth observation database
- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults
- Climate: understanding earth climate, find patterns of atmospheric and ocean
- Economic Science: market resarch

Clustering as a Preprocessing Tool (Utility)

- Summarization:
  - Preprocessing for regression, PCA, classification, and association analysis
- Compression:
  - Image processing: vector quantization
- Finding K-nearest Neighbors
  - Localizing search to one or a small number of clusters
- Outlier detection
  - Outliers are often viewed as those “far away” from any cluster

Quality: What Is Good Clustering?

- A good clustering method will produce high quality clusters
  - high intra-class similarity: cohesive within clusters
  - low inter-class similarity: distinctive between clusters
- The quality of a clustering method depends on
  - the similarity measure used by the method
  - its implementation, and
  - Its ability to discover some or all of the hidden patterns

Measure the Quality of Clustering

- Dissimilarity/Similarity metric
  - Similarity is expressed in terms of a distance function, typically metric: $d(i, j)$
  - The definitions of distance functions are usually different for interval-scaled, Boolean, categorical, ordinal, and vector variables
  - Weights should be associated with different variables based on applications and data semantics
- Quality of clustering:
  - There is usually a separate "quality" function that measures the "goodness" of a cluster.
  - It is hard to define "similar enough" or "good enough"
    - The answer is typically highly subjective

Considerations for Cluster Analysis

- Partitioning criteria
  - Single level vs. hierarchical partitioning (often, multi-level hierarchical partitioning is desirable)
- Separation of clusters
  - Exclusive (e.g., one customer belongs to only one region) vs. non-exclusive (e.g., one document may belong to more than one class)
- Similarity measure
  - Distance-based (e.g., Euclidian, road network, vector) vs. connectivity-based (e.g., density or contiguity)
- Clustering space
  - Full space (often when low dimensional) vs. subspaces (often in high-dimensional clustering)

Requirements and Challenges

- Scalability
  - Clustering all the data instead of only on samples
- Ability to deal with different types of attributes
  - Numerical, binary, categorical, ordinal, linked, and mixture of these
- Constraint-based clustering
  - User may give inputs on constraints
  - Use domain knowledge to determine input parameters
- Interpretability and usability
- Others
  - Discovery of clusters with arbitrary shape
  - Ability to deal with noisy data
  - Incremental clustering and insensitivity to input order
  - High dimensionality

Major Clustering Approaches

- Partitioning approach:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical methods: k-means, k-medoids, CLARANS
- Hierarchical approach:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, CAMELEON
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSACN, OPTICS, DenClue

Major Clustering Approaches

- Model-based
- Grid-based
- Frequent pattern-based
- User-guided or constraint-based
- Link-based clustering

Partitioning Algorithms: Basic Concept

- Partitioning method: Partitions a database ***D*** of ***n*** objects into a set of ***k*** clusters, such that the sum of squared distances is minimized (where *p* is an object and c_i is the centroid or medoid of cluster C_i).

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} (p - c_i)^2$$

- This is also called *within-cluster variation* or *SSE* (*sum of squared errors*).

Partitioning Algorithms: Basic Concept

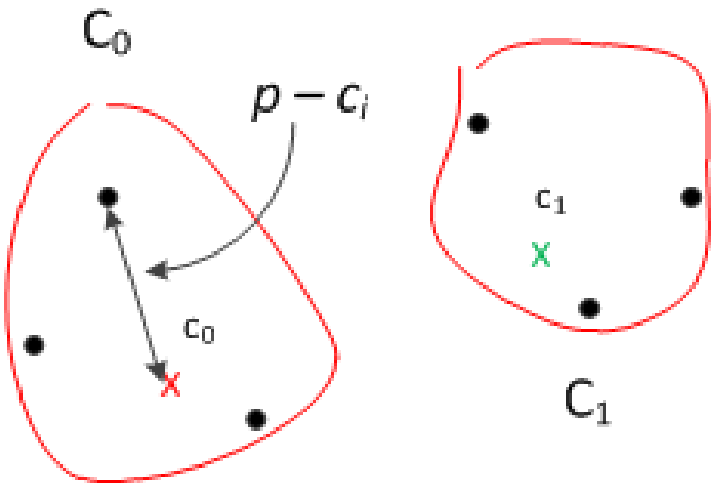
- More about *SSE*:

$$E = \Sigma_{i=1}^{k} \Sigma_{p \in C_i} (p - c_i)^2$$

sum for all clusters

squared distance between an object and the centroid of its cluster

sum for all objects in a cluster



Partitioning Algorithms: Basic Concept

- Given *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means* (MacQueen'67, Lloyd'57/'82): Each cluster is represented by the center of the cluster
  - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

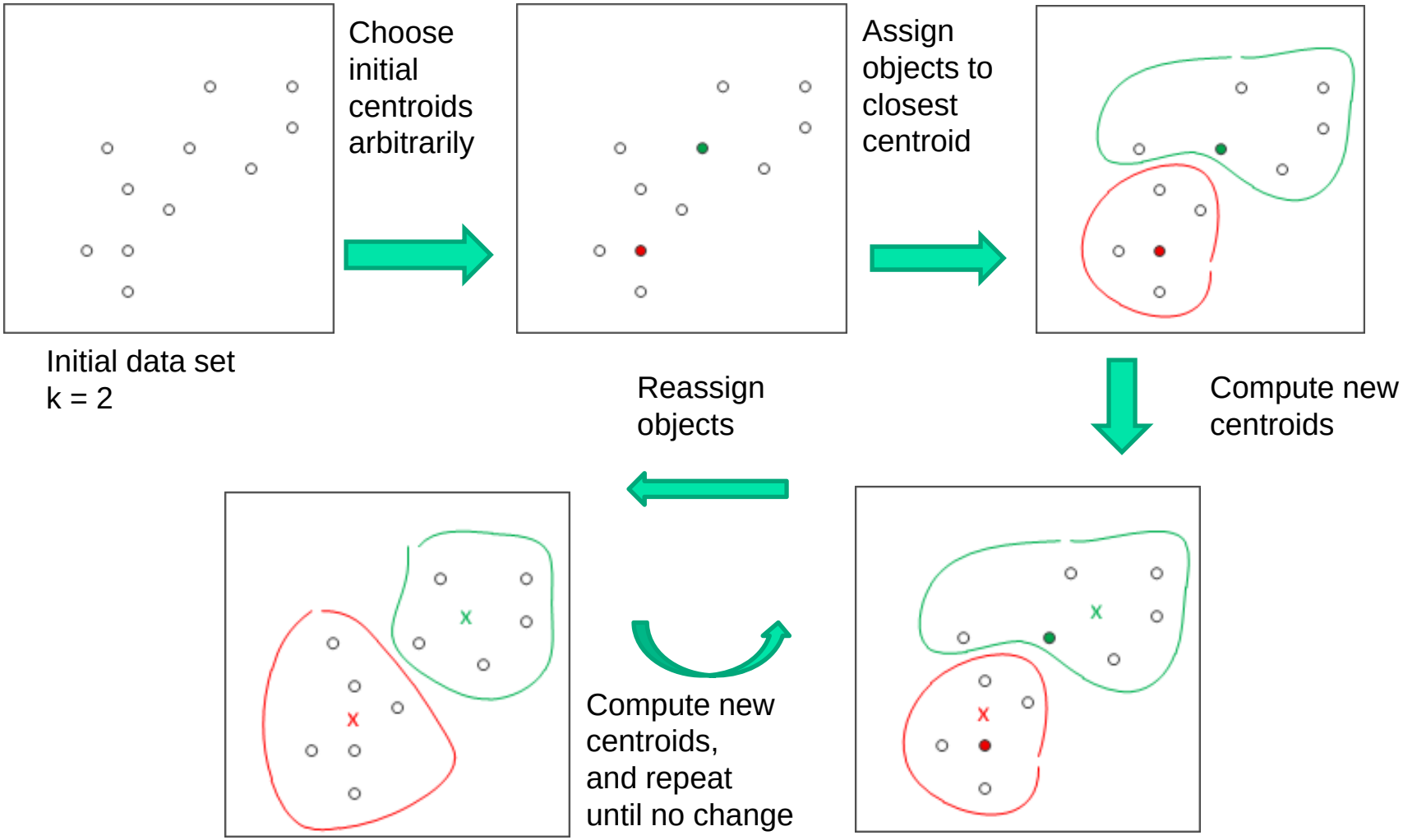
The *K-Means* Clustering Method

- Given *k*, the *k-means* algorithm works as follows:
  1. Arbitrarily choose *k* points as initial centroids (the centroid is the center, i.e., *mean point*, of the cluster). Each centroid represents a cluster.
  2. Assign each object to the cluster with the nearest centroid.
  3. Compute new centroids.
  4. Go back to Step 2. Stop when the membership assignment does not change or other criterion is met.

