

Information Processing Technology of Internet of Things

Chapter 2 Data Mining


Wu Liu

Beijing Key Lab of Intelligent Telecomm. Software and Multimedia
Beijing University of Posts and Telecommunications

2.3 Cluster Analysis



Outline

- Cluster Analysis: Basic Concepts 
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of 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: Application Examples

- Biology: taxonomy of living things, phylum, class, order, family, genus and species
- Information retrieval: document clustering
- 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 research

Basic Steps to Develop a Clustering Task

- Feature selection
 - Select info concerning the task of interest
 - Minimal information redundancy
 - Proximity measure
 - Similarity of two feature vectors
 - Clustering criterion
 - Expressed via a cost function or some rules
 - Clustering algorithms
 - Choice of algorithms
 - Validation of the results
 - Validation test (also, *clustering tendency* test)
 - Interpretation of the results
 - Integration with applications
-

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 rather different for interval-scaled, boolean, categorical, ordinal ratio, 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
 - 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
- Grid-based approach:
 - based on a multiple-level granularity structure
 - Typical methods: STING, WaveCluster, CLIQUE



Outline

- Cluster Analysis: Basic Concepts
- Partitioning Methods 
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering



Partitioning Algorithms: Basic Concept

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

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

- 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*: Each cluster is represented by the center of the cluster
 - *k-medoids* or PAM (Partition around medoids) : Each cluster is represented by one of the objects in the cluster

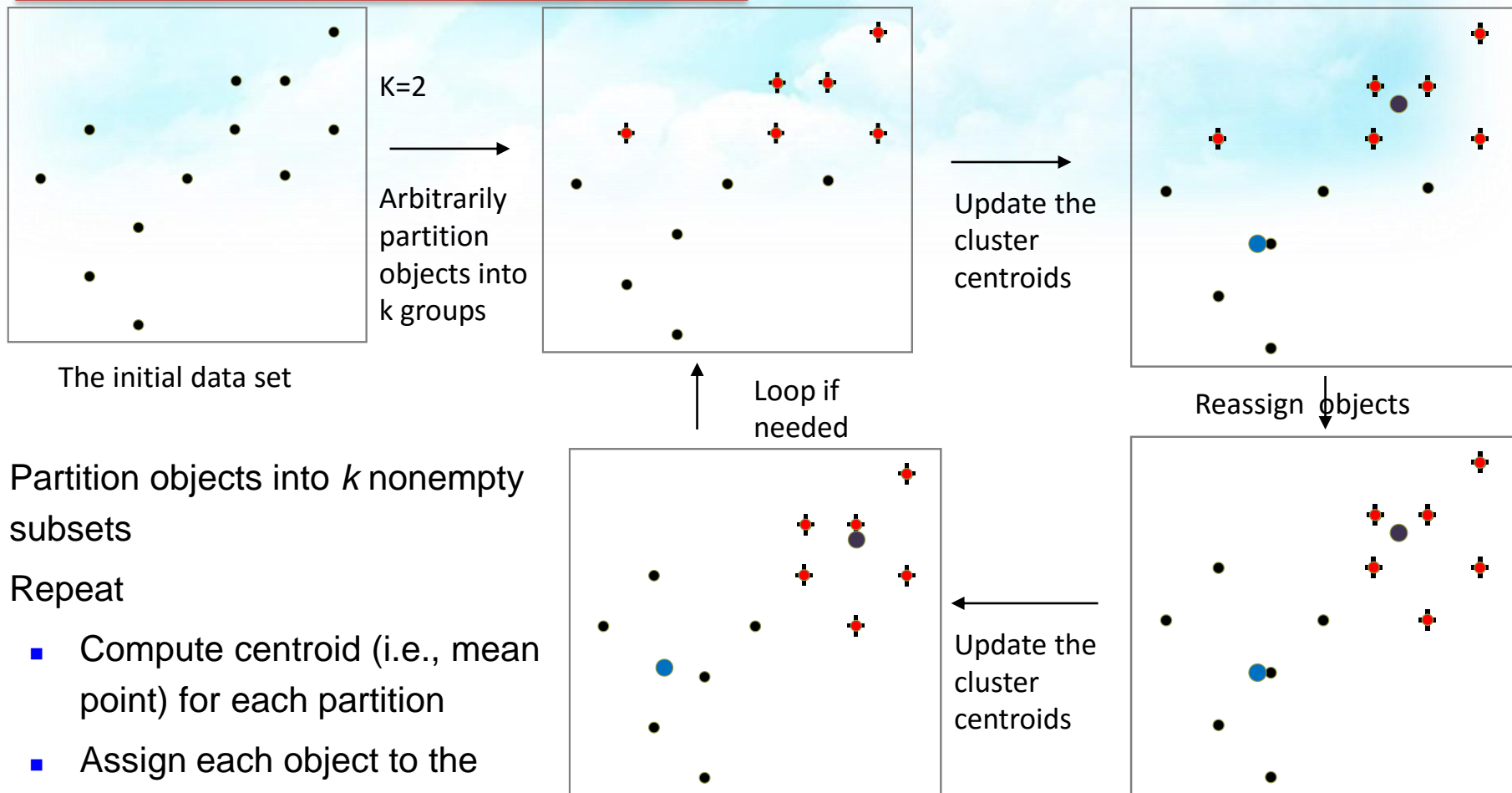


The K -Means Clustering Method

- Given k , the k -means algorithm is implemented in four steps:
 - Partition objects into k nonempty subsets
 - Compute seed points as the centroids of the clusters of the current partitioning (the centroid is the center, i.e., *mean point*, of the cluster)
 - Assign each object to the cluster with the nearest seed point
 - Go back to Step 2, stop when the assignment does not change



An Example of K -Means Clustering



- Partition objects into k nonempty subsets
- Repeat
 - Compute centroid (i.e., mean point) for each partition
 - Assign each object to the cluster of its nearest centroid
- Until no change



K-Means Clustering Algorithm

Algorithm: *k*-means. The *k*-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

Input:

- *k*: the number of clusters,
- *D*: a data set containing *n* objects.

Output: A set of *k* clusters.

