Clustering I

- Clustering Concepts, Applications and Issues
- Data Similarity
- Clustering Approaches
  - K-mean clustering
  - Hierarchical clustering
- Take-home messages

What is a Cluster? Cluster Analysis? Clustering?

- Cluster: A collection of data objects that are
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters
- Cluster analysis
  - Grouping a set of data objects into clusters
- Clustering is an *unsupervised learning approach, with* no predefined classes (or supervision signals)
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

General Applications of Clustering

- Spatial Data Analysis
  - create thematic maps in GIS by clustering feature spaces
  - detect spatial clusters and explain them in spatial data mining
- Image Processing (cf. face detection via clustering of skin color pixels)
- Economic Science (especially market research; grouping of customers)
- WWW
  - Document classification
  - Cluster Weblog data to discover groups of similar access patterns

Specific Examples of Clustering Applications

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Land use: Identification of areas of similar land use in an earth observation database
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost
- City-planning: Identifying groups of houses according to their house type, value, and geographical location

What is Good Clustering?

- A good clustering method will produce high quality clusters with
    - high intra-class similarity
    - low inter-class similarity
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation.
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns.
- Measure of clustering quality:
    - Normalized Mutual Information (NMI) [see sklearn.metrics.NMI]
    - Rand Index [see Wiki]

  Both measure the similarity between two data clusterings (e.g. ground truth vs clustering result)

Requirements of Clustering in Data Mining

- Scalability
- Ability to deal with different types of attributes
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- High dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability

Data Similarity

Data Structures

- Data matrix
  - object-by-variable structure (n objects & p variables/ attributes)

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

- Dissimilarity matrix
  - object-by-object structure
  - n objects here!

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ : & : & : & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

Measure the Quality of Clustering

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

Similarity and Dissimilarity Between Objects

- Distances are normally used to measure the similarity or dissimilarity between two data objects

- Some popular ones include: *Minkowski distance*:

$$d(\vec{i},\vec{j}) = \sqrt[q]{(|x_{i_1} - x_{j_1}|^q + |x_{i_2} - x_{j_2}|^q + \ldots + |x_{i_p} - x_{j_p}|^q)} \qquad \text{L}_q \text{ norm}$$

  where $\vec{i} = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and $\vec{j} = (x_{j1}, x_{j2}, \ldots, x_{jp})$ are two *p*-dimensional data objects, and *q* is a positive integer

- If $q=1$, d is Manhattan (or city block) distance

$$d(\vec{i},\vec{j}) = |x_{i_1} - x_{j_1}| + |x_{i_2} - x_{j_2}| + \ldots + |x_{i_p} - x_{j_p}| \qquad \text{L}_1 \text{ norm}$$

Similarity and Dissimilarity Between Objects (cont.)

- *If $q=2$, d* is Euclidean distance:

$$d(\vec{i},\vec{j}) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \ldots + |x_{i_p} - x_{j_p}|^2)}$$

L_2 norm

  - Properties
    - $d(i,j) \geq 0$
    - $d(i,i) = 0$
    - $d(i,j) = d(j,i)$
    - $d(i,j) \leq d(i,k) + d(k,j)$

Remember this?

$x^2 + y^2 = z^2$

- Also, one can use weighted distance, parametric Pearson product moment correlation, or other dissimilarity measures.

Distance Measure for Binary Variables

- A contingency table for binary data

		Object j		
		1	0	*sum*
	1	a	b	$a+b$
Object *i*	0	c	d	$c+d$
	sum	$a+c$	$b+d$	p

- Simple matching coefficient (invariant, if the binary variable is *symmetric*): $$d(i,j) = \frac{b+c}{a+b+c+d}$$
- Jaccard coefficient (noninvariant, if the binary variable is *asymmetric*): $$d(i,j) = \frac{b+c}{a+b+c}$$

Dissimilarity between Binary Variables

Example

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

- gender is a symmetric attribute
- the remaining attributes are asymmetric binary
- let the values Y and P be set to 1, and the value N be set to 0