The *K-Means* Clustering Method

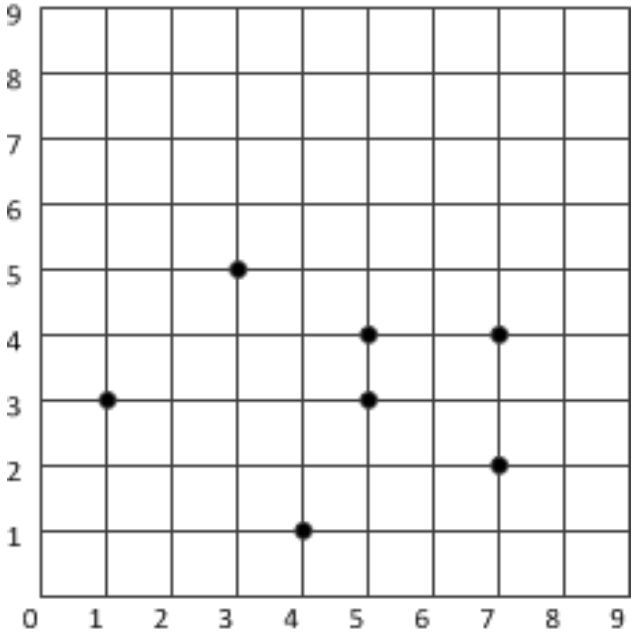
- Other stopping criteria
  - After each reassignment, E is computed and if E falls below a predefined threshold.
  - If the decrease in E, between two consecutive iterations, falls below a predefined threshold.
  - Run for a predetermined number of iterations (e.g., run 20 iterations).

Outline of *K-Means*

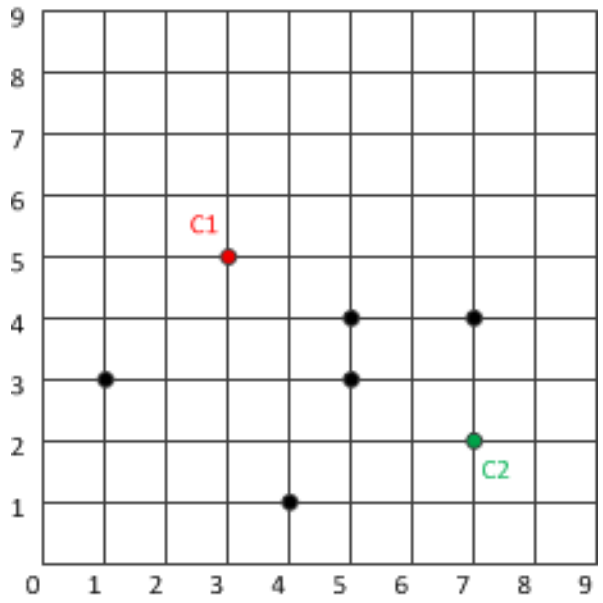


K-Means Illustration

Initial dataset, D = {(1,3), (3,5), (4,1), (5, 3), (5, 4), (7, 2), (7, 4)}

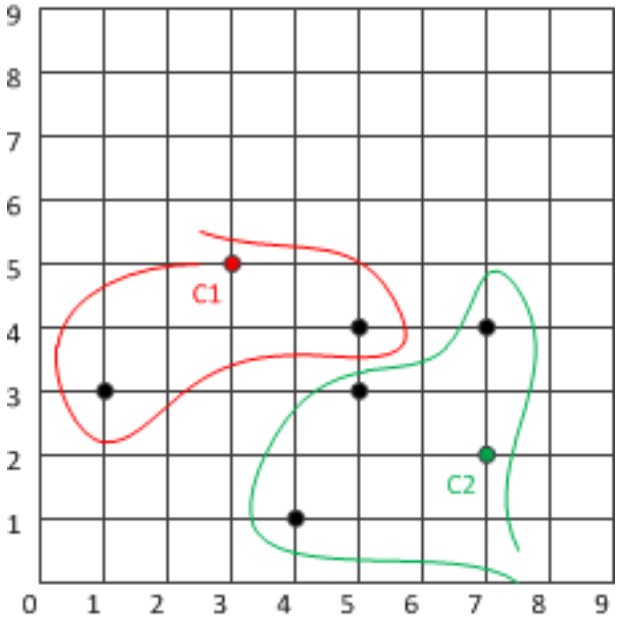


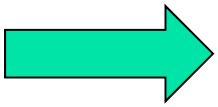
Initial dataset

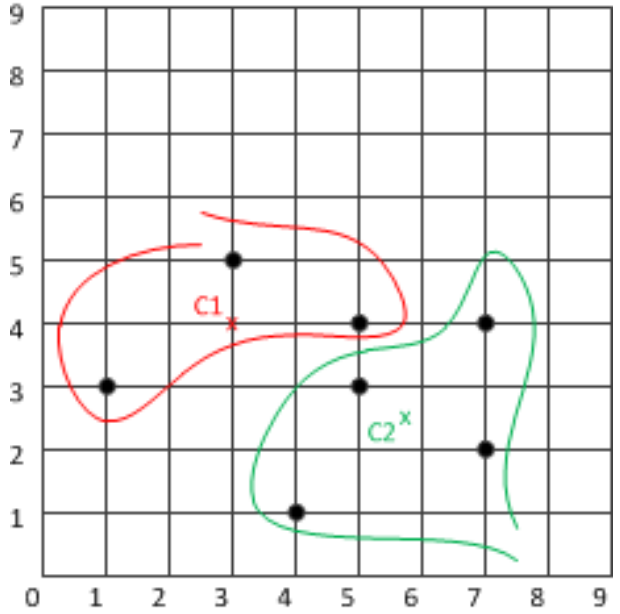


Two objects are randomly chosen as initial centroids

K-Means Illustration



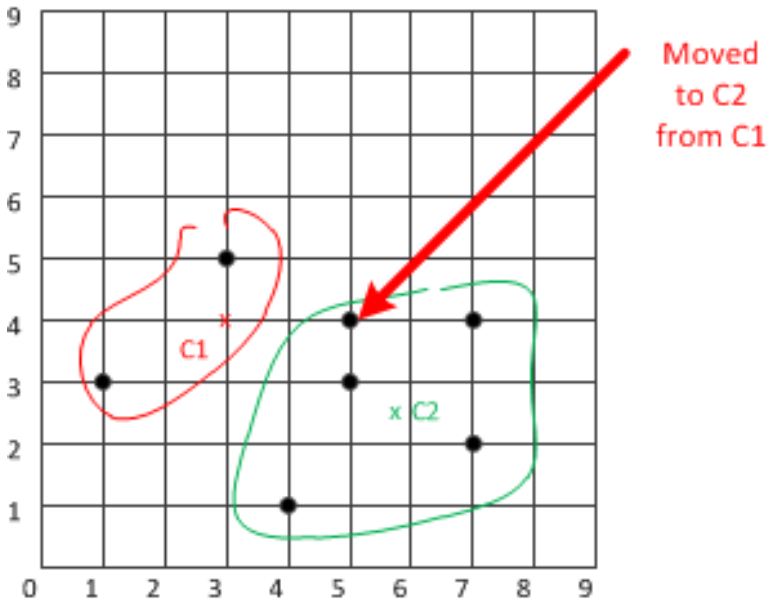


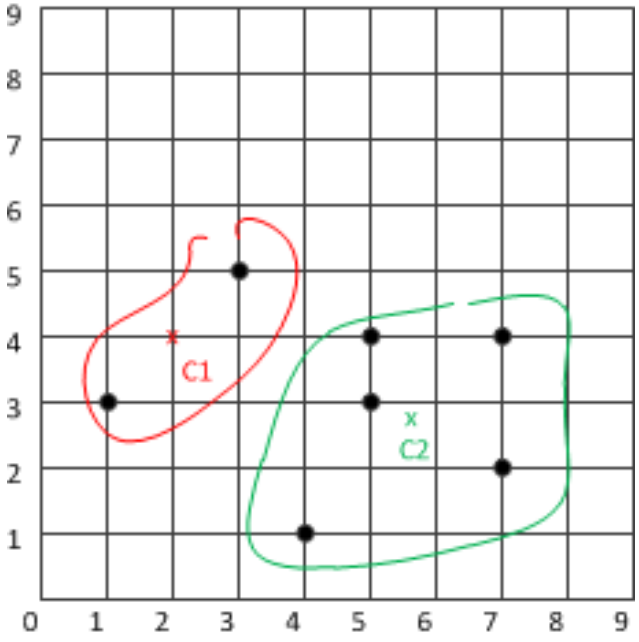


Objects are assigned to the cluster with the closest centroid

New centroids are computed.
C1.x = (1+3+5)/3 = 3
C1.y = (3+4+5)/3 = 4
C2.x = (4+5+7+7)/4= 5.75
C2.y = (1+2+3+4)/4 = 2.5

K-Means Illustration

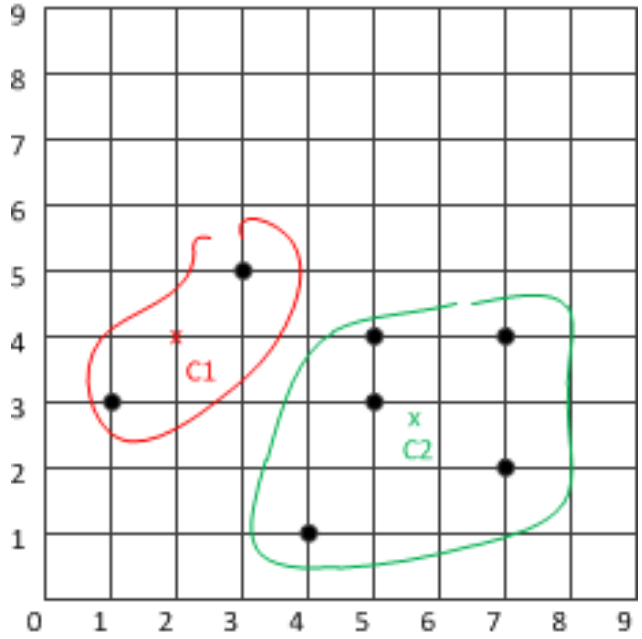




Objects are reassigned based on the distances to new centroids. Note that object (5,4) moved to cluster of C2.

