Lecture 10
Clustering

Preview

- Introduction
- Partitioning methods
- Hierarchical methods
- Model-based methods
- Density-based methods

What is Clustering?

- Cluster: a collection of data objects
  - 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 unsupervised classification: no predefined classes
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

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
- Urban planning: Identifying groups of houses according to their house type, value, and geographical location
- Seismology: Observed earth quake epicenters should be clustered along continent faults

What Is a Good Clustering?

- A good clustering method will produce clusters with
  - High intra-class similarity
  - Low inter-class similarity
- Precise definition of clustering quality is difficult
  - Application-dependent
  - Ultimately subjective

Requirements for Clustering in Data Mining

- Scalability
- Ability to deal with different types of attributes
- Discovery of clusters with arbitrary shape
- Minimal domain knowledge required to determine input parameters
- Ability to deal with noise and outliers
- Insensitivity to order of input records
- Robustness wrt high dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability

Similarity and Dissimilarity Between Objects

- Same we used for IBL (e.g, L_p norm)
- Euclidean distance (p = 2):

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

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

Major Clustering Approaches

- Partitioning: Construct various partitions and then evaluate them by some criterion
- Hierarchical: Create a hierarchical decomposition of the set of objects using some criterion
- Model-based: Hypothesize a model for each cluster and find best fit of models to data
- Density-based: Guided by connectivity and density functions

Partitioning Algorithms

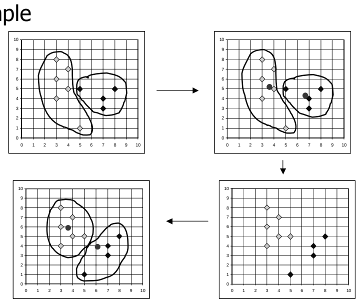
- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means* (MacQueen, 1967): Each cluster is represented by the center of the cluster
  - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw, 1987): Each cluster is represented by one of the objects in the cluster

K-Means Clustering

- Given *k*, the *k-means* algorithm consists of four steps:
  - Select initial centroids at random.
  - Assign each object to the cluster with the nearest centroid.
  - Compute each centroid as the mean of the objects assigned to it.
  - Repeat previous 2 steps until no change.

K-Means Clustering (contd.)

- Example

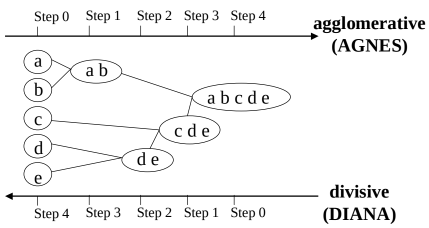


Comments on the *K-Means* Method

- Strengths
  - *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 *simulated annealing* and *genetic algorithms*
- Weaknesses
  - Applicable only when *mean* is defined (what about categorical data?)
  - Need to specify *k,* the *number* of clusters, in advance
  - Trouble with noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

Hierarchical Clustering

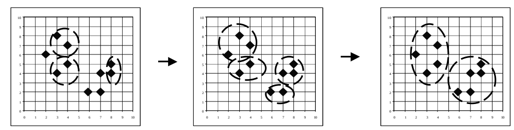
- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition



13

AGNES (Agglomerative Nesting)

- Produces tree of clusters (nodes)
- Initially: each object is a cluster (leaf)
- Recursively merges nodes that have the least dissimilarity
- Criteria: min distance, max distance, avg distance, center distance
- Eventually all nodes belong to the same cluster (root)

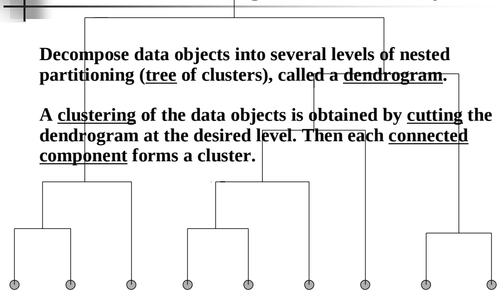


14

A *Dendrogram* Shows How the Clusters are Merged Hierarchically

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

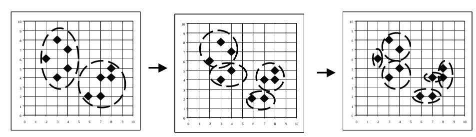
A <u>clustering</u> of the data objects is obtained by <u>cutting</u> the dendrogram at the desired level. Then each <u>connected component</u> forms a cluster.