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	1	1	0	1	0	0	0
Mary	0	1	0	1	0	1	0
Jim	1	1	1	0	0	0	0

$$\left.\begin{aligned} d(jack, mary) &= \frac{0+1}{2+0+1} = 0.33 \\ d(jack, jim) &= \frac{1+1}{1+1+1} = 0.67 \\ d(jim, mary) &= \frac{1+2}{1+1+2} = 0.75 \end{aligned}\right\}$$

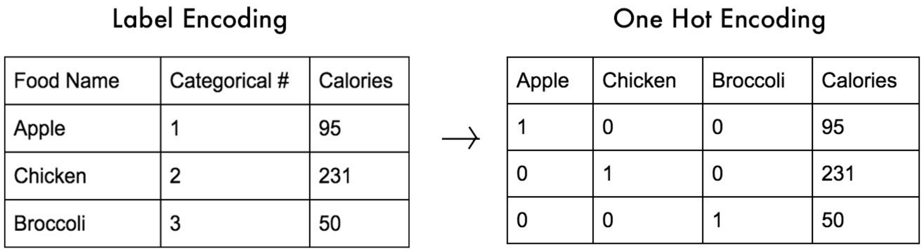
Only asymmetric variables are considered!!!

Distance Measure for Nominal/Categorical Variables

- A generalization of the binary variable in that it can take more than 2 states, e.g., red, yellow, blue, green
- Method 1: Simple matching
  - *m*: # of matches, *p*: total # of variables

$$d(i,j) = \frac{p-m}{p}$$

- Method 2: use a large number of binary variables
  - creating a new binary variable for each of the *M* nominal states
  - 1-hot encoding



Label Encoding

Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

→

One Hot Encoding

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

Distance Measure for Transactional Data

- Basic ideas:
  - Let T_1 = $\{A,B,C\}$, T_2=$\{C,D,E\}$ where A-E denote items
  - Similarity function defined as:

$$Sim(T_1,T_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}$$

    where ∩ & ∪ denote the intersection and union of two transaction records respectively.

  - For our example, we have

$$Sim(T_1,T_2) = \frac{|\{C\}|}{|\{A,B,C,D,E\}|} = \frac{1}{5}$$

Clustering Approaches

Major Clustering Approaches

- Partitioning algorithms: Construct various partitions and then evaluate them by some criteria
- Hierarchical algorithms: Create a hierarchical decomposition of the set of data (or objects) using some criteria

These two are most well-known in general applications

- Density-based: based on connectivity and density functions
- Grid-based: based on a multiple-level granularity structure
- 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