New centroids are computed.
C1.x = (1+3)/2 = 2
C1.y = (3+5)/2 = 4
C2.x = (4+5+5+7+7)/5= 5.6
C2.y = (1+2+3+4+4)/5 = 2.8

K-Means Illustration



Objects are reassigned based on the distances to new centroids.

There is no membership change.

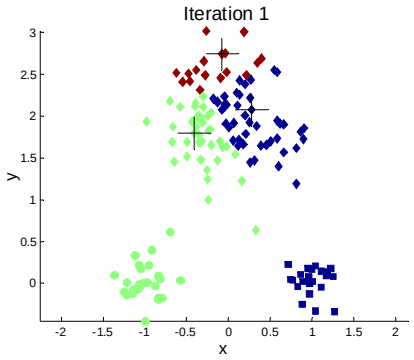
So, stop here.

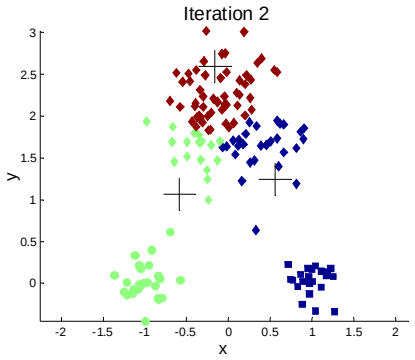
Comments on the *K-Means* Method

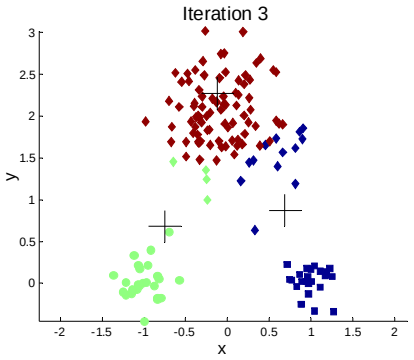
- Strength: *Efficient*:
- Weakness
  - Initial random selection of centroids affects the results (i.e., may not converge or may end up with a local optimum)
    - Run k-means multiple times with different initial centroids
  - Applicable only when the mean of objects can be defined
    - Use the k-modes method for categorical data
  - Need to specify *k,* the *number* of clusters, in advance
  - Sensitive to noisy data and *outliers*
  - Not suitable to discover clusters with *arbitrary shapes*

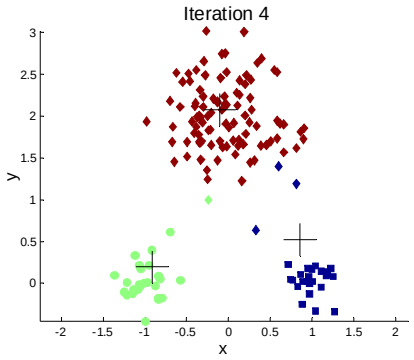
Comments on the *K-Means* Method

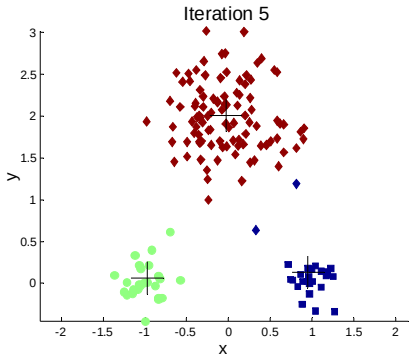
- Weakness (continued)
  - Selection of initial centroids (good choice)

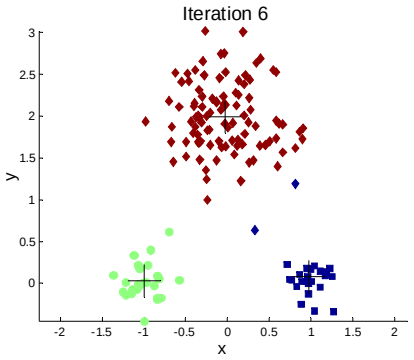






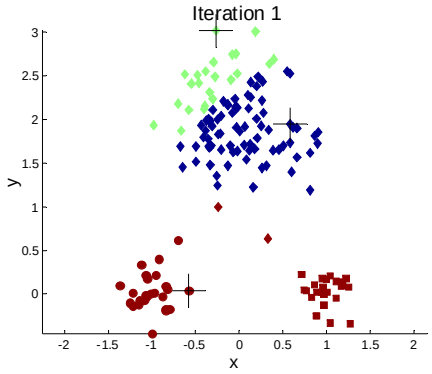


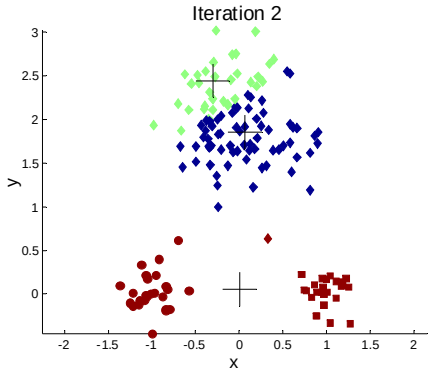


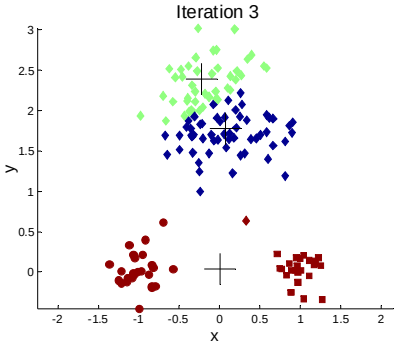


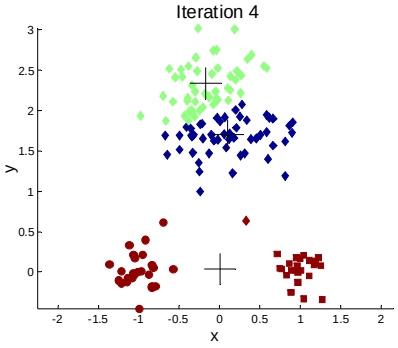
Comments on the *K-Means* Method

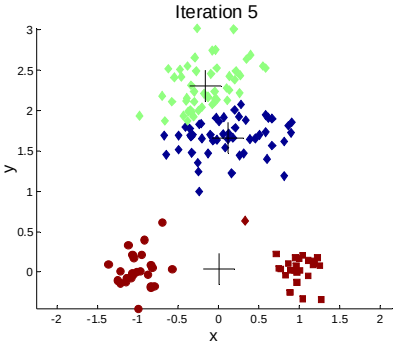
- Weakness (continued)
  - Selection of initial centroids (bad choice)











Comments on the *K-Means* Method

- Weakness (continued)
  - Sensitive to noisy data and *outliers*
    - Consider one-dimensional objects: {1, 2, 3, 8, 9, 10, 25}
    - Reasonable clustering:

      Two clusters {1, 2, 3,} and {8, 9, 10}, and an outlier 25.