Method:

- (1) arbitrarily choose *k* objects from *D* as the initial cluster centers;
- (2) repeat
- (3) (re)assign each object to the cluster to which the object is the most similar,
 based on the mean value of the objects in the cluster;
- (4) update the cluster means, that is, calculate the mean value of the objects for
 each cluster;
- (5) until no change;



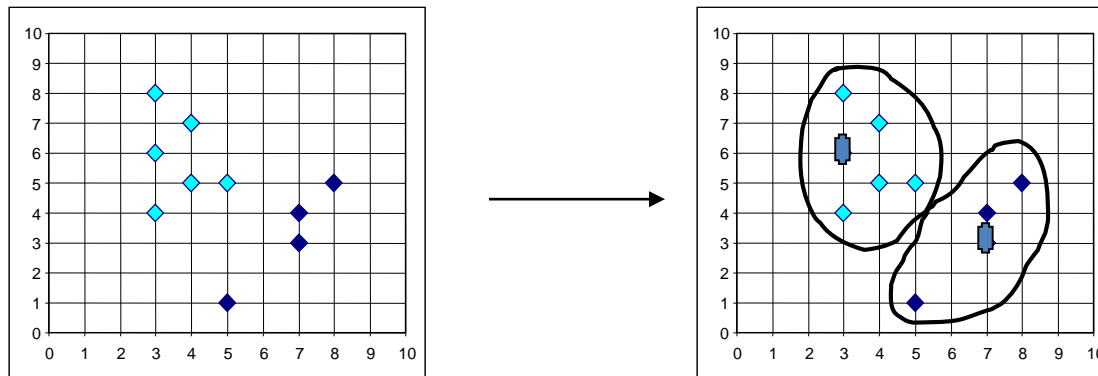
Comments on the \mathcal{K} -Means Method

- Strength: *Efficient*: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, $k, t \ll n$.
- Comment: Often terminates at a *local optimal*
- Weakness
 - Applicable only to objects in a **continuous n-dimensional space**
 - Using the k-modes method for categorical data
 - In comparison, k-medoids can be applied to a wide range of data
 - Need to **specify k** , the *number* of clusters, in advance (there are ways to automatically determine the best k)
 - Sensitive to noisy data and *outliers*
 - Not suitable to discover clusters with *non-convex shapes*



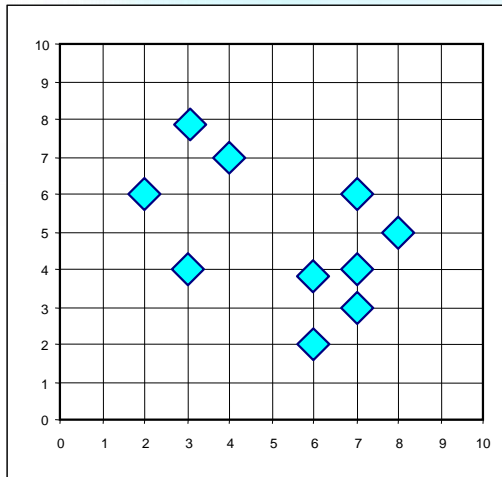
What Is the Problem of the K-Means Method

- The k-means algorithm is sensitive to outliers !
 - Since an object with an extremely large value may substantially distort the distribution of the data
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster. (medoids are always members of the data set)

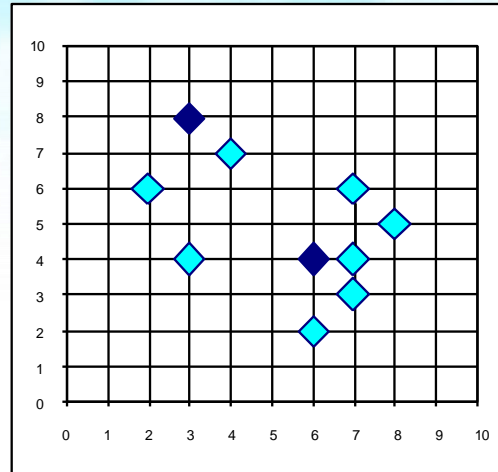


PAM: A Typical K-Medoids Algorithm

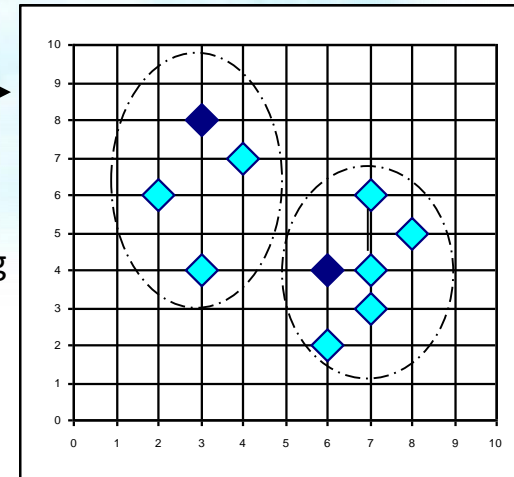
Total Cost = 20



Arbitrary
choose k
object as
initial
medoids



Assign
each
remaining
object to
nearest
medoids



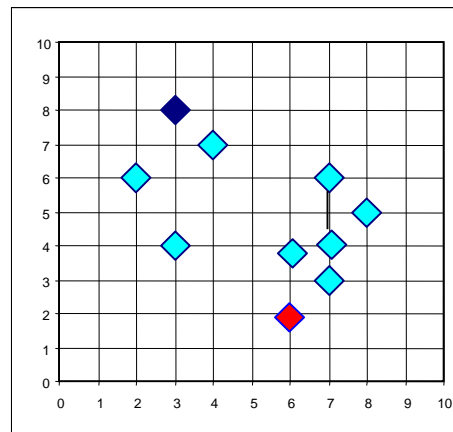
$K=2$

Randomly select a
nonmedoid object, O_{random}

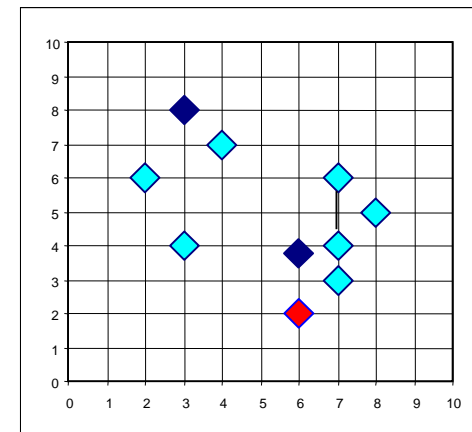
Do loop
Until no change

Swapping O
and O_{random}
If quality is
improved.

Total Cost = 26



Compute
total cost of
swapping



The K-Medoid Clustering Method

- *K-Medoids* Clustering: Find *representative* objects (medoids) in clusters
 - *PAM*
 - 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 (due to the computational complexity)
- Efficiency improvement on PAM
 - *CLARA*: PAM on samples



Algorithm: k -medoids. PAM, a k -medoids algorithm for partitioning based on medoid or central objects.

Input:

- k : the number of clusters,
- D : a data set containing n objects.


Output: A set of k clusters.