Partitioning Algorithms: Basic Concept

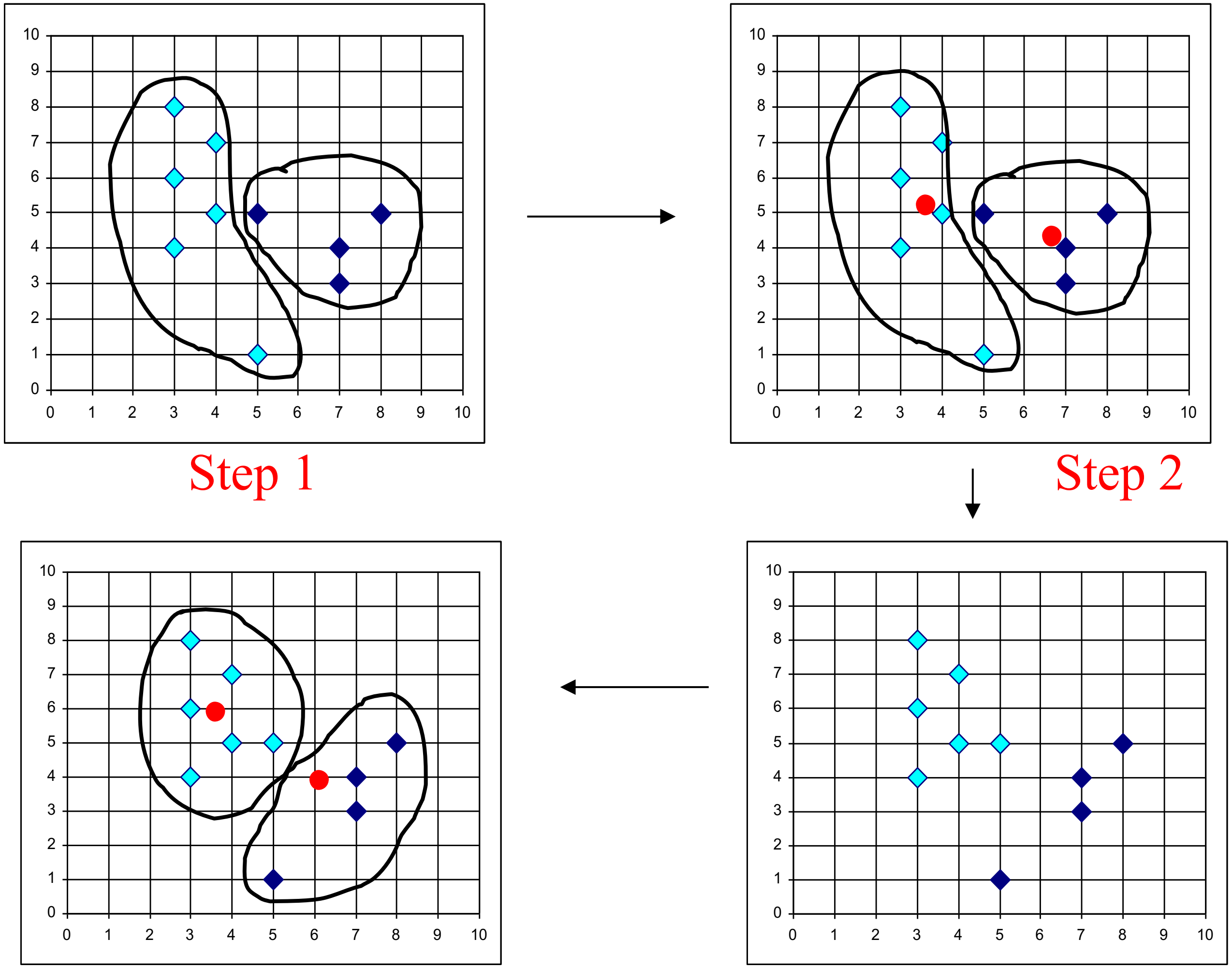
- *Partitioning approach:*
  - Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters
- Given a particular ***k***, find a partition of *k clusters* that optimizes the chosen partitioning criterion (e.g. high intra-class similarity)
- Two methods
  - Globally optimal method: exhaustively enumerate all partitions (nearly impossible for large *n*)
  - Heuristic methods: *k-means* and *k-medoids* algorithms
    - *k-means* (MacQueen'67): 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

K-Means Clustering Method

- Given *k*, the *k-means* algorithm can be implemented by these four steps:
  1. *Initialization: Partition objects into k nonempty subsets*
  2. *Mean-op: Compute seed points as the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.*
  3. *Nearest_Centroid-op: Assign each object to the cluster with the nearest seed point.*
  4. *Go back to the step 2, stop when no more new assignment.*

K-Means Clustering Method (see demo here)

- Example



Step 1

Step 2

Step 3

Step 4

Red dots denote the cluster centroids

Comments on the *K-Means* Method

- *Strength*
  - *Relatively efficient*: *O*(*tkn*), where *n* is # objects, *k* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.
  - Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *deterministic annealing* and *genetic algorithms*
- *Weakness*
  - Applicable only when *mean* is defined, then what about categorical data? What is the mean of red, orange and blue?
  - 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; the basic cluster shape is spherical (convex shape)*

Variations of the *K-Means* Method

- There exist a few variants of the *k-means* which differ in
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means
- Handling categorical data: *k-modes* (Huang'98)
  - Replacing means of clusters with modes
  - Using new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters
  - A mixture of categorical and numerical data: *k-prototype* method

K-Medoids Clustering Method

- Find *representative* objects, called medoids, in clusters
- *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 clustering
  - *PAM* works effectively for small data sets, but does not scale well for large data sets
- *Attempts to improve it:*
  - *CLARA* (Kaufmann & Rousseeuw, 1990)
  - *CLARANS* (Ng & Han, 1994): Randomized sampling
  - Focusing + spatial data structure (Ester et al., 1995)

CLARA (Clustering Large Applications) (1990)

