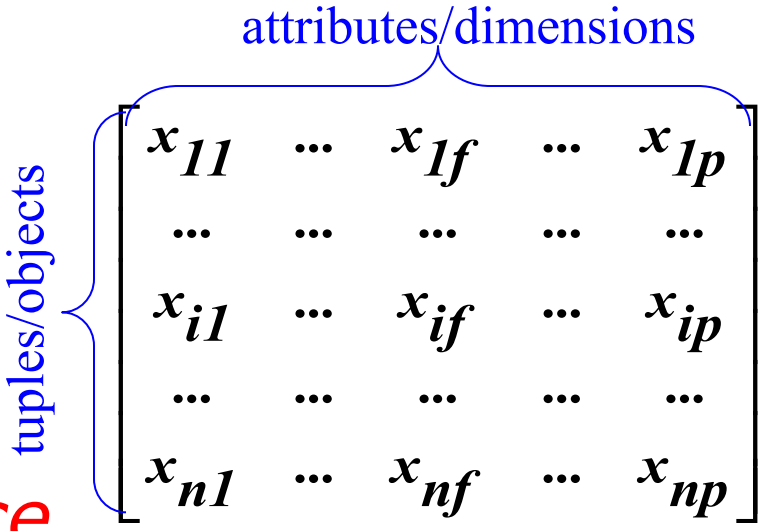
Two types of Input

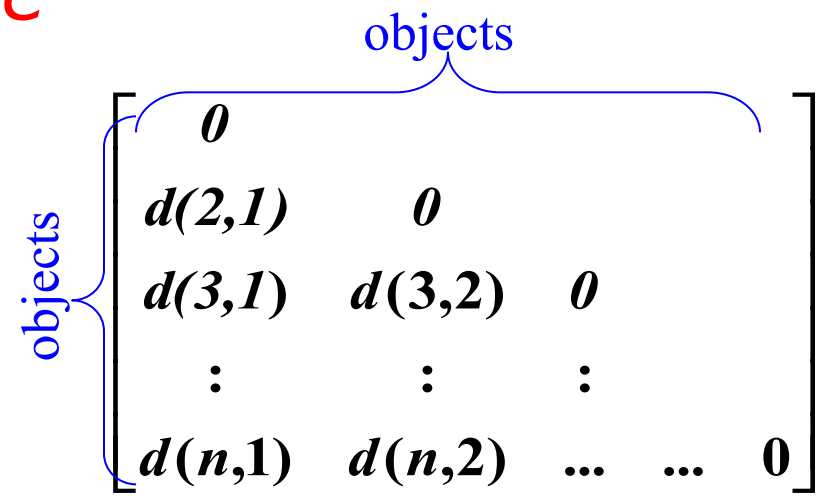
- *data* matrix

the "classic" data input



- *dissimilarity* or *distance* matrix

the desired data input to some clustering algorithms



Measuring Similarity in Clustering

- Dissimilarity/Similarity metric:
  - The dissimilarity $d(i, j)$ between two objects i and j is expressed in terms of a distance function, which is typically a **metric**. A metric satisfies:
  - $d(i, j) \geq 0$ (non-negativity)
  - $d(i, i) = 0$ (coincidence)
  - $d(i, j) = d(j, i)$ (symmetry)
  - $d(i, j) \leq d(i, h) + d(h, j)$ (triangular inequality)
- The definitions of distance functions are usually different for interval-scaled, boolean, categorical, ordinal and ratio-scaled variables.
- Weights may be associated with different variables based on applications and data semantics.

Similarity and Dissimilarity Between Objects

- Distance metrics are normally used to measure the similarity or dissimilarity between two data objects
- The most popular conform to *Minkowski distance*:

$$L_p(i,j)=\left(|x_{i1}-x_{j1}|^p+|x_{i2}-x_{j2}|^p+...+|x_{in}-x_{jn}|^p\right)^{1/p}$$

where $i = (x_{i1}, x_{i2}, \ldots, x_{in})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jn})$ are two n-dimensional data objects, and p is a positive integer

- If $p = 1$, L_1 is the Manhattan (or city block) distance:

$$L_1(i,j)=|x_{i1}-x_{j1}|+|x_{i2}-x_{j2}|+...+|x_{in}-x_{jn}|$$

Similarity and Dissimilarity Between Objects (Cont.)