Method:

- (1) arbitrarily choose k objects in D as the initial representative objects or seeds;
- (2) **repeat**
- (3) assign each remaining object to the cluster with the nearest representative object;
- (4) randomly select a nonrepresentative object, o_{random} ;
- (5) compute the total cost, S , of swapping representative object, o_j , with o_{random} ;
- (6) **if** $S < 0$ **then** swap o_j with o_{random} to form the new set of k representative objects;
- (7) **until** no change;



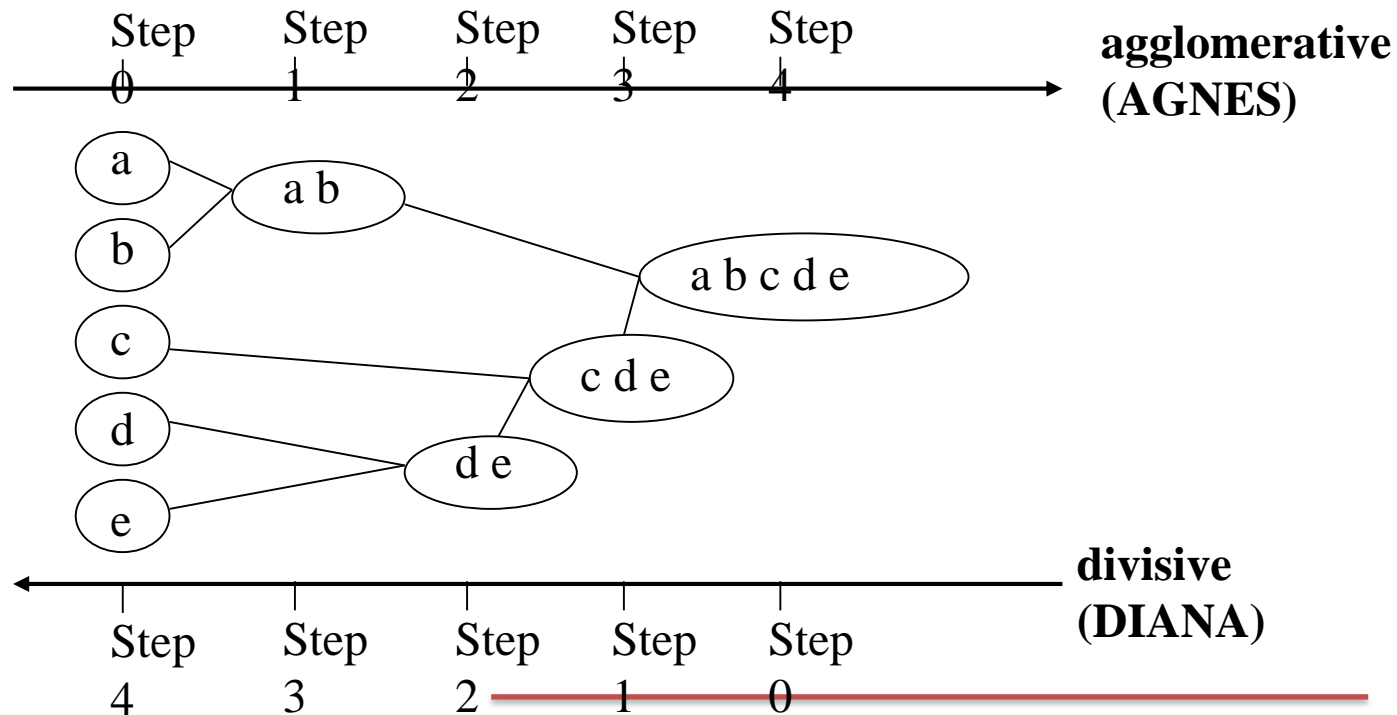
Outline

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods 
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering



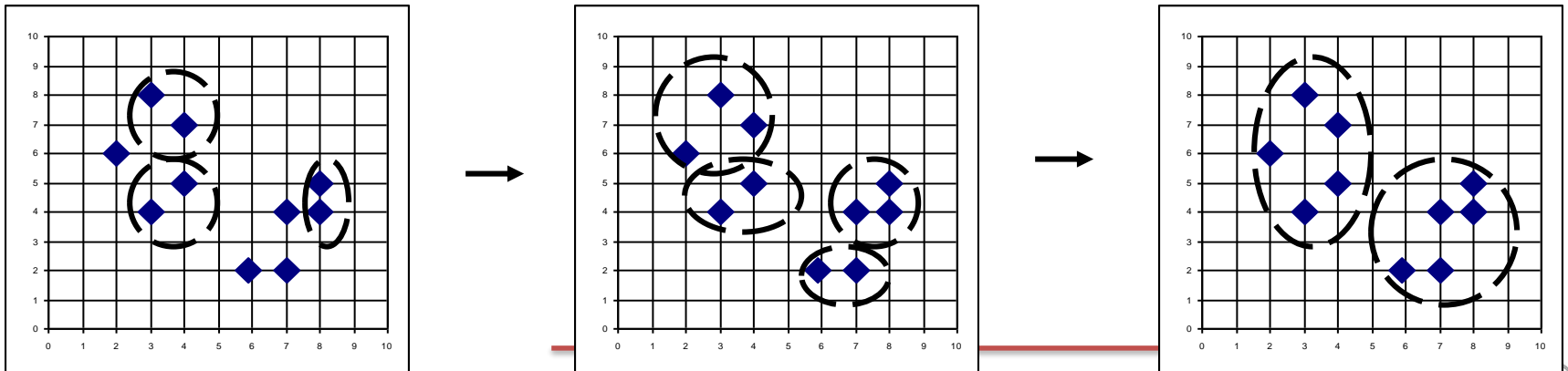
Hierarchical Clustering

- A **hierarchical clustering** method works by grouping data objects into a hierarchy or “tree” of clusters.
- This method does not require the number of clusters k as an input, but needs a termination condition
- Agglomerative versus divisive hierarchical clustering, which organize objects into a hierarchy using a bottom-up or top-down strategy, respectively.