- *CLARA* (Kaufmann and Rousseeuw in 1990)
  - Built in statistical analysis packages, such as S-Plus
- It draws *multiple samples* of the data set, applies *PAM* on each sample, and gives the best clustering as the output
- *Strength*: deals with larger data sets than *PAM*
- *Weakness*:
  - Efficiency depends on the sample size
  - A good clustering based on samples will not necessarily represent a good clustering of the whole data set if the sample is biased

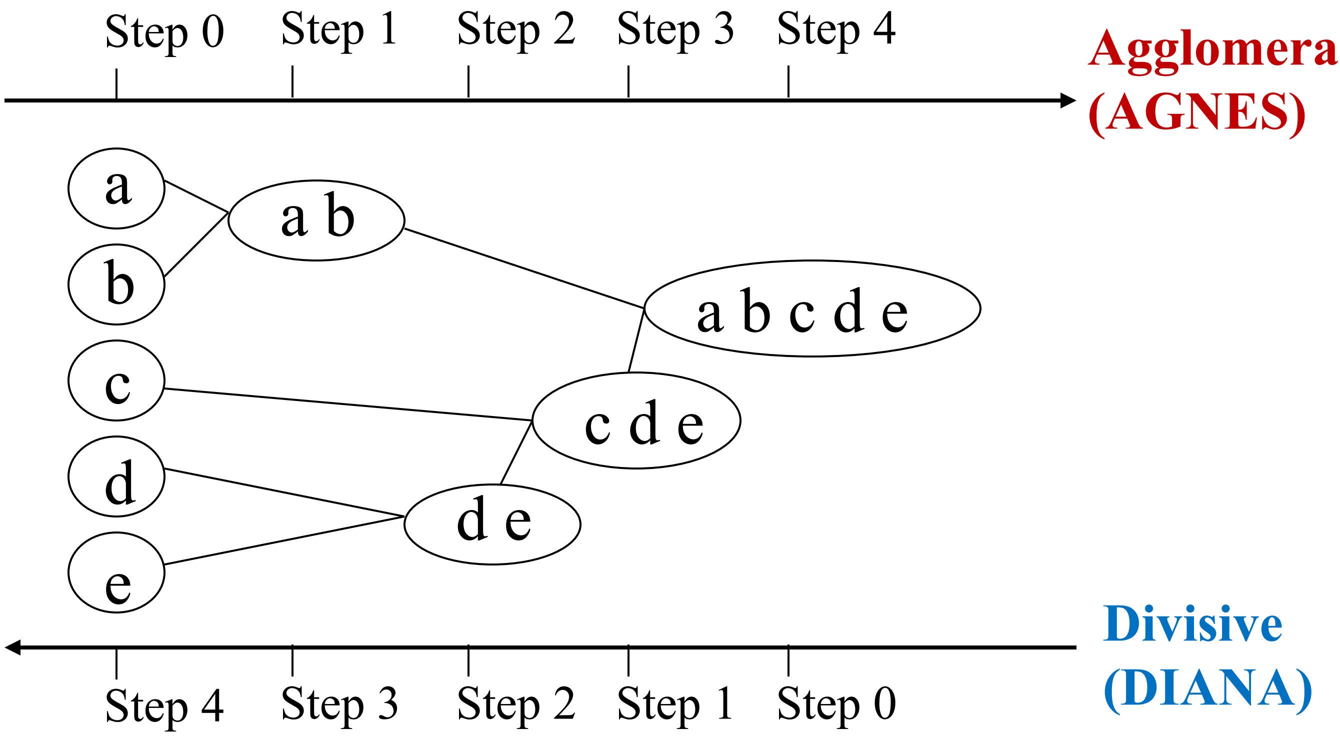
Hierarchical Clustering

Hierarchical Clustering Methods

- The clustering process involves a series of partitioning of the data
  - It may run from a single cluster containing all records to n clusters each containing a single record.
- Two popular approaches
  - Agglomerative (ANGES) & divisive (DIANA) methods
- The results may be represented by a dendrogram
  - Diagram illustrating the fusions or divisions made at each successive stage of the analysis.