- *If p = 2*, L_2 is the Euclidean distance:

$$d(i,j)=\sqrt{(|x_{i_1}-x_{j_1}|^2+|x_{i_2}-x_{j_2}|^2+...+|x_{i_n}-x_{j_n}|^2)}$$

  - Properties
    - $d(i,j) \geq 0$
    - $d(i,i) = 0$
    - $d(i,j) = d(j,i)$
    - $d(i,j) \leq d(i,k) + d(k,j)$
- Also one can use weighted distance:

$$d(i,j)=\sqrt{(w_1|x_{i_1}-x_{j_1}|^2+w_2|x_{i_2}-x_{j_2}|^2+...+w_n|x_{i_n}-x_{j_n}|^2)}$$

Type of data in cluster analysis

- Interval-scaled variables
  - e.g., salary, height
- Binary variables
  - e.g., gender (M/F), has_cancer(T/F)
- Nominal (categorical) variables
  - e.g., religion (Christian, Muslim, Buddhist, Hindu, etc.)
- Ordinal variables
  - e.g., military rank (soldier, sergeant, lutenant, captain, etc.)
- Ratio-scaled variables
  - population growth (1,10,100,1000,...)
- Variables of mixed types
  - multiple attributes with various types

Interval-scaled variables

- Continuous measurements on a roughly linear scale
- If we have multiple continuous attributes, it is good to normalize (or standardize) them to have equal importance in clustering:
  - Popular method: min-max normalization ⇒ scale to [0,1]

$$z_{if} = \frac{x_{if} - \min_f}{\max_f - \min_f}$$

  - Other: scale to around 0 using the mean absolute deviation:

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \ldots + |x_{nf} - m_f|)$$

where

$$m_f = \frac{1}{n}(x_{1f} + x_{2f} + \ldots + x_{nf})$$

    - Calculate the standardized measurement (*z-score*)

$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

    - Using mean absolute deviation is more robust than using standard deviation

Binary Variables

- A binary variable has two states: 0 absent, 1 present
- A contingency table for binary data

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

assymetric variable:
0 is very frequent
compared to 1

- 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 (Jaccard coefficient)

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

  - all attributes are asymmetric binary
  - 1 denotes presence or positive test
  - 0 denotes absence or negative test

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

A simpler definition

- Each variable is mapped to a bitmap

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

  - Jack: 101000
  - Mary: 101010
  - Jim: 110000
- Simple match distance:

$$d(i,j) = \frac{\text{number of non-common bit positions}}{\text{total number of bits}}$$

- Jaccard coefficient:

$$d(i,j) = 1 - \frac{\text{number of 1's in } i \wedge j}{\text{number of 1's in } i \vee j}$$

Nominal 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. E.g.:
    - {sunny, overcast, rain} ⇒ {001,010,100}
    - {hot,mild,cool} ⇒ {001,010,100}
    - d(<sunny,mild>,<sunny,cool>) = *d*(001010,001100)

Ordinal Variables

- An ordinal variable can be discrete or continuous
- order is important, e.g., rank
- Can be treated like interval-scaled
  - replacing x_{if} by their rank $r_{if} \in \{1,...,M_f\}$
  - map the range of each variable onto [0, 1] by replacing *i*-th object in the *f*-th variable by

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

  - compute the dissimilarity using methods for interval-scaled variables.
    - E.g., military rank (soldier=1, sergeant=2, lutenant=3, captain=4, major=5, colonel=6, general=7)
    - z(soldier) = 0, z(major) = 4/6, z(general) = 6/6

Ratio-Scaled Variables

- Ratio-scaled variable: a positive measurement on a nonlinear scale, approximately at exponential scale, such as Ae^{Bt} or Ae^{-Bt}
- Methods:
  - treat them like interval-scaled variables — *not a good choice! (why?)*
  - apply logarithmic transformation

$$y_{if} = log(x_{if})$$

  - treat them as continuous ordinal data treat their rank as interval-scaled.

Variables of Mixed Types

- A database may contain all the six types of variables
  - symmetric binary, asymmetric binary, nominal, ordinal, interval and ratio-scaled.
- One may use a weighted formula to combine their effects.

$$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}}$$

  - $\delta_{ij}^{(f)} = 0$ if x_{if} or x_{jf} missing, or $x_{if} = x_{jf} = 0$ in bin. assym.
  - *f* is binary or nominal:

    $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$, or $d_{ij}^{(f)} = 1$ otherwise
  - *f* is interval-based: use the normalized distance (min-max)
  - *f* is ordinal or ratio-scaled
    - compute ranks r_{if} and $z_{if} = \frac{r_{if} - 1}{M_f - 1}$
    - and treat z_{if} as interval-scaled

Major Clustering Approaches

- Hierarchical algorithms: Create a hierarchical decomposition of the set of data (or objects) using some criterion
- Partitioning algorithms: Construct random partitions and then iteratively refine them by some criterion
- 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