    - Run k-means with $k = 2$:

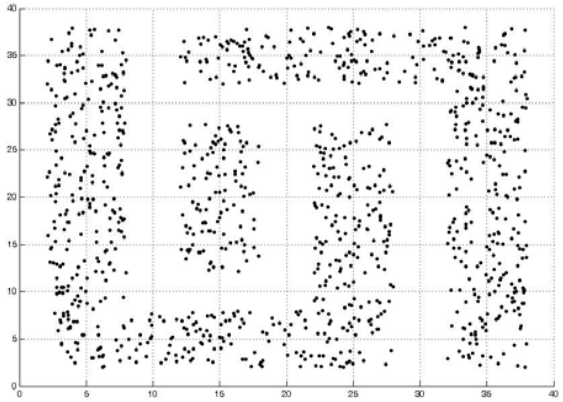
      Two clusters {1, 2, 3, 8} and {9, 10, 25}

    - K-medoids method is more robust in the presence of noise/outliers

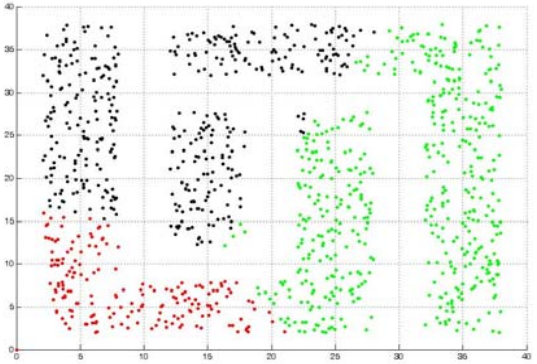
Comments on the *K-Means* Method

- Weakness (continued)
  - Not suitable to discover clusters with *arbitrary shapes*

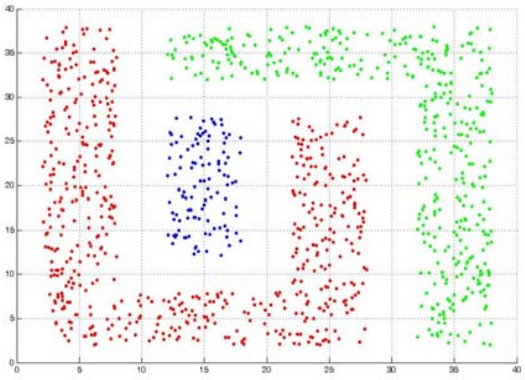
Initial dataset



K-means with $k = 3$

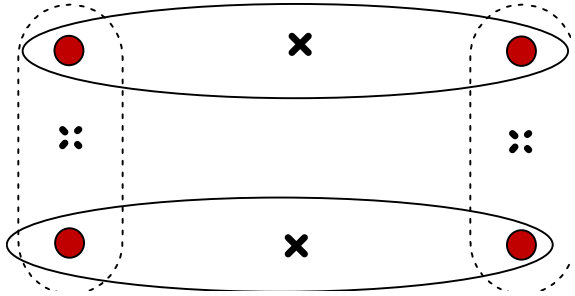


Natural clustering
This is an output of DBSCAN algorithm



Variations of the *K-Means* Method

- Most of the variants of the *k-means* differ in
  - Selection of the initial *centroids*
  - Dissimilarity calculations
  - Strategies to calculate cluster means



Variations of the *K-Means* Method

- Bisecting k-means

  1. Initially a single cluster includes all objects.
  2. The cluster is partitioned into two clusters using K-means

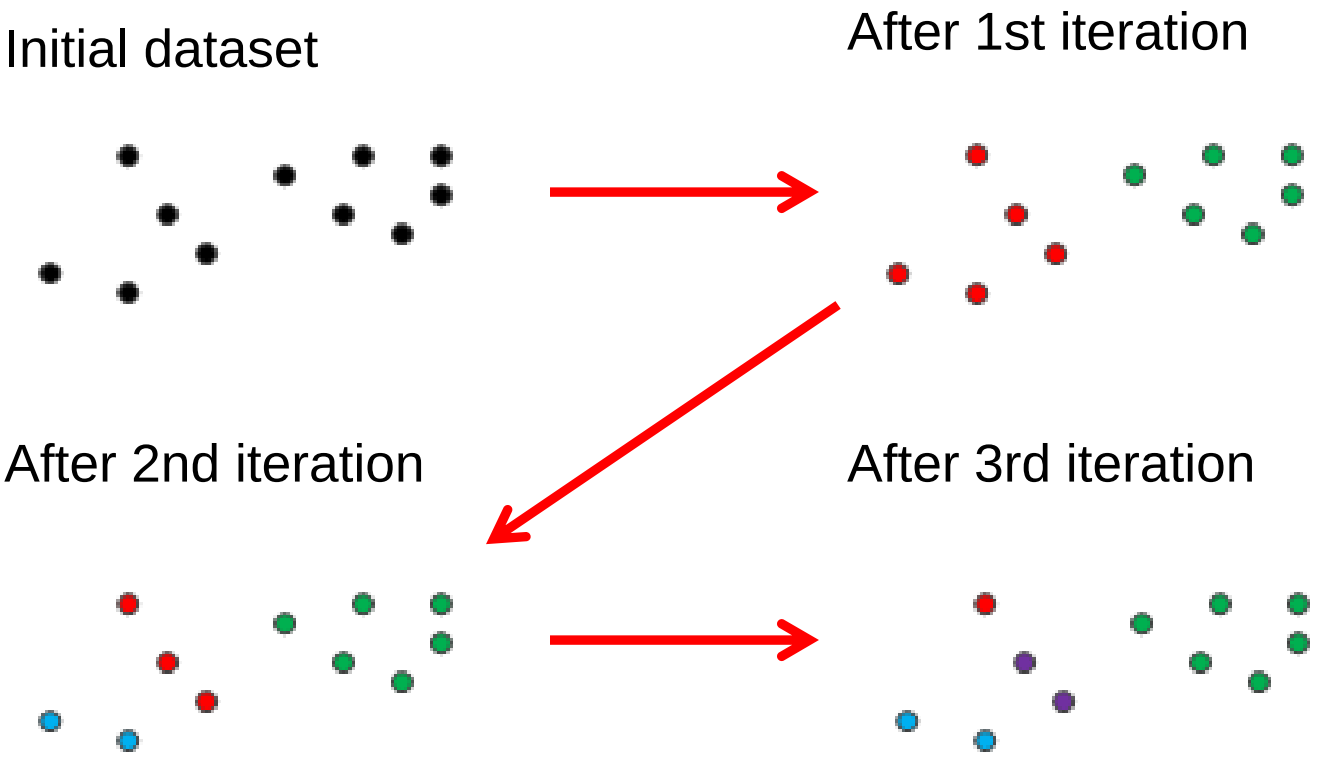
  (This can be repeated multiple times and the one with the smallest SSE can be chosen)

  3. A cluster is selected (based on certain criterion), and go to step 2
  4. Stop when we have *k* clusters

  (becomes a hierarchical clustering – refer to later slides)