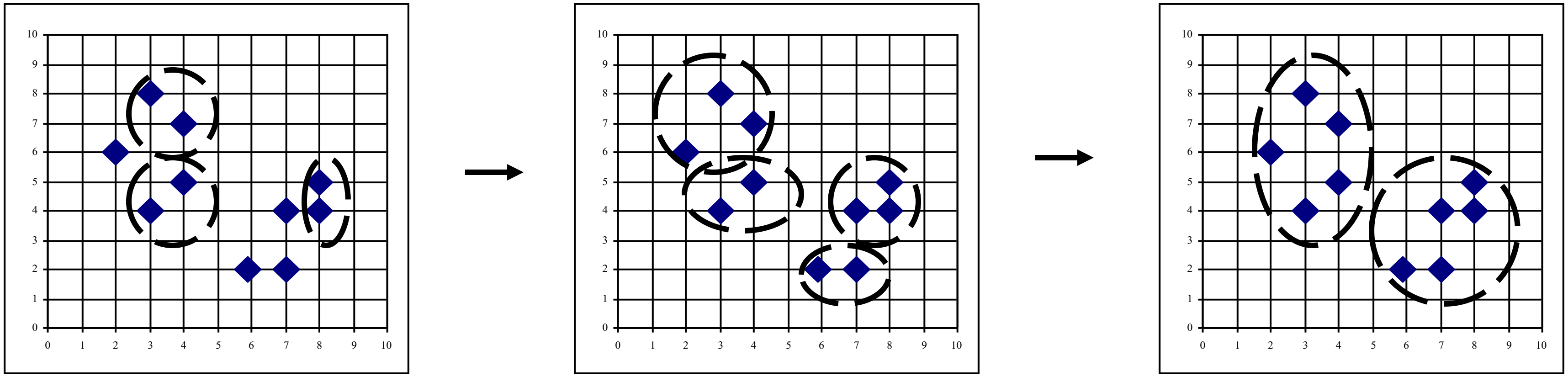
Hierarchical Clustering

- 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 analysis packages, e.g. S-Plus
- Use the Single-Linkage 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



Agglomerative Nesting/Clustering: Single Linkage Method

Basic operations:

START:

♦ Each cluster of $\{C_1, \cdots, C_j, \cdots, C_n\}$ contains a single individual.

Step 1.

♦ Find nearest pair of distinct clusters C_i & C_j

♦ Merge C_i & C_j .

♦ Decrement the number of cluster by one.

Step 2.

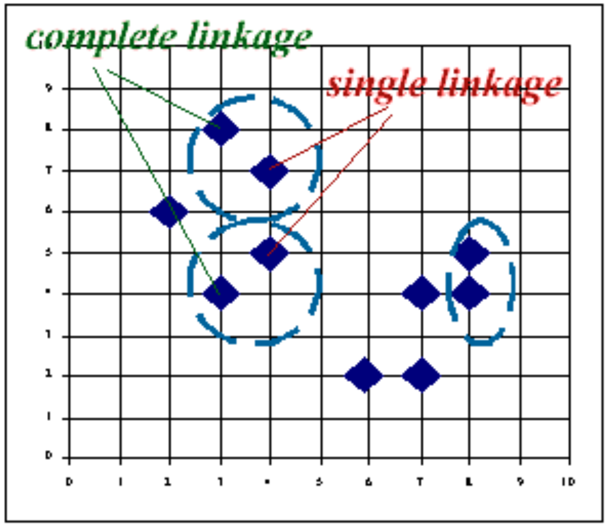
♦ If the number of clusters equals to one then stop, else return to 1.

Single linkage clustering

♦ Also known as nearest neighbor (1NN) technique.

♦ The distance between groups is defined as the closest pair of records from each group.

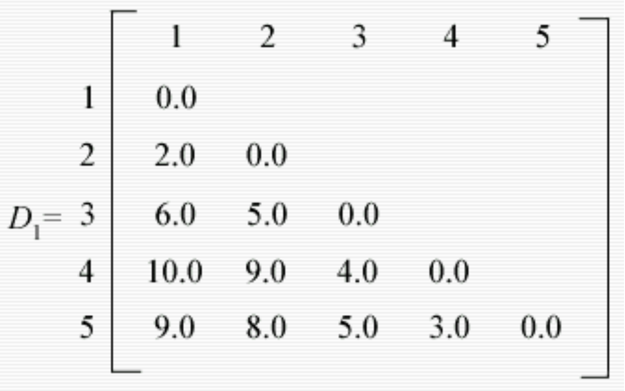
Agglomerative Nesting/Clustering: Complete Linkage and others