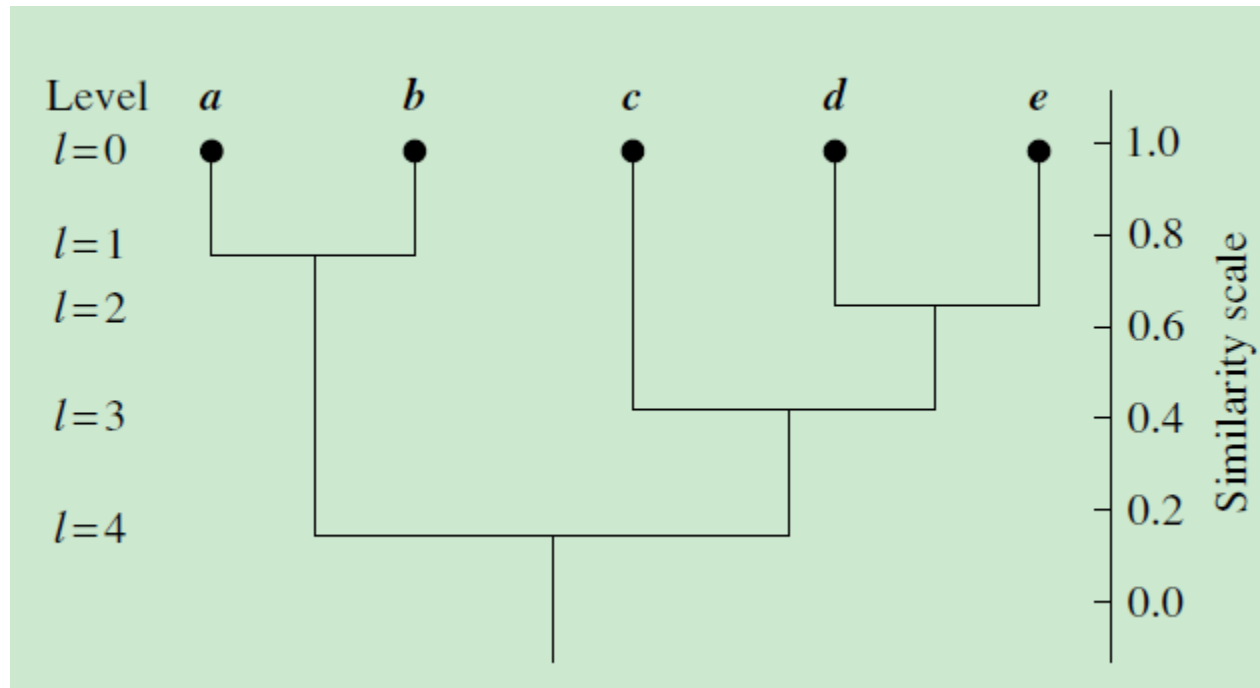
AGNES (Agglomerative Nesting)

- **Agglomerative methods** start with individual objects as clusters, which are iteratively merged to form larger clusters, until all the objects are in a single cluster or certain termination conditions are satisfied.
- For the merging step, it finds the two clusters that are closest to each other (according to some **similarity measure**), and combines the two to form one cluster.
- Use the **single-link method**: each cluster is represented by all the objects in the cluster, and the similarity between two clusters is measured by the similarity of the closest pair of data points belonging to different clusters.
- Eventually all nodes belong to the same cluster



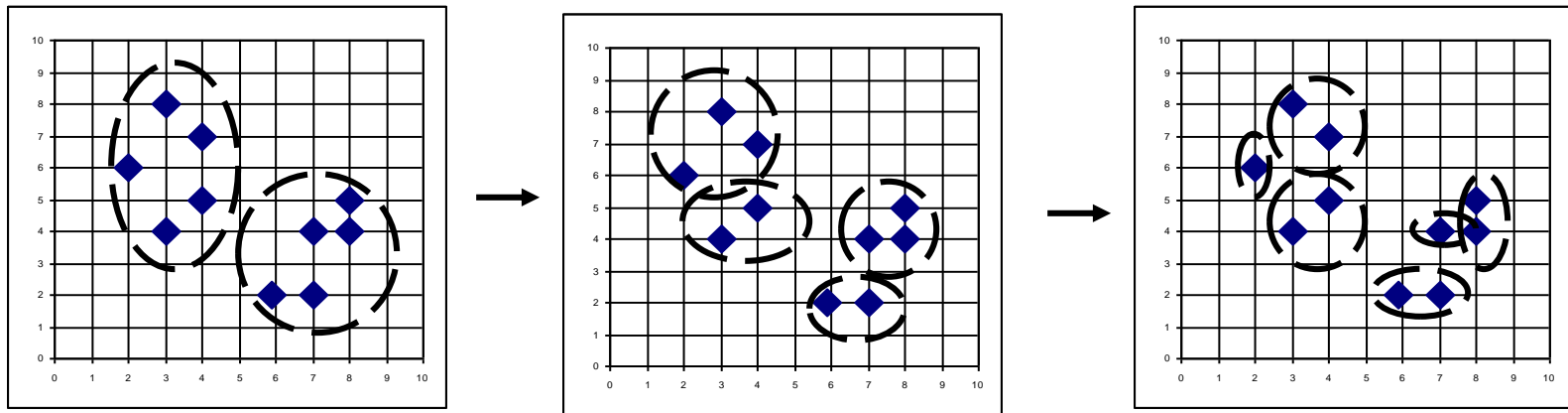
Dendrogram: Shows How Clusters are Merged

- Go on in a non-descending fashion
- 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

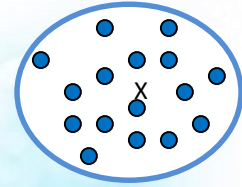
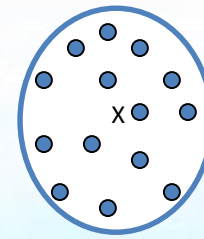


DIANA (Divisive Analysis)

- **Divisive methods** initially let all the given objects form one cluster, which they iteratively split into smaller clusters.
- Inverse order of AGNES
- The cluster is **split according to some principle** such as the maximum Euclidean distance between the closest neighboring objects in the cluster.
- Eventually each node forms a cluster on its own, or the objects within a cluster are sufficiently similar to each other.



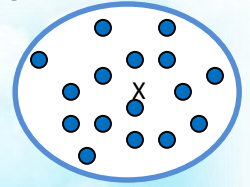
Distance between Clusters



- **Single link:** smallest distance between an element in one cluster and an element in the other, i.e., $\text{dist}(K_i, K_j) = \min(t_{ip}, t_{jq})$
- **Complete link:** largest distance between an element in one cluster and an element in the other, i.e., $\text{dist}(K_i, K_j) = \max(t_{ip}, t_{jq})$
- **Average:** avg distance between an element in one cluster and an element in the other, i.e., $\text{dist}(K_i, K_j) = \text{avg}(t_{ip}, t_{jq})$
- **Centroid:** distance between the centroids of two clusters, i.e., $\text{dist}(K_i, K_j) = \text{dist}(C_i, C_j)$
- **Medoid:** distance between the medoids of two clusters, i.e., $\text{dist}(K_i, K_j) = \text{dist}(M_i, M_j)$
 - Medoid: a chosen, centrally located object in the cluster



Centroid, Radius and Diameter of a Cluster (for numerical data sets)



- Centroid: the “middle” of a cluster

$$C_m = \frac{\sum_{i=1}^N (t_i)}{N}$$

- Radius: the average distance from member objects to the centroid

$$R_m = \sqrt{\frac{\sum_{i=1}^N (t_i - c_m)^2}{N}}$$

- Diameter: the average pairwise distance within a cluster

$$D_m = \sqrt{\frac{\sum_{i=1}^N \sum_{j=1}^N (t_i - t_j)^2}{N(N-1)}}$$




Extensions to Hierarchical Clustering

- Major weakness of agglomerative clustering methods
 - 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
- Integration of hierarchical & distance-based clustering
 - BIRCH: uses CF-tree and incrementally adjusts the quality of sub-clusters
 - CHAMELEON: hierarchical clustering using dynamic modeling