Cluster Parameters

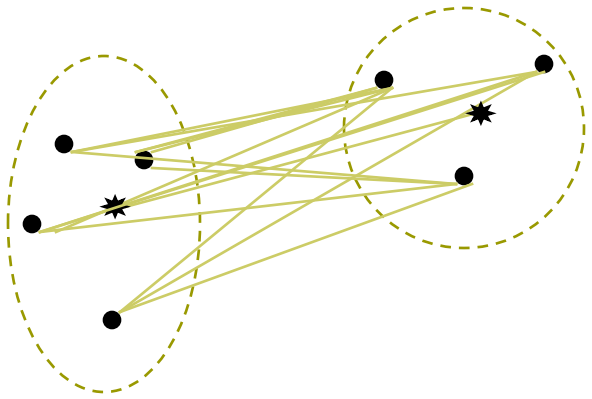
$$centroid = C_m = \frac{\sum_{i=1}^{N}(t_{mi})}{N}$$

$$radius = R_m = \sqrt{\frac{\sum_{i=1}^{N}(t_{mi} - C_m)^2}{N}}$$

$$diameter = D_m = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{N}(t_{mi} - t_{mj})^2}{(N)(N-1)}}$$

Distance Between Clusters

- ***Single Link***: smallest distance between points
- ***Complete Link:*** largest distance between points
- ***Average Link:*** average distance between points
- ***Centroid:*** distance between centroids

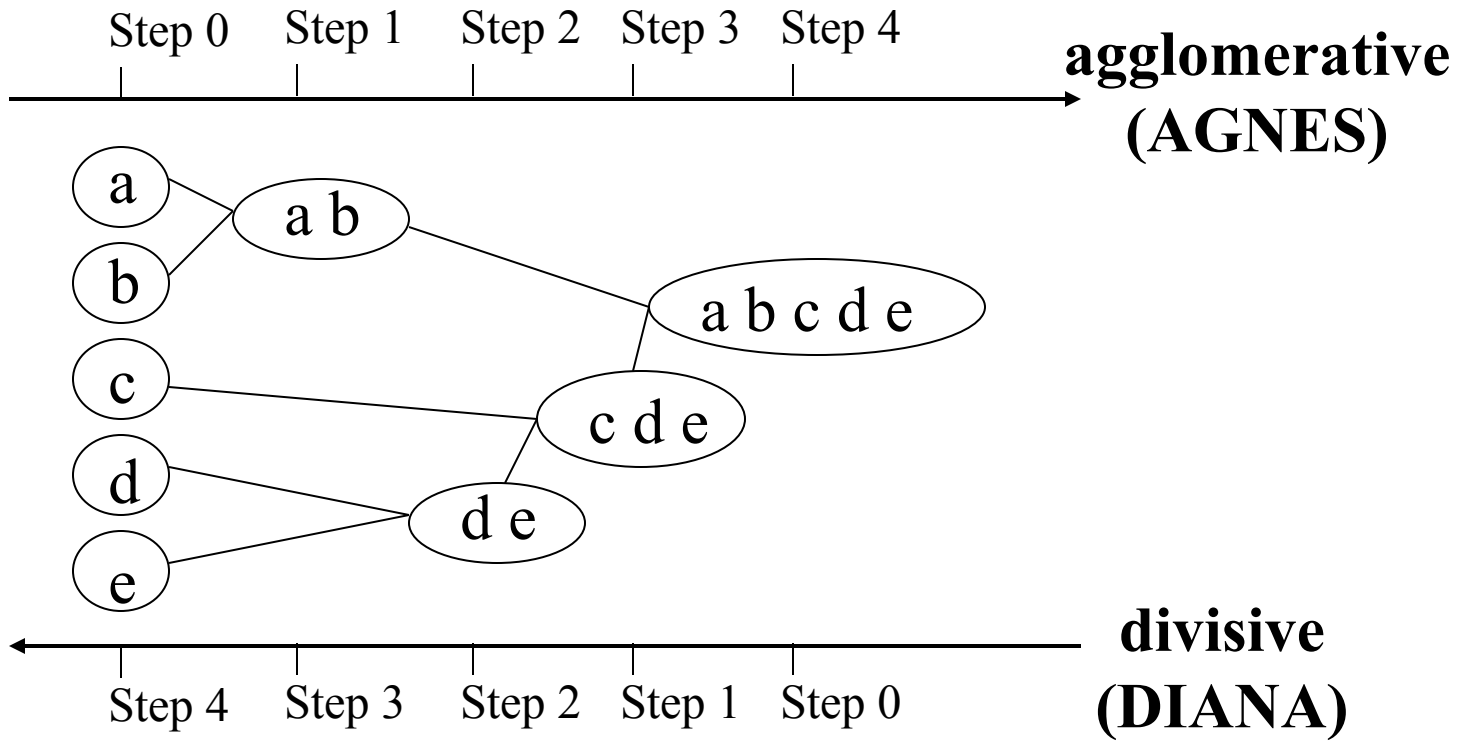


Hierarchical Clustering

- Clusters are created in levels actually creating sets of clusters at each level.
- ***Agglomerative***
  - Initially each item in its own cluster
  - Iteratively clusters are merged together
  - Bottom Up
- ***Divisive***
  - Initially all items in one cluster
  - Large clusters are successively divided
  - Top Down