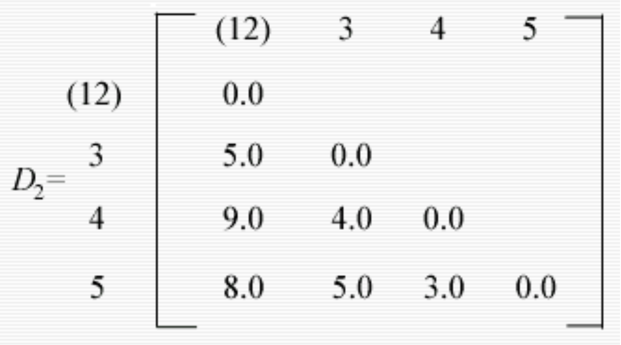
- Complete linkage clustering
  - Also known as furthest neighbor technique.
  - Distance between groups is now defined as that of the most distant pair of individuals (opposite to single linkage method).
- Group-average clustering
  - Distance between two clusters is defined as the average of the distances between all pairs of individuals between the two clusters.
- Centroid clustering
  - Groups once formed are represented by the mean values computed for each attribute (i.e. a mean vector).
  - Inter-group distance is now defined in terms of distance between two such mean vectors.

Single Linkage Method: An Example

- Assume the distance matrix D_1.
- The smallest entry in the matrix is that for individuals 1 and 2, consequently these are joined to form a two-member cluster. Distances between this cluster and the other three individuals are recomputed as
  - *d*(12)3 = min[*d*13,d23] = *d*23 = 5.0
  - *d*(12)4 = min[*d*14,d24] = *d*24 = 9.0
  - *d*(12)5 = min[*d*15,d25] = *d*25 = 8.0
- A new matrix D_2 may now be constructed whose entries are inter-individual distances and cluster-individual values.



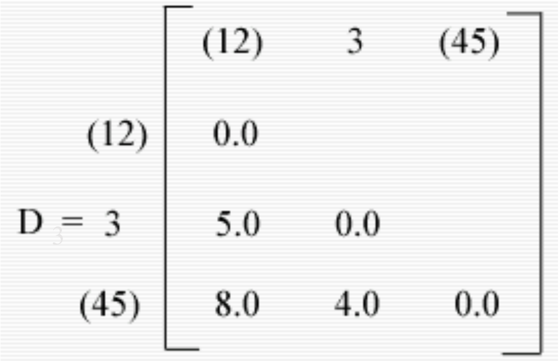
$$D_1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \left[\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ 0.0 & & & & \\ 2.0 & 0.0 & & & \\ 6.0 & 5.0 & 0.0 & & \\ 10.0 & 9.0 & 4.0 & 0.0 & \\ 9.0 & 8.0 & 5.0 & 3.0 & 0.0 \end{array} \right]$$



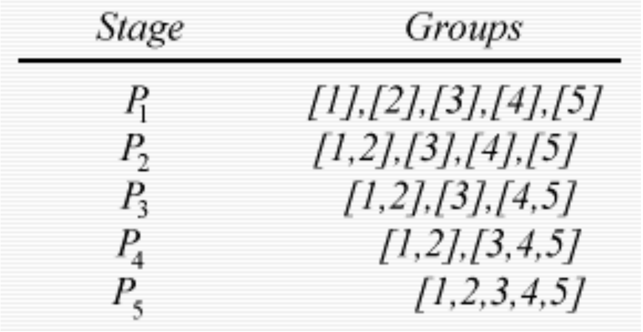
$$D_2 = \begin{array}{c} \\ (12) \\ 3 \\ 4 \\ 5 \end{array} \left[\begin{array}{cccc} (12) & 3 & 4 & 5 \\ 0.0 & & & \\ 5.0 & 0.0 & & \\ 9.0 & 4.0 & 0.0 & \\ 8.0 & 5.0 & 3.0 & 0.0 \end{array} \right]$$

Single Linkage Method: An Example (cont.)