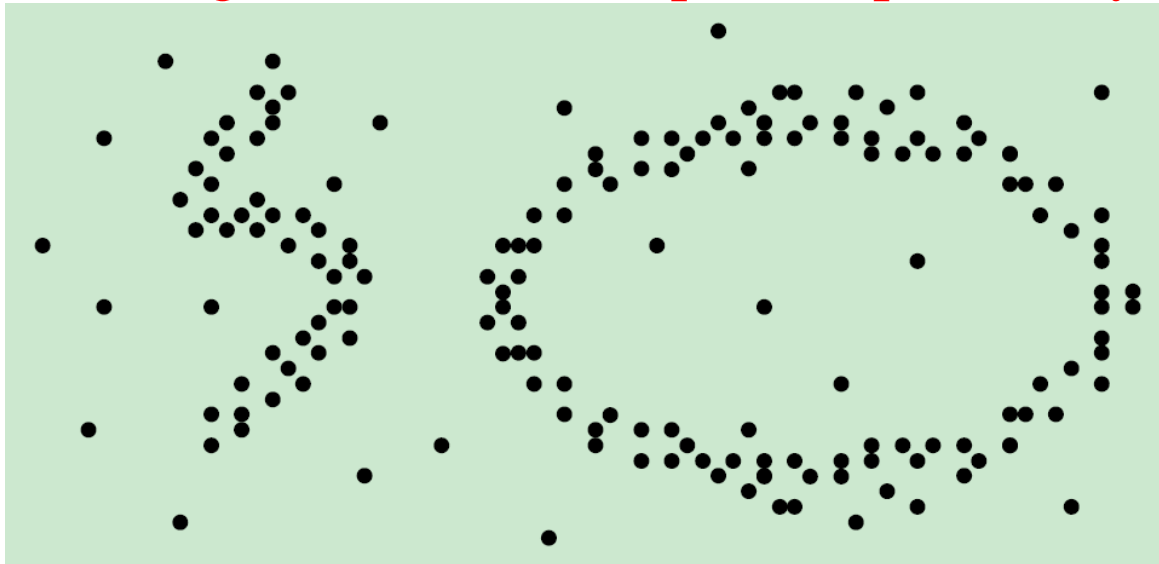
Outline

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods 
- Grid-Based Methods
- Evaluation of Clustering



Density-Based Clustering Methods

- Partitioning and hierarchical methods are designed to find spherical-shaped clusters.
- Given such data, they would likely inaccurately identify convex regions, where **noise or outliers** are included in the clusters.
- To find clusters of arbitrary shape, we can model clusters as **dense regions in the data space, separated by sparse regions.**



Clusters of arbitrary shape



Density-Based Clustering Methods

- Clustering based on density (local cluster criterion), such as **density-connected points**
- The *density of an object* o can be measured by the number of objects close to o .
- Major features:
 - Discover clusters of arbitrary shape
 - Handle noise
 - One scan
 - Need density parameters as termination condition
- Several interesting studies:
 - DBSCAN
 - OPTICS
 - DENCLUE
 - CLIQUE (more grid-based)



Density-Based Clustering: Basic Concepts

- Two parameters:
 - *Eps*: Maximum radius of the neighbourhood
 - *MinPts*: Minimum number of points in an Eps-neighbourhood of that point
- The ϵ -neighborhood of an object o is the space within a radius centered at o . The density of a neighborhood can be measured simply by the number of objects in the neighborhood.
- $N_{Eps}(q)$: $\{p \text{ belongs to } D \mid \text{dist}(p,q) \leq Eps\}$ (a set, D , of objects)
- An object is a **core object** if the Eps-neighborhood of the object contains at least *MinPts* objects.
- The clustering task is therein reduced to using **core objects and their neighborhoods** to form dense regions, where the dense regions are clusters.

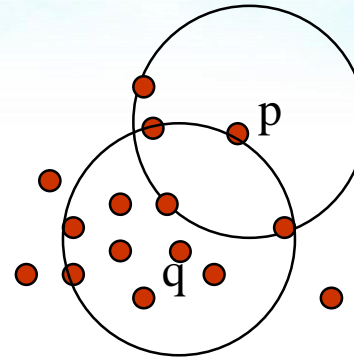


Density-Based Clustering: Basic Concepts

- **Directly density-reachable**: A point p is directly density-reachable from a point q w.r.t. *Eps*, *MinPts* if

- p belongs to $N_{Eps}(q)$
- **core point condition:**

$$|N_{Eps}(q)| \geq MinPts$$



MinPts = 5

Eps = 1 cm

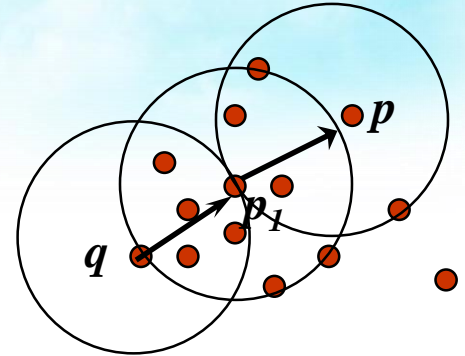
- Using the directly density-reachable relation, a core object can “bring” all objects from its Eps-neighborhood into a dense region.



Density-Reachable and Density-Connected

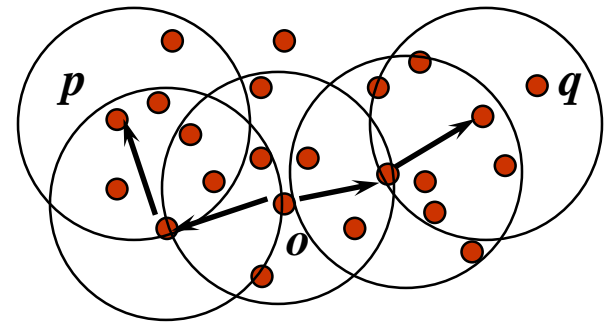
■ Density-reachable:

- A point p is **density-reachable** from a point q w.r.t. Eps , $MinPts$ if there is a chain of points p_1, \dots, p_n , $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i