Hierarchical Clustering

- Hierarchical clustering does not require the number of clusters ***k*** as an input, but needs a termination condition

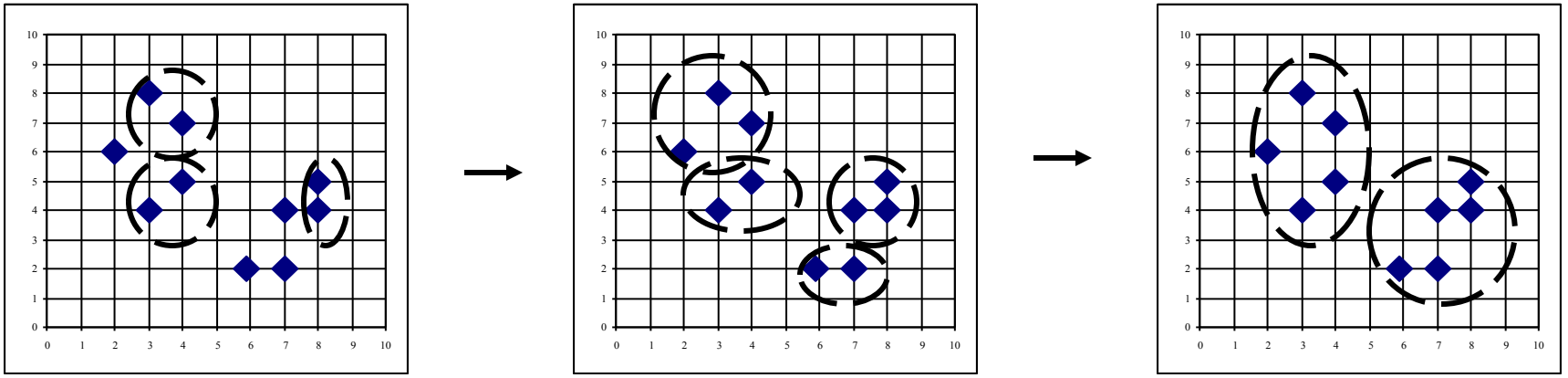


Hierarchical Algorithms

- Single Link
- MST Single Link
- Complete Link
- Average Link

AGNES (Agglomerative Nesting)

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



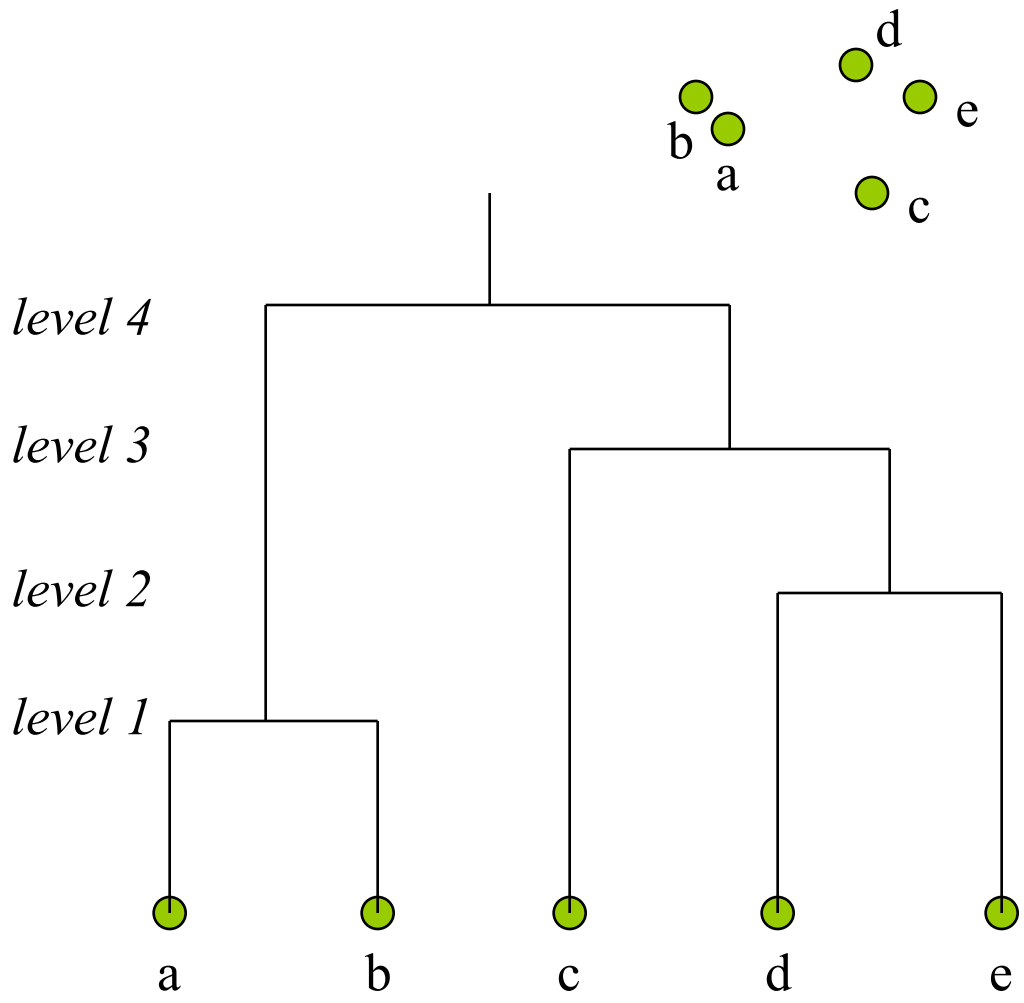
- Single-Link: each time merge the clusters (C_1,C_2) which are connected by the *shortest single link* of objects, i.e., $\min_{p \in C1, q \in C2} \text{dist}(p,q)$