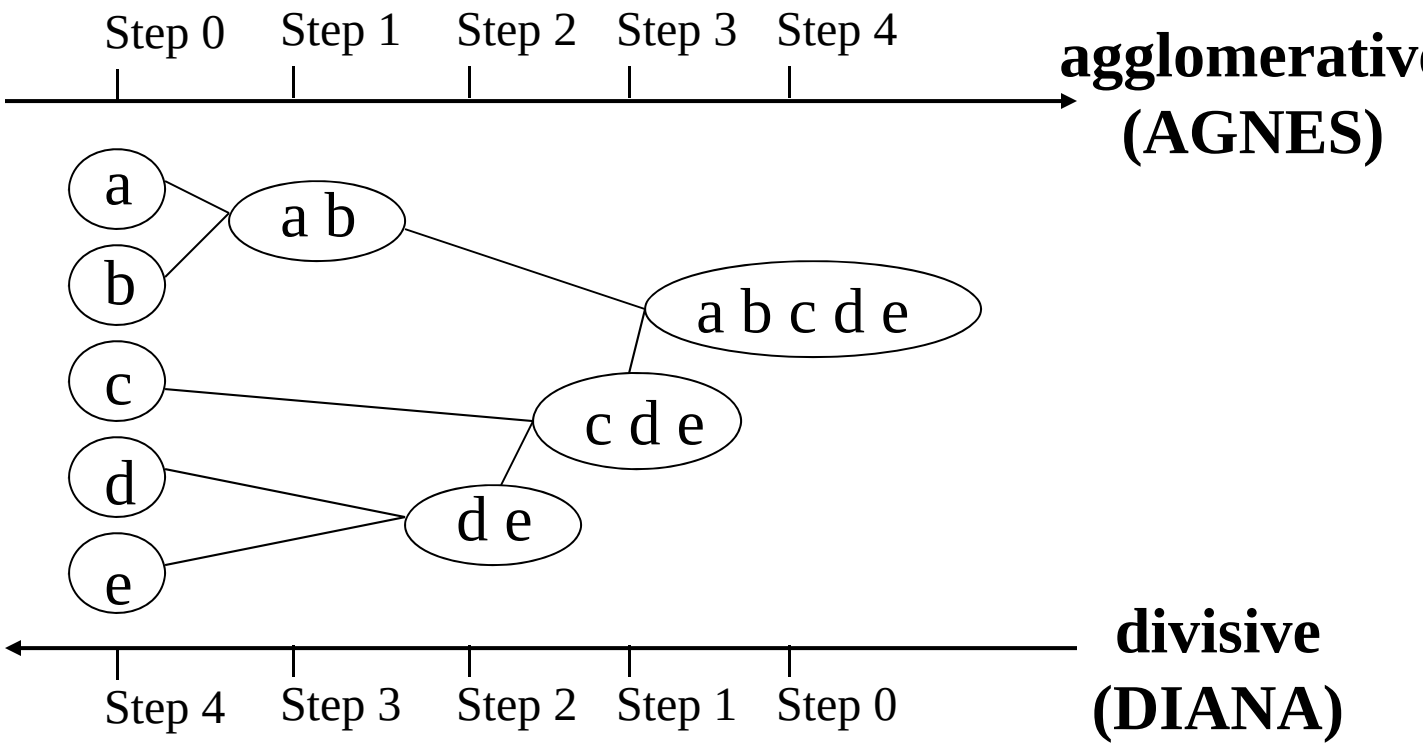
Variations of the *K-Means* Method

- Bisecting k-means illustration



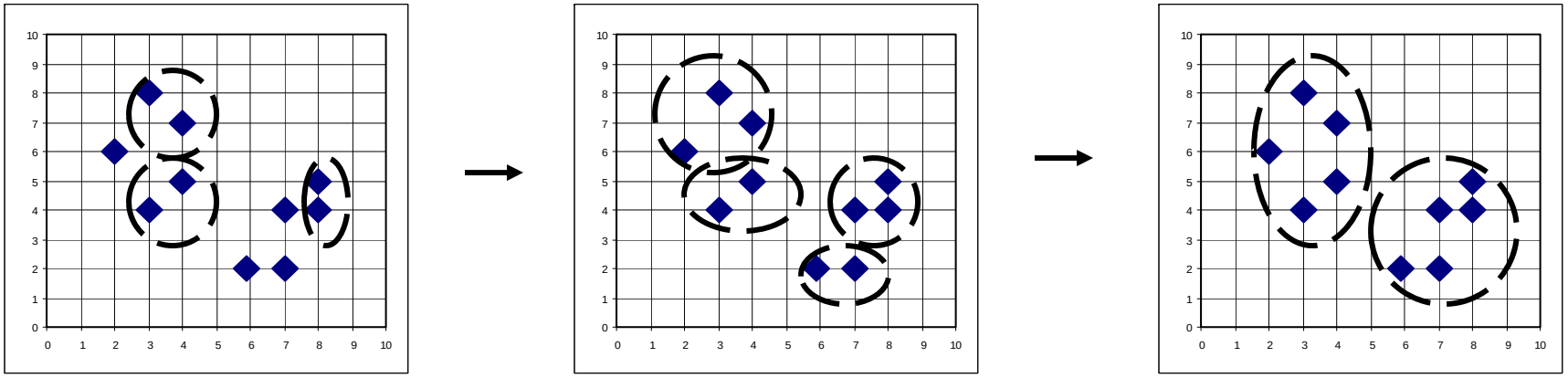
Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition



AGNES (Agglomerative Nesting)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical packages, e.g., Splus
- Use the **single-link** method and the dissimilarity matrix
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



Weakness of Agglomerative Clustering

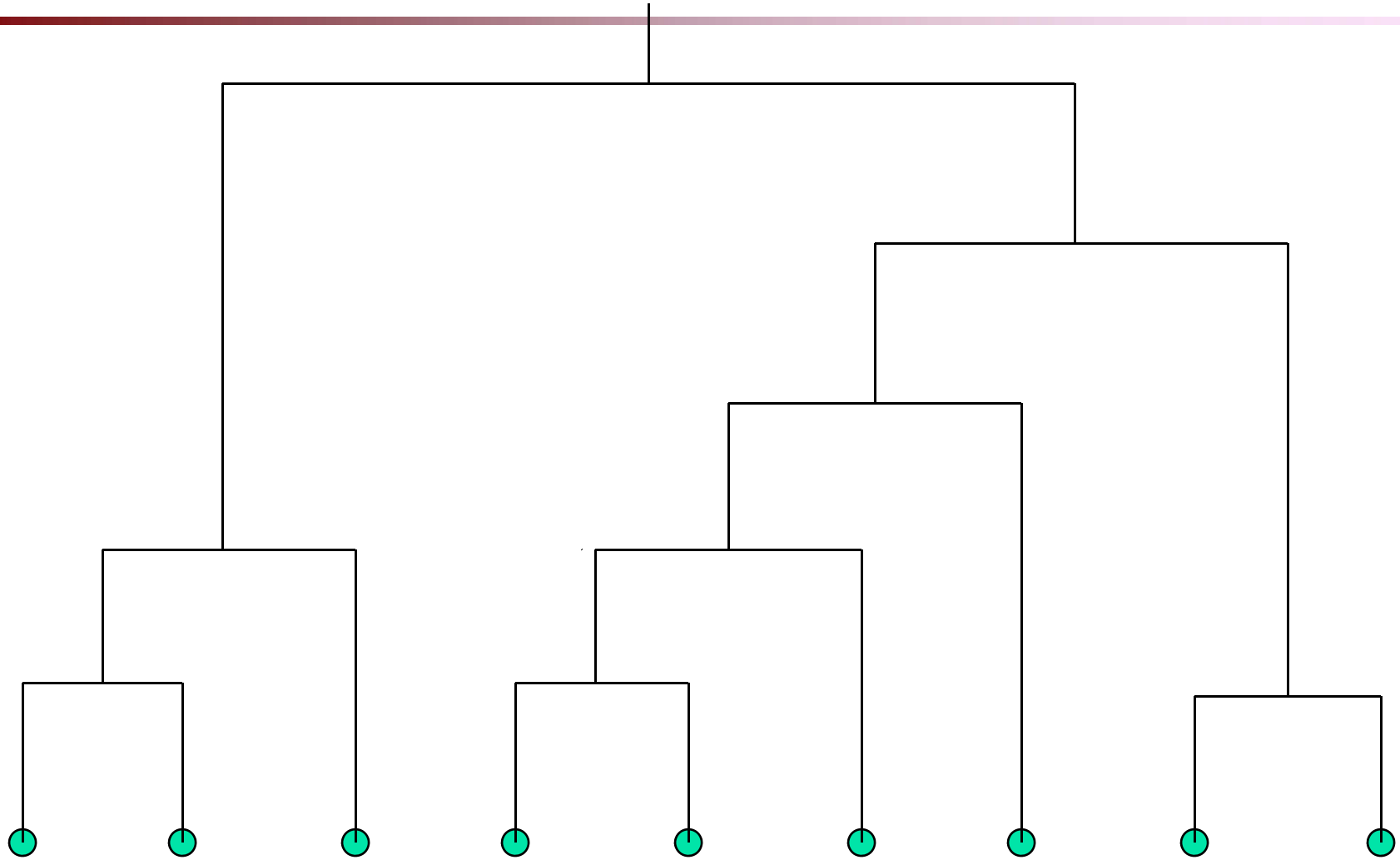
- Can never undo what was done previously
- Do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects

Dendrogram: Shows How Clusters are Merged

- Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram

- A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster

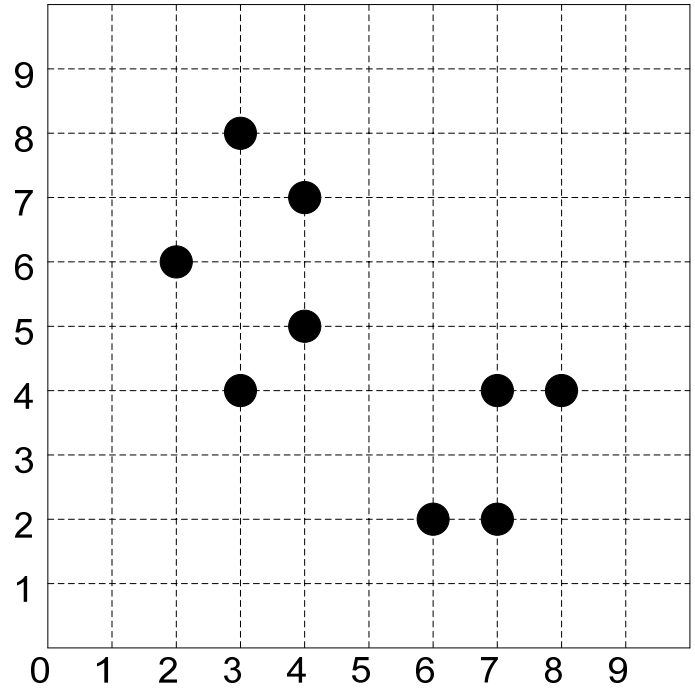
Dendrogram: Shows How Clusters are Merged



Hierarchical Clustering

- Example (agglomerative)