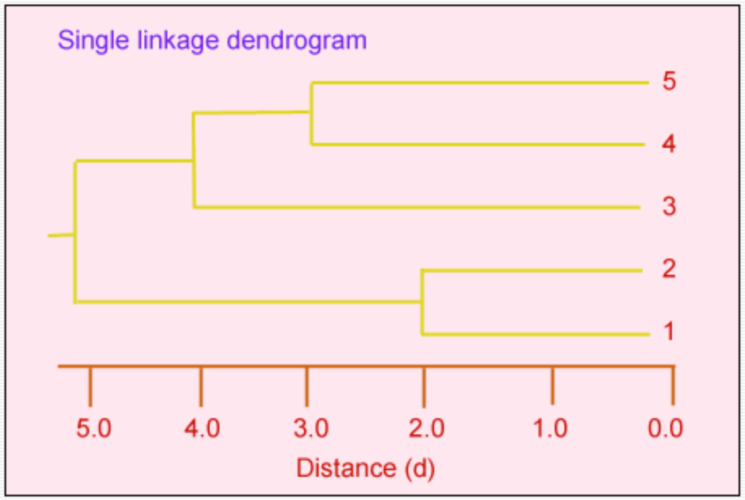
- The smallest entry in D_2 is that for individuals 4 and 5, so these now form a second two-member cluster, and a new set of distances found
    - $d(12)3 = 5.0$ as before
    - $d(12)(45) = \min[d14,d15,d24,d25] = 8.0$
    - $d(45)3 = \min[d34,d35] = d34 = 4.0$
- These may be arranged in a matrix D_3.
- The smallest entry in now $d(45)3$ and so individual 3 is added to the cluster containing individuals 4 and 5. Finally the groups containing individuals 1, 2 and 3, 4, 5 are combined into a single cluster. The partitions produced at each stage are on the right.



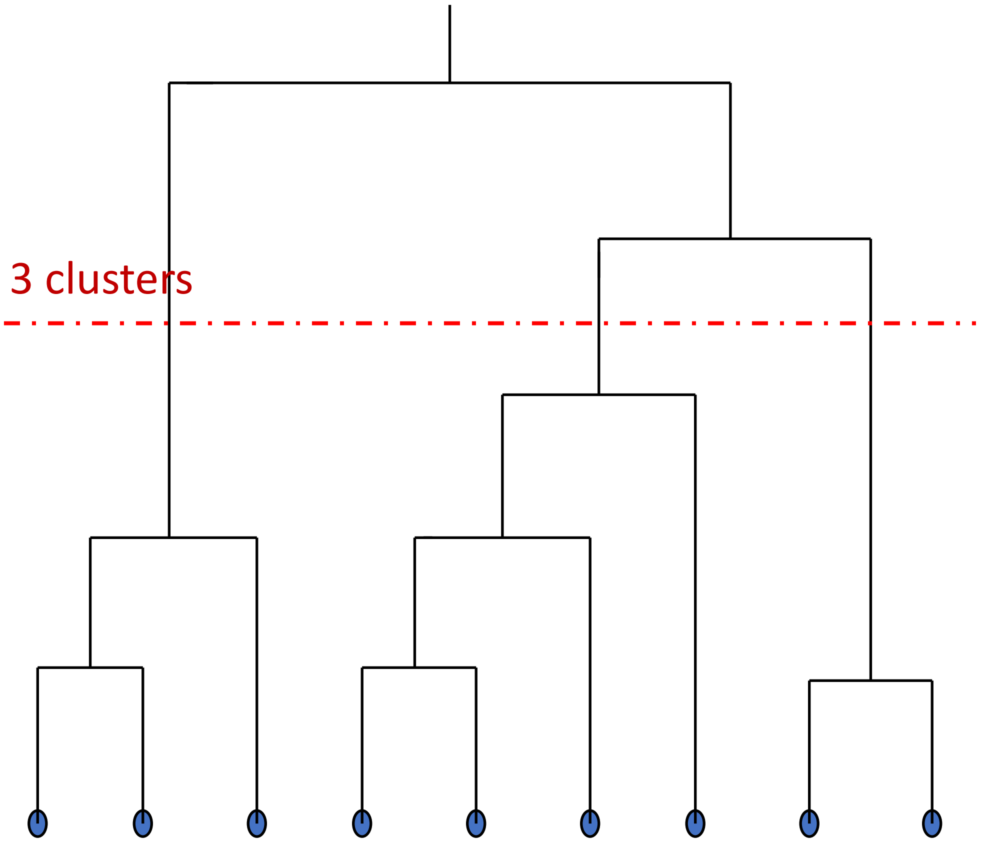
$$D_3 = \begin{array}{c|ccc} & (12) & 3 & (45) \\ \hline (12) & 0.0 & & \\ 3 & 5.0 & 0.0 & \\ (45) & 8.0 & 4.0 & 0.0 \end{array}$$