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.

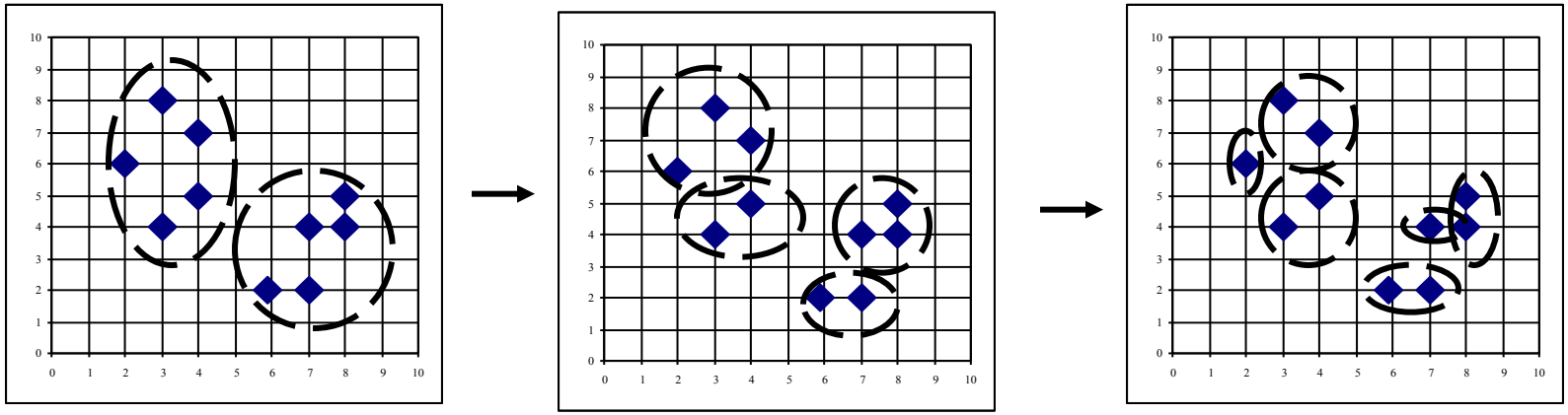
A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.

E.g., level 1 gives 4 clusters:
{a,b},{c},{d},{e},
level 2 gives 3 clusters:
{a,b},{c},{d,e}
level 3 gives 2 clusters:
{a,b},{c,d,e}, etc.



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



More on Hierarchical Clustering Methods

- 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
  - can never undo what was done previously
- Integration of hierarchical with distance-based clustering
  - BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters
  - CURE (1998): selects well-scattered points from the cluster and then shrinks them towards the center of the cluster by a specified fraction
  - CHAMELEON (1999): hierarchical clustering using dynamic modeling