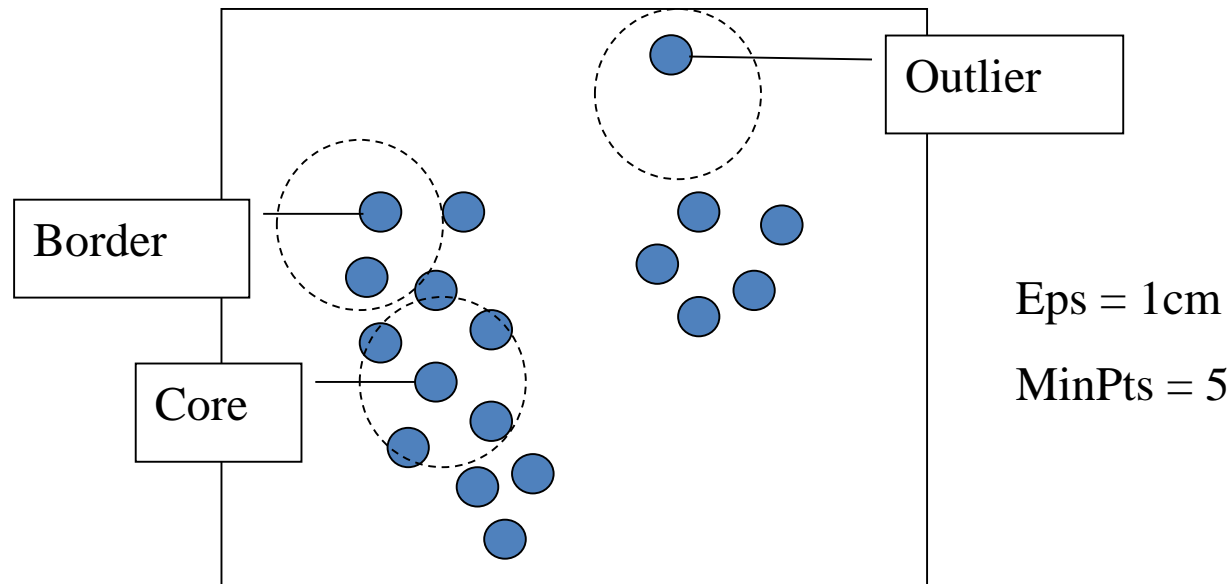
■ Density-connected

- A point p is **density-connected** to a point q w.r.t. Eps , $MinPts$ if there is a point o such that both, p and q are density-reachable from o w.r.t. Eps and $MinPts$



DBSCAN: Density-Based Spatial Clustering of Applications with Noise

- **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise) finds *core objects*, that is, objects that have dense neighborhoods. It connects core objects and their neighborhoods to form *dense regions as clusters*.
- Relies on a *density-based* cluster: *A cluster is defined as a maximal set of density-connected points*
- Discovers clusters of arbitrary shape in spatial databases with noise



Algorithm: DBSCAN: a density-based clustering algorithm.

Input:

- D : a data set containing n objects,
- ϵ : the radius parameter, and
- $MinPts$: the neighborhood density threshold.

Output: A set of density-based clusters.

Method:

- (1) mark all objects as **unvisited**;
- (2) **do**
- (3) randomly select an unvisited object p ;
- (4) mark p as **visited**;
- (5) **if** the ϵ -neighborhood of p has at least $MinPts$ objects
- (6) create a new cluster C , and add p to C ;
- (7) let N be the set of objects in the ϵ -neighborhood of p ;
- (8) **for** each point p' in N
- (9) **if** p' is **unvisited**
- (10) mark p' as **visited**;
- (11) **if** the ϵ -neighborhood of p' has at least $MinPts$ points,
 add those points to N ;
- (12) **if** p' is not yet a member of any cluster, add p' to C ;
- (13) **end for**
- (14) output C ;
- (15) **else** mark p as **noise**;
- (16) **until** no object is **unvisited**;



DBSCAN: Sensitive to Parameters

Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.

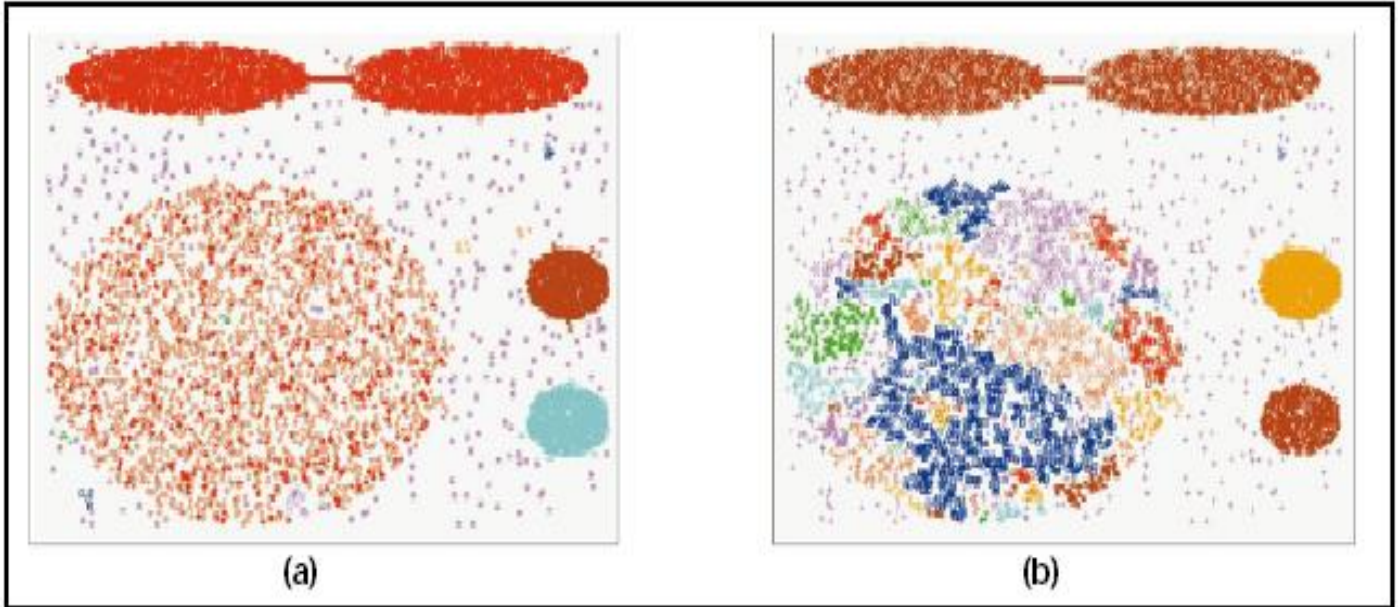
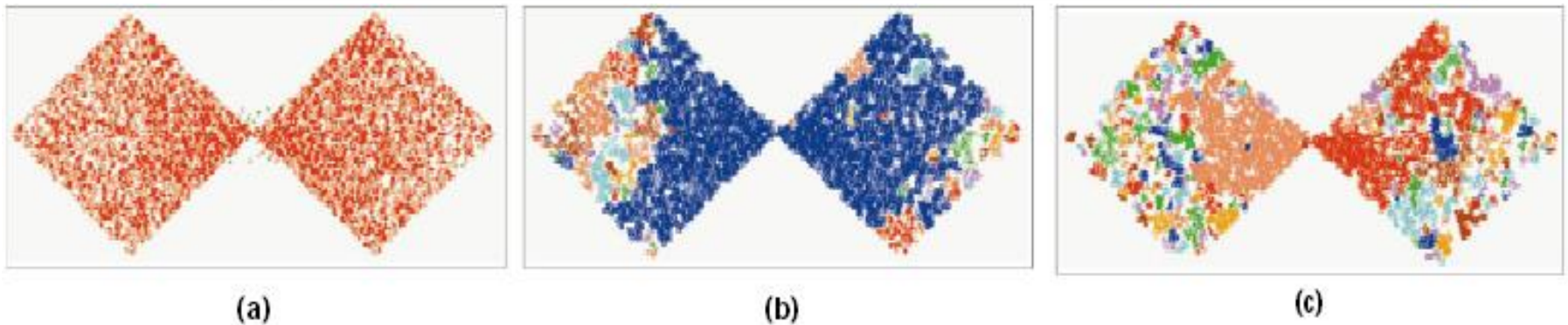



Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



Outline

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods 
- Evaluation of Clustering



Grid-Based Clustering Method

- A **grid-based clustering** method takes a space-driven approach by partitioning the embedding space into cells independent of the distribution of the input objects.
- Using multi-resolution grid data structure
- quantizes the object space into a finite number of cells, and all of the operations for clustering are performed on the grid structure
- fast processing time, which is dependent on only the number of cells
- Several interesting methods
 - **CLIQUE**:
 - Both grid-based and subspace clustering
 - **STING** (a S**T**atistical **I**Nformation Grid approach)
 - **WaveCluster**
 - A multi-resolution clustering approach using wavelet method



CLIQUE (Clustering In QUES)

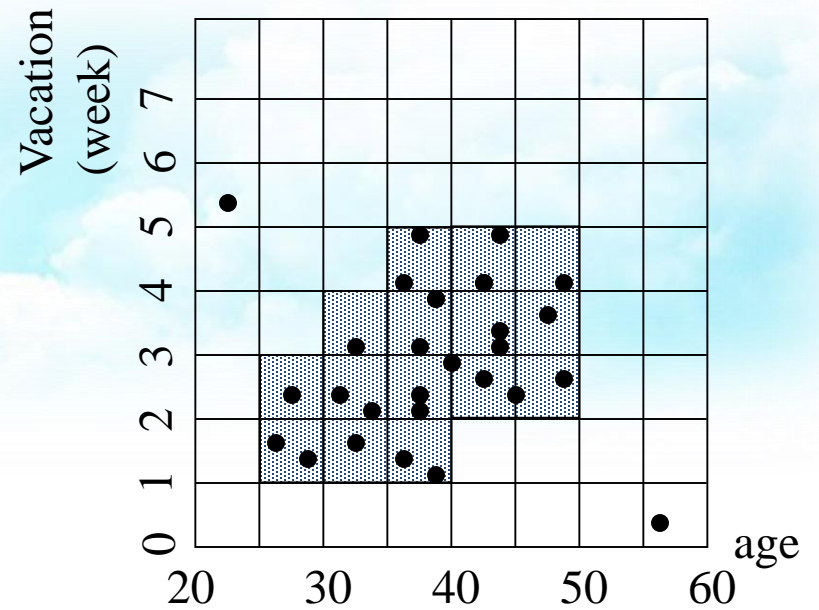
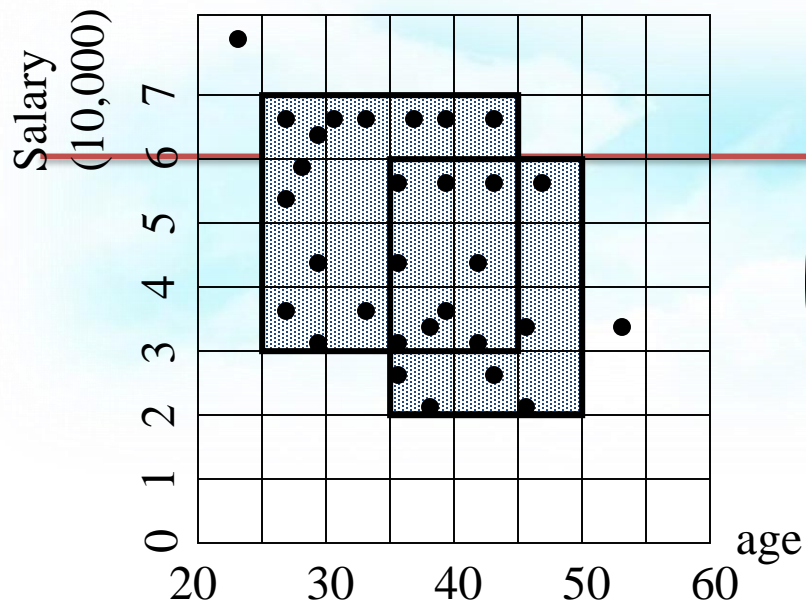
- Automatically identifying subspaces of a high dimensional data space that allow better clustering than original space
- CLIQUE can be considered as both density-based and grid-based
 - It partitions each dimension into the same number of equal length interval
 - It partitions an m-dimensional data space into non-overlapping rectangular units
 - A unit is dense if the fraction of total data points contained in the unit exceeds the input model parameter
 - A cluster is a maximal set of connected dense units within a subspace