15

DIANA (Divisive Analysis)

- Inverse order of AGNES
- Start with root cluster containing all objects
- Recursively divide into subclusters
- Eventually each cluster contains a single object



16

Other Hierarchical Clustering Methods

- Major weakness of agglomerative clustering methods
  - <u>Do not scale</u> 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
  - <u>BIRCH</u>: uses CF-tree and incrementally adjusts the quality of sub-clusters
  - <u>CURE</u>: selects well-scattered points from the cluster and then shrinks them towards the center of the cluster by a specified fraction

17

BIRCH

- BIRCH: Balanced Iterative Reducing and Clustering using Hierarchies (Zhang, Ramakrishnan & Livny, 1996)
- Incrementally construct a CF (Clustering Feature) tree
  - Parameters: max diameter, max children
  - Phase 1: scan DB to build an initial in-memory CF tree (each node: #points, sum, sum of squares)
  - Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree
- *Scales linearly*: finds a good clustering with a single scan and improves the quality with a few additional scans
- *Weaknesses:* handles only numeric data, sensitive to order of data records.

18

Clustering Feature Vector

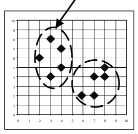
Clustering Feature: $CF = (N, \vec{LS}, SS)$

N: **Number of data points**

$LS: \sum_{i=1}^{N} \vec{X_i}$

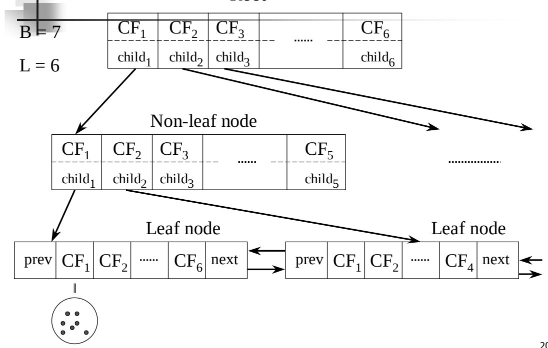
$SS: \sum_{i=1}^{N} \vec{X_i}^2$

CF = (5, (16,30),(54,190))

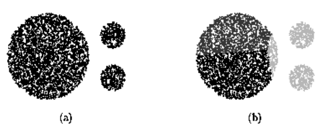


(3,4)
(2,6)
(4,5)
(4,7)
(3,8)

CF Tree

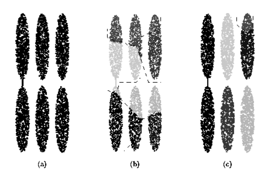


CURE (Clustering Using REpresentatives)



- CURE: non-spherical clusters, robust wrt outliers
  - Uses multiple representative points to evaluate the distance between clusters
  - Stops the creation of a cluster hierarchy if a level consists of *k* clusters

Drawbacks of Distance-Based Method



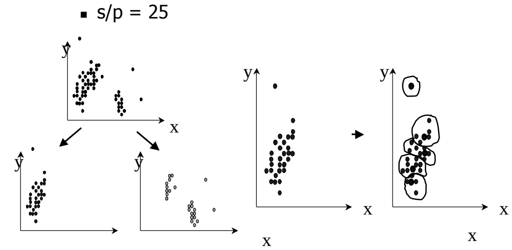
- Drawbacks of square-error-based clustering method
  - Consider only one point as representative of a cluster
  - Good only for convex clusters, of similar size and density, and if *k* can be reasonably estimated