	A1	A2
1	2	6
2	3	4
3	3	8
4	4	5
5	4	7
6	6	2
7	7	2
8	7	4
9	8	4



Hierarchical Clustering

- SPSS hierarchical clustering output

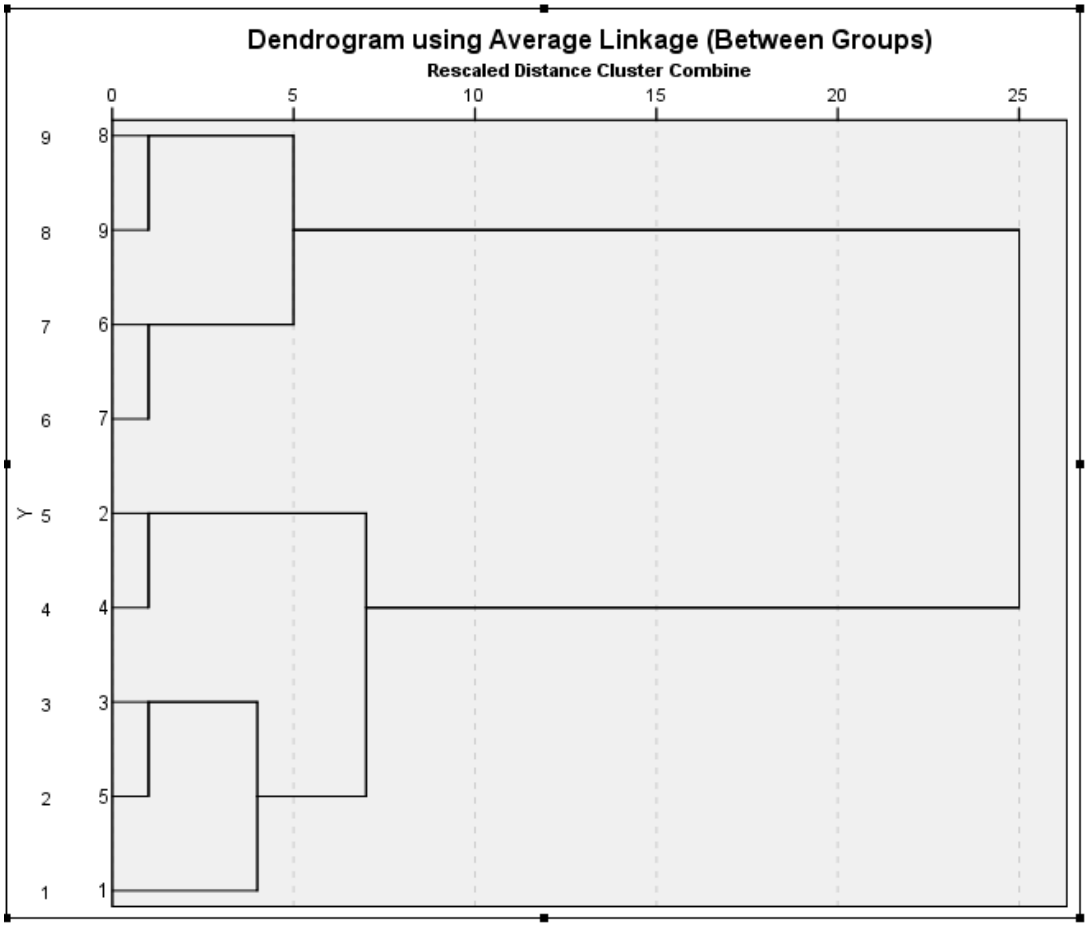
Average Linkage (Between Groups)

Agglomeration Schedule

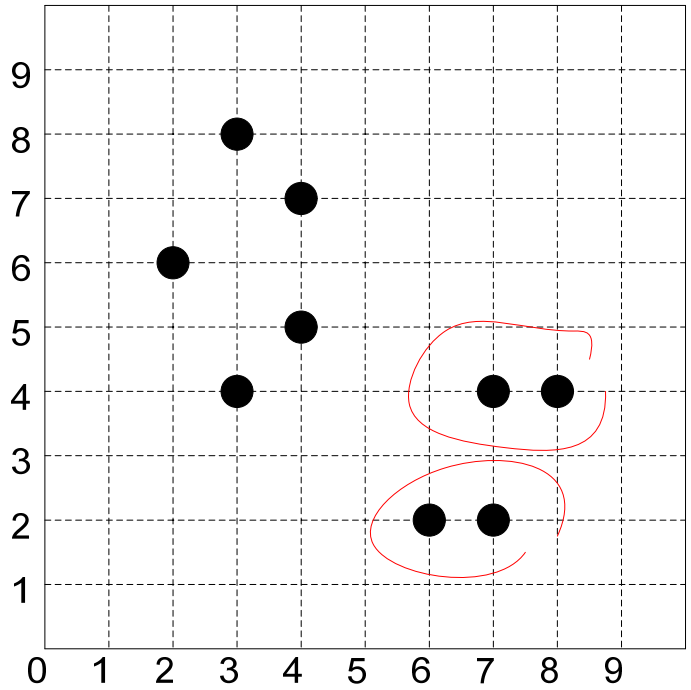
Stage	Cluster Combined		Coefficients	Stage Cluster First Appears		Next Stage
	Cluster 1	Cluster 2		Cluster 1	Cluster 2	
1	8	9	1.000	0	0	6
2	6	7	1.000	0	0	6
3	3	5	2.000	0	0	5
4	2	4	2.000	0	0	7
5	1	3	5.000	0	3	7
6	6	8	5.500	2	1	8
7	1	2	8.333	5	4	8
8	1	6	27.500	7	6	0