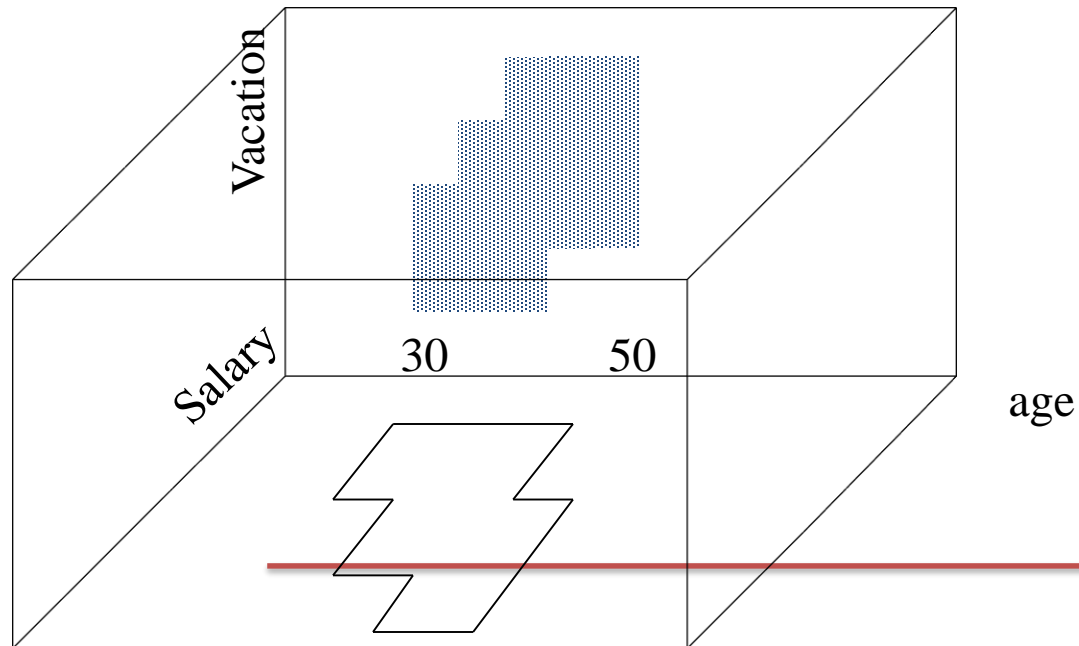
CLIQUE: The Major Steps

- Partition the data space and find the number of points that lie inside each cell of the partition.
 - Identify the subspaces that contain clusters using the Apriori principle
 - Identify clusters
 - Determine dense units in all subspaces of interests
 - Determine connected dense units in all subspaces of interests.
 - Generate minimal description for the clusters
 - Determine maximal regions that cover a cluster of connected dense units for each cluster
 - Determination of minimal cover for each cluster
-





$\tau = 3$



Strength and Weakness of CLIQUE

■ Strength


- *automatically* finds subspaces of the highest dimensionality such that high density clusters exist in those subspaces
- *insensitive* to the order of records in input and does not presume some canonical data distribution
- scales *linearly* with the size of input and has good scalability as the number of dimensions in the data increases

■ Weakness

- The accuracy of the clustering result may be degraded at the expense of simplicity of the method



Outline

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering 



Determine the Number of Clusters

- Empirical method
 - # of clusters: $k \approx \sqrt{\frac{n}{2}}$ for a dataset of n points, e.g., $n = 200$, $k = 10$
- Elbow method
 - Use the turning point in the curve of sum of within cluster variance w.r.t the # of clusters
 - plot the curve of var with respect to k . The first (or most significant) turning point of the curve suggests the “right” number.
- Cross validation method
 - Divide a given data set into m parts
 - Use $m - 1$ parts to obtain a clustering model
 - Use the remaining part to test the quality of the clustering
 - E.g., For each point in the test set, find the closest centroid, and use the sum of squared distance between all points in the test set and the closest centroids to measure how well the model fits the test set
 - For any $k > 0$, repeat it m times, compare the overall quality measure w.r.t. different k 's, and find # of clusters that best fits the data



Measuring Clustering Quality

- 2 kinds of measures: External, internal, these methods can be categorized into two groups according to whether ground truth is available.
- **External:** supervised, employ criteria not inherent to the dataset
 - Compare a clustering against prior or expert-specified knowledge (i.e., the ground truth) using certain clustering quality measure
- **Internal:** unsupervised, criteria derived from data itself
 - Evaluate the goodness of a clustering by considering how well the clusters are separated, and how compact the clusters are, e.g., Silhouette coefficient

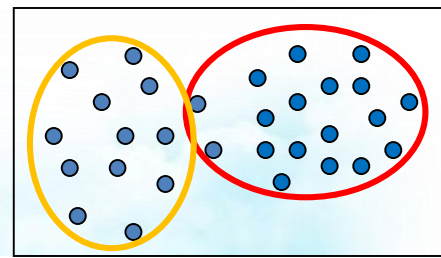


Measuring Clustering Quality: External Methods

- Clustering quality measure: $Q(C, T)$, for a clustering C given the ground truth T
- Q is good if it satisfies the following 4 essential criteria
 - **Cluster homogeneity**: the purer, the better
 - **Cluster completeness**: should assign objects belong to the same category in the ground truth to the same cluster
 - **Rag bag**: putting a heterogeneous object into a pure cluster should be penalized more than putting it into a *rag bag* (i.e., “miscellaneous” or “other” category)
 - **Small cluster preservation**: splitting a small category into pieces is more harmful than splitting a large category into pieces



Entropy-Based Measure: Conditional Entropy



- Entropy of clustering \mathcal{C} :
- Entropy of partitioning \mathcal{T} :
- Entropy of \mathcal{T} w.r.t. cluster C_i :
- Conditional entropy of \mathcal{T}

$$H(\mathcal{C}) = - \sum_{i=1}^r p_{C_i} \log p_{C_i}$$

$p_{C_i} = \frac{n_i}{n}$ the prob. of cluster C_i

$$H(\mathcal{T}) = - \sum_{j=1}^k p_{T_j} \log p_{T_j}$$

$$H(\mathcal{T}|C_i) = - \sum_{j=1}^k \left(\frac{n_{ij}}{n_i} \right) \log \left(\frac{n_{ij}}{n_i} \right)$$

w.r.t. clustering \mathcal{C} :

$$H(\mathcal{T}|\mathcal{C}) = - \sum_{i=1}^r \left(\frac{n_i}{n} \right) H(\mathcal{T}|C_i) = - \sum_{i=1}^r \sum_{j=1}^k p_{ij} \log \left(\frac{p_{ij}}{p_{C_i}} \right)$$

–The more a cluster's members are split into different partitions, the higher the conditional entropy

–For a perfect clustering, the conditional entropy value is 0, where the worst possible conditional entropy value is $\log k$

$$\begin{aligned} H(\mathcal{T}|\mathcal{C}) &= - \sum_{i=1}^r \sum_{j=1}^k p_{ij} (\log p_{ij} - \log p_{C_i}) = - \sum_{i=1}^r \sum_{j=1}^k p_{ij} \log p_{ij} + \sum_{i=1}^r (\log p_{C_i} \sum_{j=1}^k p_{ij}) \\ &= - \sum_{i=1}^r \sum_{j=1}^k p_{ij} \log p_{ij} + \sum_{i=1}^r (p_{C_i} \log p_{C_i}) = H(\mathcal{C}, \mathcal{T}) - H(\mathcal{C}) \end{aligned}$$



Measuring Clustering Quality: Internal Methods

- Many intrinsic methods have the advantage of a **similarity metric between objects** in the data set.

- **silhouette coefficient:**

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

- where

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

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

- The value of the silhouette coefficient is between -1 and 1. The value of **a(o)** reflects the **compactness** of the cluster to which o belongs. The value of **b(o)** captures the degree to which o is **separated** from other clusters. The larger b(o) is, the more separated o is from other clusters.
- When the silhouette coefficient value of o approaches 1, the cluster containing o is compact and o is far away from other clusters, which is the preferable case.