Stage	*Groups*
P_1	*[1],[2],[3],[4],[5]*
P_2	*[1,2],[3],[4],[5]*
P_3	*[1,2],[3],[4,5]*
P_4	*[1,2],[3,4,5]*
P_5	*[1,2,3,4,5]*



A *Dendrogram* Shows How the Clusters are Merged Hierarchically

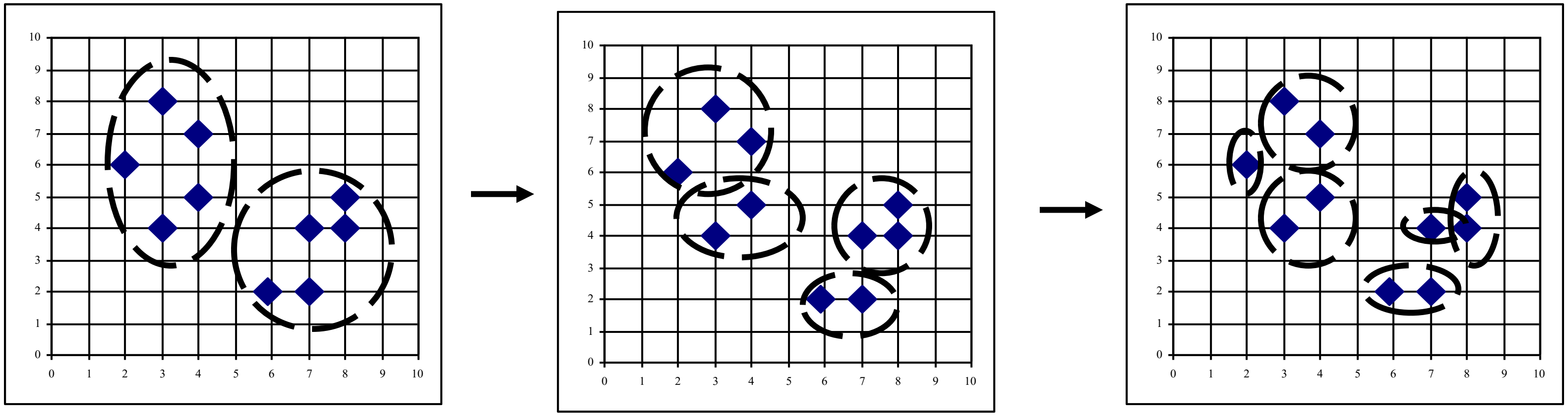


Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram.

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

DIANA (Divisive Analysis)

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



Major weakness of Agglomerative Clustering Methods

- Do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects (need to compute the similarity or dissimilarity of each pair of objects)
- Can never undo what was done previously
- Hierarchical methods are biased towards finding 'spherical' clusters even when the data contain clusters of other shapes.
- Partitions are achieved by 'cutting' a dendrogram or selecting one of the solutions in the nested sequence of clusters that comprise the hierarchy.
- Deciding of appropriate number of clusters for the data is difficult.
  - An informal method is to examine the differences between fusion levels in the dendrogram. Large changes are taken to indicate a particular number of clusters

Choosing *k*

- Defined by the application, e.g., image quantization
- Plot data (after dimensionality reduction (to be taught later)) and check for clusters
- Incremental (leader-cluster) algorithm: Add one at a time until “elbow” (reconstruction error/log likelihood/intergroup distances)
- Manually check for meaning

Take-home Messages

- With the class label “disappeared”, the learning problem becomes an unsupervised one.
- A notation of data similarity (or distance) is needed!
- For practitioners, they always struggle with how to compute data similarity! E.g. similarity between hacking activities, similarity between time series, etc.
- Clustering is NOT exclusive from classification/regression. It helps classification/regression!