Partitioning Algorithms: Basic Concepts

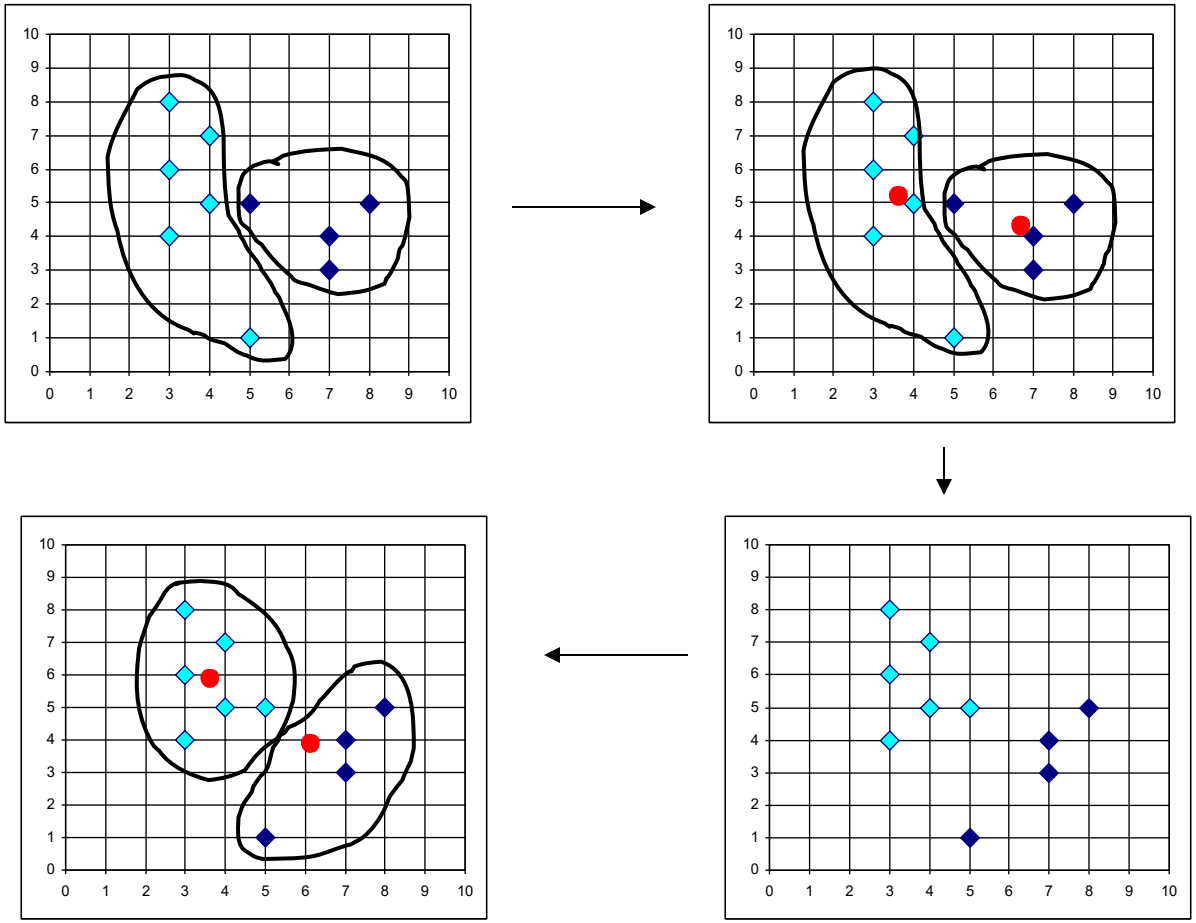
- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means* (MacQueen'67): 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 is implemented in 4 steps:
  1. Partition objects into *k* nonempty subsets
  2. Compute seed points as the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
  3. Assign each object to the cluster with the nearest seed point.
  4. Go back to Step 2, stop when no more new assignment.

The k-means Clustering Method

- Example



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 (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*

Variations of the k-means Method

- A few variants of the *k-means* which differ in
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means

The 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
- *CLARA* (Kaufmann & Rousseeuw, 1990)
- *CLARANS* (Ng & Han, 1994): Randomized sampling

PAM (Partitioning Around Medoids)

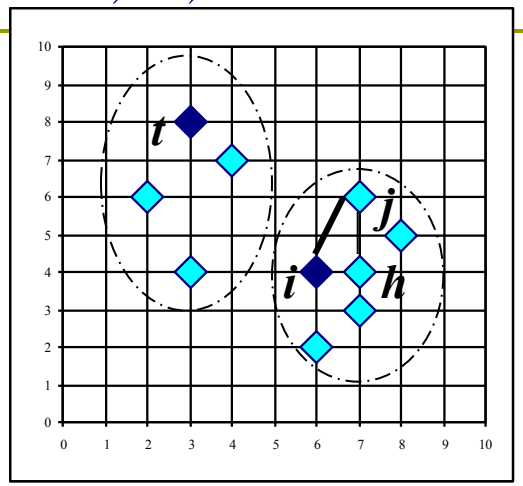
- PAM (Kaufman and Rousseeuw, 1987), built in statistical package S+
- Use real object to represent the cluster
  1. Select ***k*** representative objects arbitrarily
  2. For each pair of non-selected object ***h*** and selected object ***i***, calculate the total swapping cost $\boldsymbol{TC_{ih}}$
  3. Find the pair of ***i*** and ***h***, for which TC_{ih} is the smallest
  4. If $TC_{ih} < 0$
     - replace ***i*** by ***h***
     - assign each non-selected object to the closest representative object
     - Goto 3