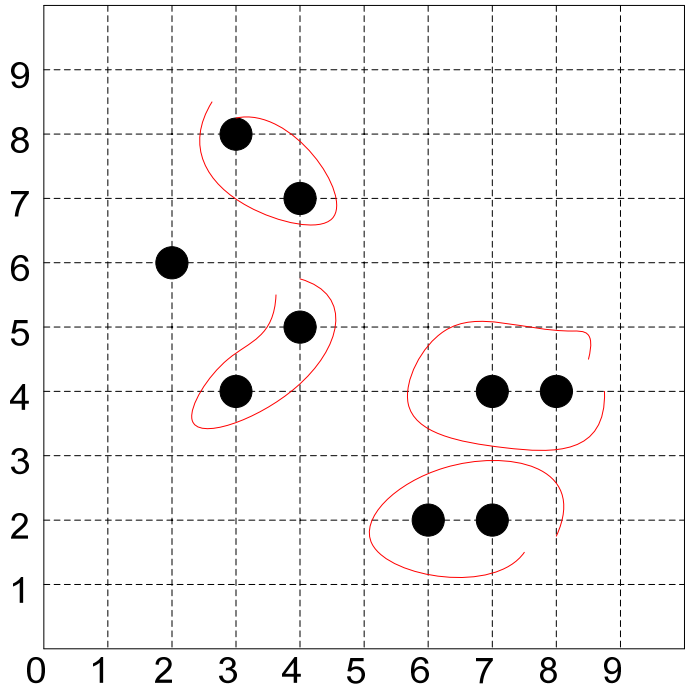
Hierarchical Clustering

- SPSS hierarchical clustering output

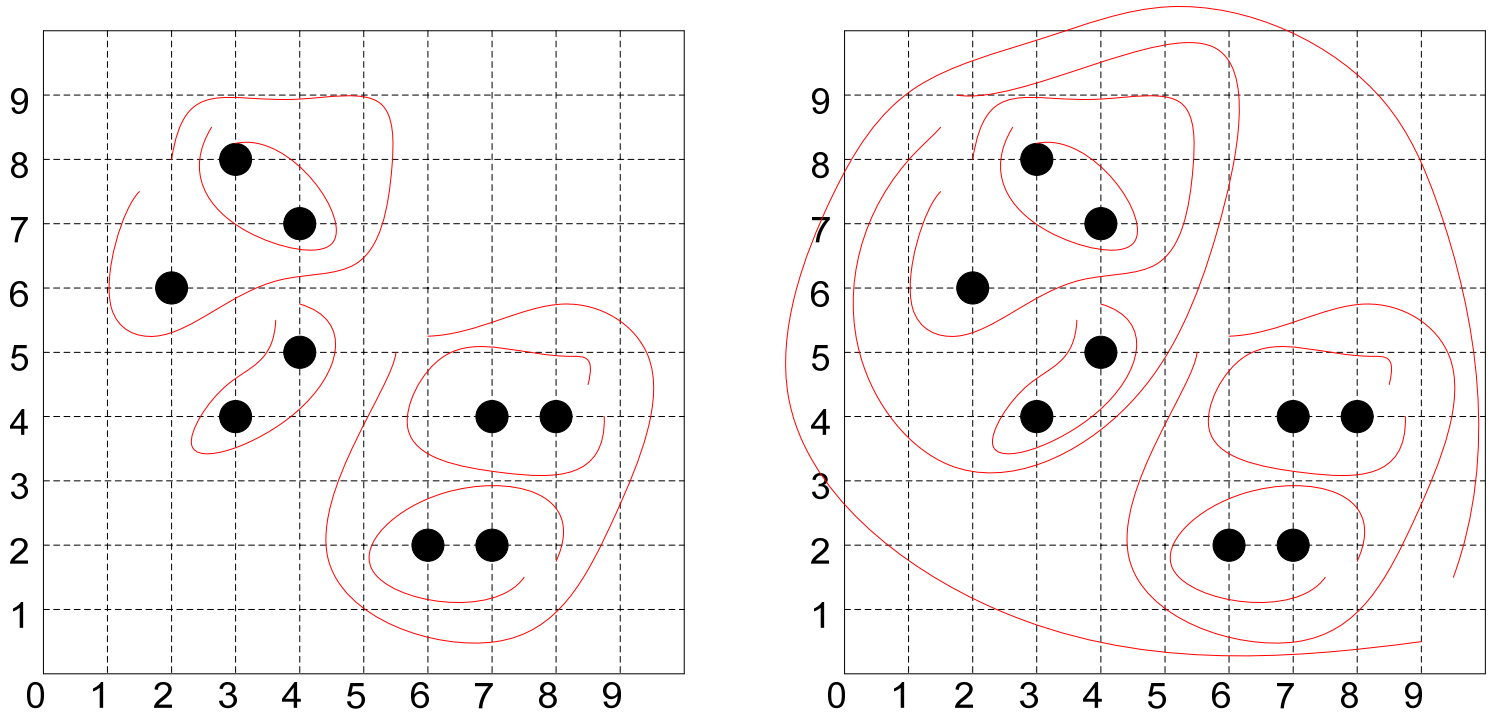


Hierarchical Clustering



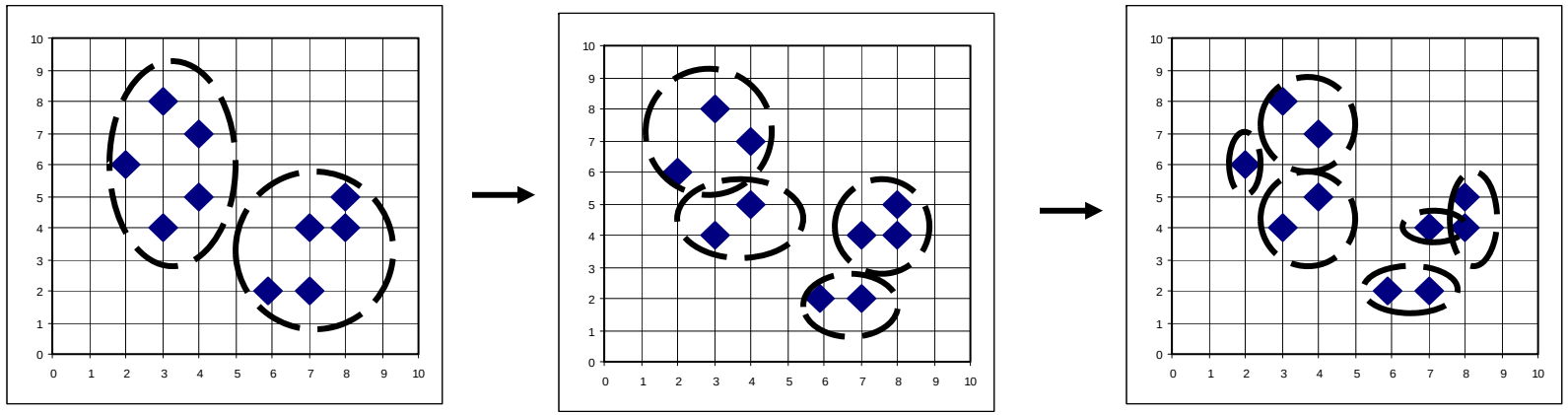


Hierarchical Clustering



DIANA (Divisive Analysis)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Inverse order of AGNES
- Eventually each node forms a cluster on its own



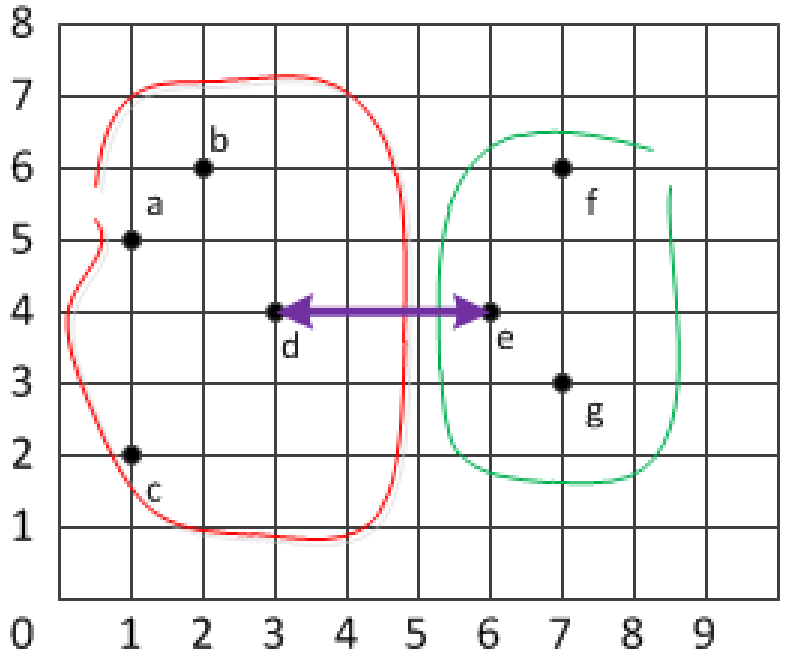
Distance between Clusters

- Minimum distance (single link)
- Maximum distance (complete link)
- Average distance
- Mean distance

Distance between Clusters

- Minimum distance (single link): smallest distance between an element in one cluster and an element in the other, i.e., dist(K_i, K_j) = min(t_{ip}, t_{jq})

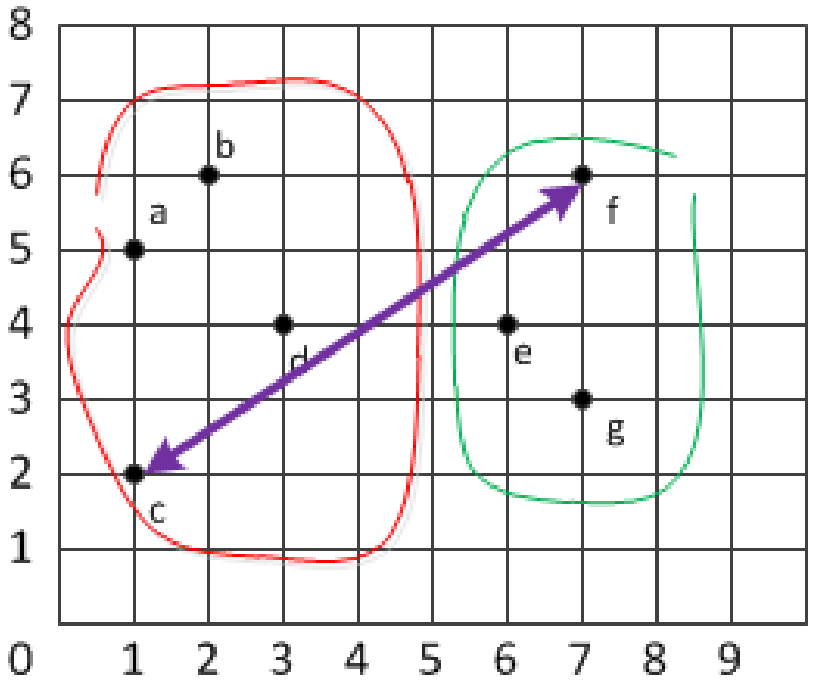
minimum distance = 3
(using Manhattan distance)



Distance between Clusters

- Maximum distance (complete link): largest distance between an element in one cluster and an element in the other, i.e., dist(K_i, K_j) = max(t_{ip}, t_{jq})

maximum distance = 10
(using Manhattan distance)



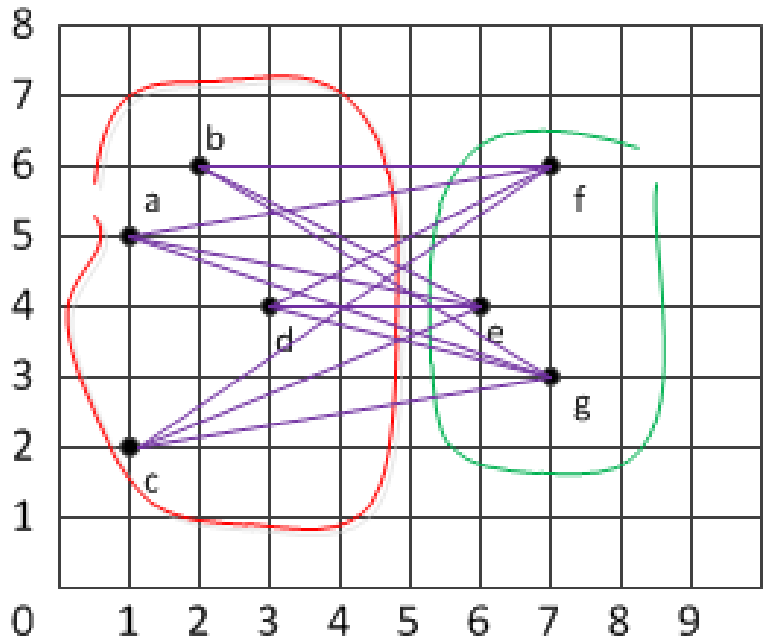
Distance between Clusters

- Average distance: average of distances between all pairs of elements, i.e., $dist(K_i, K_j) = avg(t_{ip}, t_{jq})$