PAM Clustering: Total swapping cost $TC_{ih}=\sum_j C_{jih}$

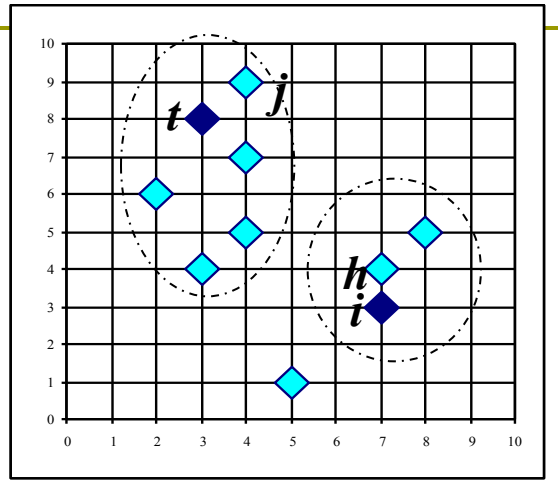
- i is a current medoid, h is a non-selected object
- Assume that i is replaced by h in the set of medoids
- $TC_{ih} = 0$;
- For each non-selected object $j \neq h$:
  - TC_{ih} += $d(j,new\_med_j)-d(j,prev\_med_j)$:
    - new_med_j = the closest medoid to j after i is replaced by h
    - $prev\_med_j$ = the closest medoid to j before i is replaced by h

PAM Clustering: Total swapping cost

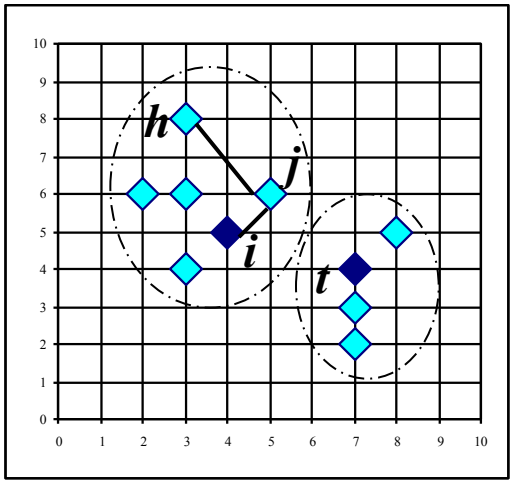
$TC_{ih} = \sum_j C_{jih}$



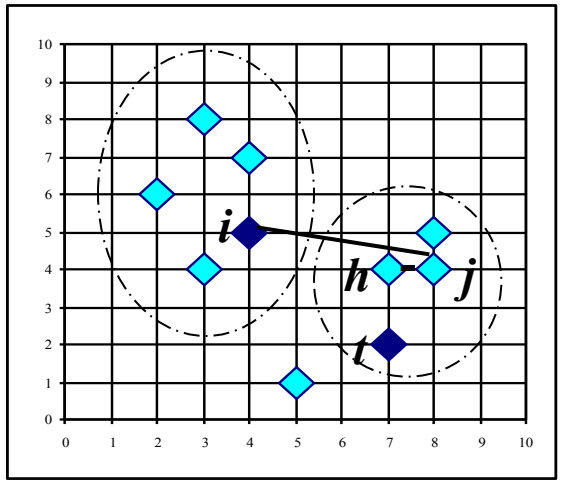
$C_{jih} = d(j, h) - d(j, i)$



$C_{jih} = 0$



$C_{jih} = d(j, t) - d(j, i)$



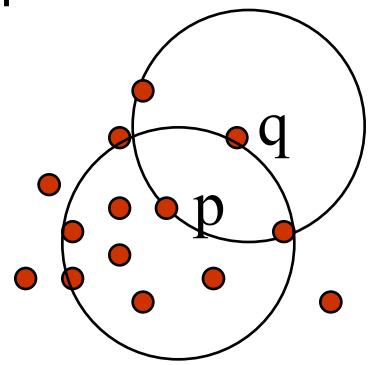
$C_{jih} = d(j, h) - d(j, t)$

Density-Based Clustering Methods

- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98)

Density-Based Clustering: Background

- Neighborhood of point p=all points within distance Eps from p:
  - $N_{Eps}(p)=\{q \mid dist(p,q) <= Eps\}$
- Two parameters***:***
  - ***Eps***: Maximum radius of the neighborhood
  - ***MinPts***: Minimum number of points in an Eps-neighborhood of that point
- If the number of points in the Eps-neighborhood of p is at least ***MinPts***, then p is called a core object.
- If an object q is not a core point, but it belongs to the Eps-neighborhood of a core point, then q is a border object.