d(a,e) = 6 d(b,e) = 6
d(a,f) = 7 d(b,f) = 5
d(a,g) = 8 d(b,g) = 8

d(c,e) = 7 d(d,e) = 3
d(c,f) = 10 d(d,f) = 6
d(c,g) = 7 d(d,g) = 5

average distance
= average of all the above
= 78 / 12
= 6.5

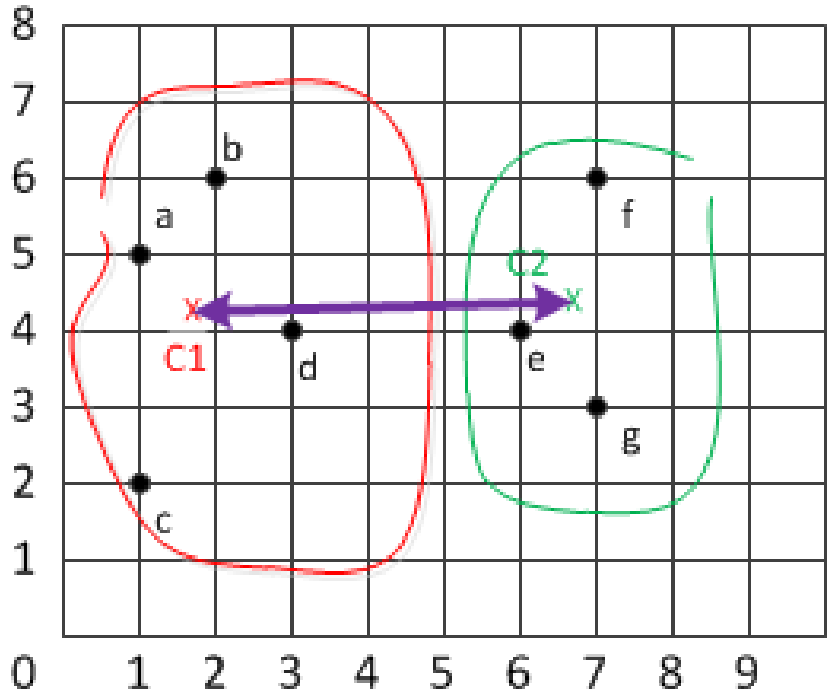


Distance between Clusters

- Mean distance: distance between the centroids of two clusters, i.e., $dist(K_i, K_j) = dist(C_i, C_j)$

C1 = (1.75, 4.25)
C2 = (6.67, 4.33)

mean distance = 5.0
(using Manhattan distance)



Evaluation of Clustering

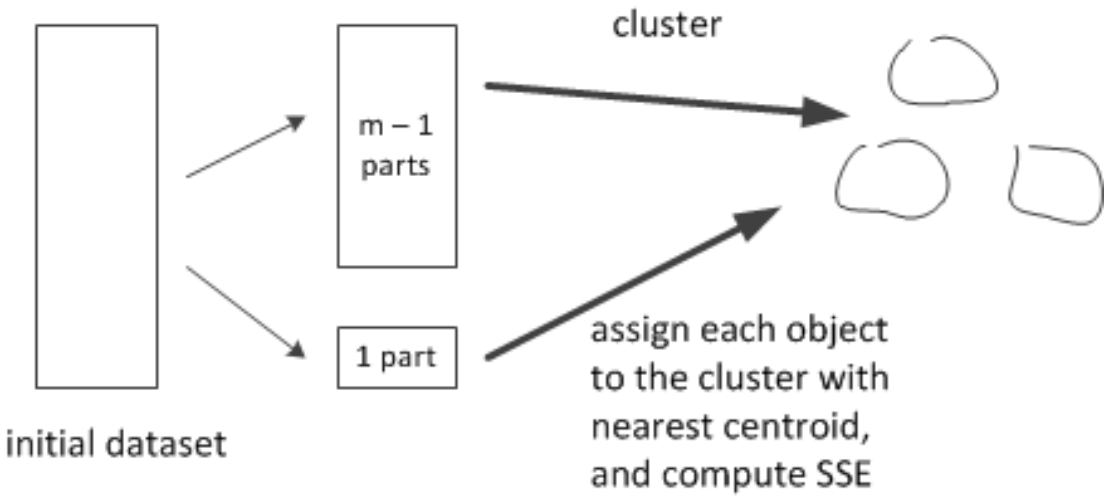
- Assessing clustering tendency
  - See whether there are non-random structures in the dataset (i.e., whether there are natural clusters in the dataset)
  - Measure the probability that the dataset was generated by a uniform data distribution. If so, there may not be any natural clusters at all.

Evaluation of Clustering

- Determining the number of clusters
  - Not trivial because, in part, we don't know "right" number of clusters
  - A simple method: $k = \frac{\sqrt{n}}{2}$
  - Elbow method: as we increase *k,* find the point where the marginal benefit in regard to *SSE* does not increase significantly (typically a turning point in a graph)
  - Cross-validation method: While changing the value of *k*, perform cross-validation and select *k* that gives the best result.

Evaluation of Clustering

- Cross-validation



  - Repeat this *m* times and get the average of SSE's
  - Do this for *k* = 2, 3, 4, … and select the k with the smallest average SSE.

Evaluation of Clustering

- Measuring clustering quality
  - Extrinsic method
    - When ground truth is available
    - BCubed precision, BCubed recall
  - Intrinsic method
    - When ground truth is not available
    - Measures how well clusters are separated and how compact each cluster is
    - Silhouette coefficient

Evaluation of Clustering

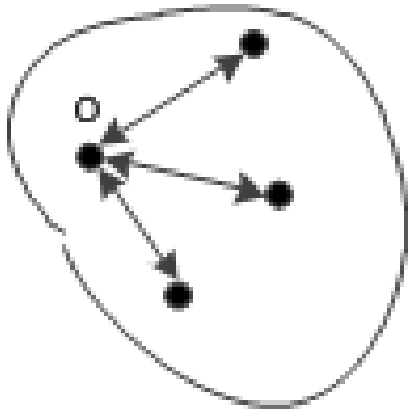
- Silhouette coefficient of an object *o*
  - Measures how well clusters are separated and how compact each cluster is.
  - Assume objects are partitioned into *k* clusters, C_1, C_2, …, C_k.
  - Silhoutte coefficient is calculated as:

$$s(o) = \frac{b(o) - a(o)}{\max\{a(o), b(o)\}}$$

Evaluation of Clustering

- $a(o)$
  - Represents compactness of the cluster o belongs to.
  - Calculates the average distance between an object o and all other objects in the same cluster.
  - Smaller values are better

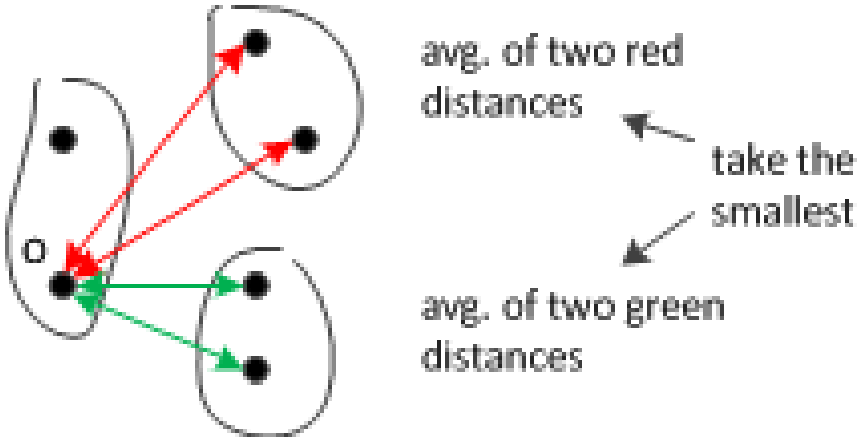
$$a(o) = \frac{\sum_{o' \in C_i, o \neq o'} dist(o, o')}{|C_i - 1|}$$



Evaluation of Clustering

- $b(o)$
  - Represents how far o is from other clusters
  - Calculates the minimum average distance between an object o in a cluster to all other clusters.
  - Larger values are better.

$$b(o) = \min_{C_j : 1 \le j \le k, j \neq i} \left\{ \frac{\sum_{o' \in C_j} dist(o, o')}{|C_j|} \right\}$$



Evaluation of Clustering

$$s(o) = \frac{b(o) - a(o)}{\max\{a(o), b(o)\}}$$

- *s(o)* is between -1 and 1
- Closer to 1 means it is better
- Negative: not good; *o* is closer to objects in other clusters than to objects in its own cluster.

- Overall cluster quality:
  - Compute average silhouette coefficient of all objects
  - Use the average to evaluate the quality of clustering

References

- Han, J., Kamber, M., Pei, J., “Data mining: concepts and techniques,” 3rd Ed., Morgan Kaufmann, 2012
- http://www.cs.illinois.edu/~hanj/bk3/
- P. Tan, M. Steinbach, V. Kumar, “Introduction to Data Mining,” Addison Wesley, 2006.
- The SPSS TwoStep Cluster Component, Technical Report, SPSS