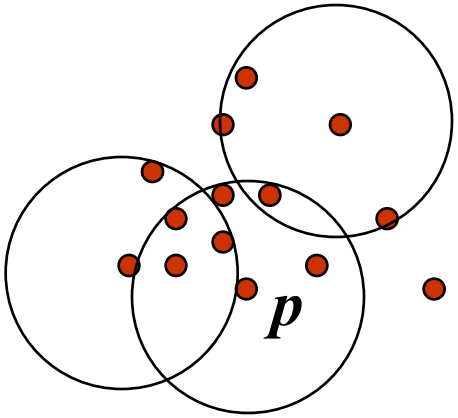


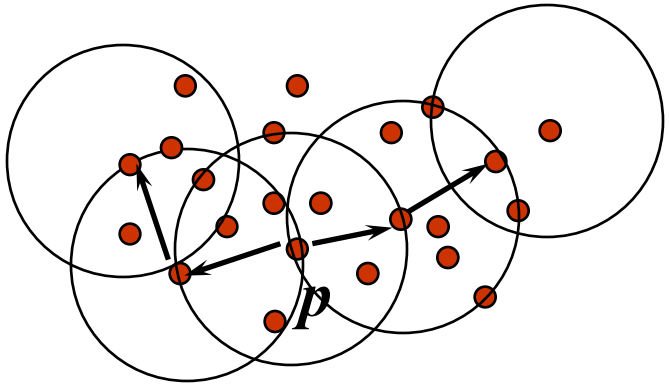
MinPts = 5

Eps = 1 cm

Density-Based Clustering: Background

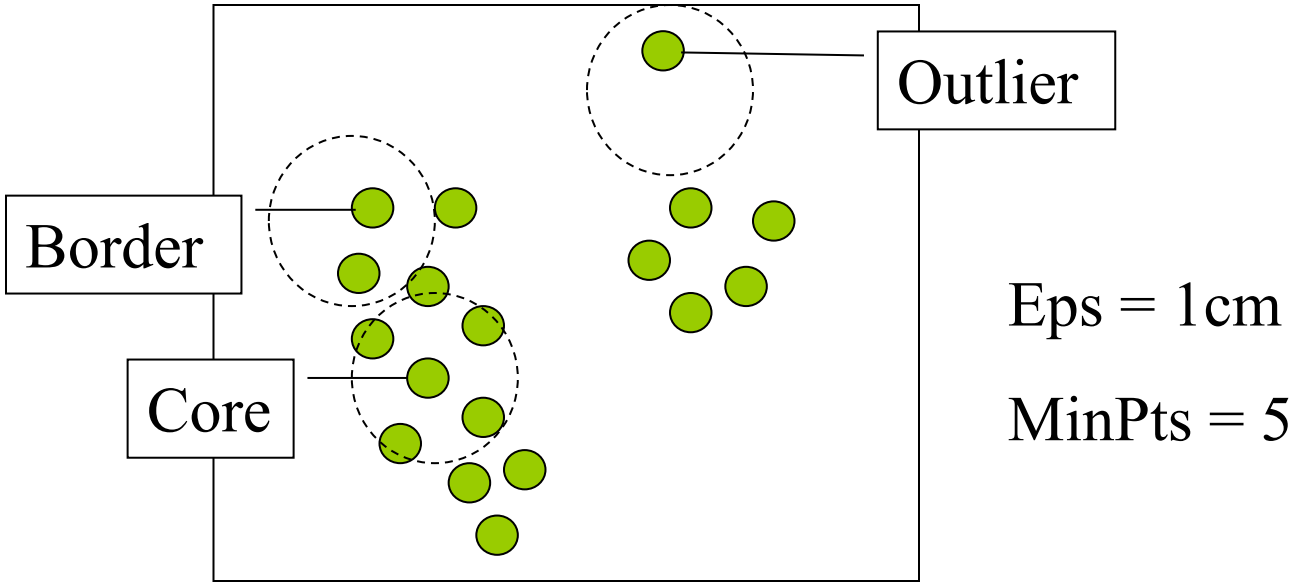
- A core point and its Eps-neighborhood define a cluster.
- If two core points p and q belong to the Eps-neighborhood of each other, the corresponding clusters are merged.





DBSCAN: Density Based Spatial Clustering of Applications with Noise

- Discovers clusters of arbitrary shape in spatial databases with noise



DBSCAN: The Algorithm

1. Select an unprocessed point ***p***
2. Find Eps-Neighborhood of ***p*** using parameter ***Eps***.
3. If ***p*** is a core point (based on ***MinPts***), a cluster is formed
   - Put all points in Eps-Neighborhood of ***p*** in a queue Q and examine the points in Q whether they are core points; expand current cluster accordingly
4. Otherwise leave ***p*** unlabeled (***p*** may be included to a cluster later if it is found to be in the Eps-Neighborhood of a core point; if not, becomes an outlier)
5. If there are more unprocessed points goto 1

Summary

- Cluster analysis groups objects based on their similarity and has wide applications
- Measure of similarity can be computed for various types of data
- Clustering algorithms can be categorized into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods
- Outlier detection and analysis are very useful for fraud detection, etc. and can be performed by statistical, distance-based or deviation-based approaches