Cure: The Algorithm

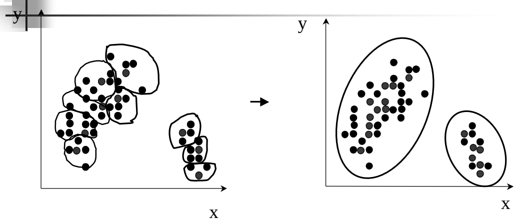
- Draw random sample *s*
- Partition sample to *p* partitions with size *s/p*
- Partially cluster partitions into *s/pq* clusters
- Cluster partial clusters, shrinking representatives towards centroid
- Label data on disk

Data Partitioning and Clustering

- s = 50
- p = 2
- s/p = 25
- s/pq = 5



Cure: Shrinking Representative Points



- Shrink the multiple representative points towards the gravity center by a fraction of α.
- Multiple representatives capture the shape of the cluster

25

Model-Based Clustering

- Basic idea: Clustering as probability estimation
- One model for each cluster
- *Generative* model:
  - Probability of selecting a cluster
  - Probability of generating an object in cluster
- Find max. likelihood or MAP model
- Missing information: Cluster membership
- Use EM algorithm
- Quality of clustering: Likelihood of test objects

26

Mixtures of Gaussians

- Cluster model: Normal distribution (mean, covariance)
- Assume: diagonal covariance, known variance, same for all clusters
- Max. likelihood: mean = avg. of samples
- But what points are samples of a given cluster?
- Estimate prob. that point belongs to cluster
- Mean = weighted avg. of points, weight = prob.
- But to estimate probs. we need model
- "Chicken and egg" problem: use EM algorithm

27

EM Algorithm for Mixtures

- **Initialization:** Choose means at random
- **E step:**
  - For all points and means, compute Prob(point|mean)
  - Prob(mean|point) = Prob(mean) Prob(point|mean) / Prob(point)
- **M step:**
  - Each mean = Weighted avg. of points
  - Weight = Prob(mean|point)
- Repeat until convergence

28

EM Algorithm (contd.)

- Guaranteed to converge to local optimum
- *K-means* is special case

29

AutoClass

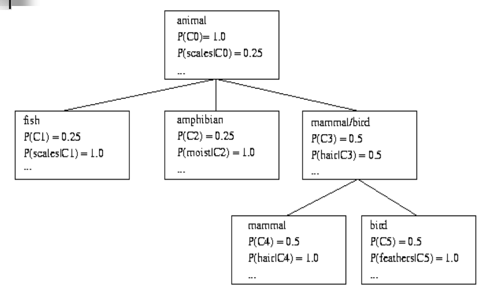
- Developed at NASA (Cheeseman & Stutz, 1988)
- Mixture of Naïve Bayes models
- Variety of possible models for Prob(attribute|class)
- Missing information: Class of each example
- Apply EM algorithm as before
- Special case of learning Bayes net with missing values
- Widely used in practice

30

COBWEB

- Grows tree of clusters (Fisher, 1987)
- Each node contains:
  P(cluster), P(attribute|cluster) for each attribute
- Objects presented sequentially
- Options: Add to node, new node; merge, split
- Quality measure: **Category utility:**
  Increase in predictability of attributes/#Clusters

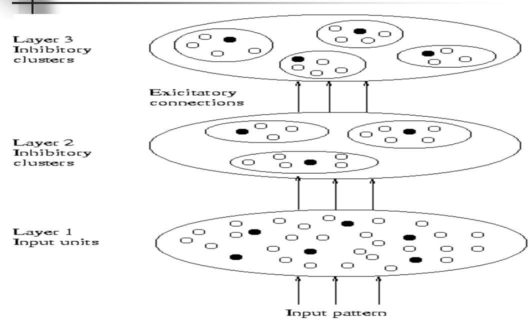
A COBWEB Tree



Neural Network Approaches

- Neuron = Cluster = Centroid in instance space
- Layer = Level of hierarchy
- Several competing sets of clusters in each layer
- Objects sequentially presented to network
- Within each set, neurons compete to win object
- Winning neuron is moved towards object
- Can be viewed as mapping from low-level features to high-level ones

Competitive Learning



Self-Organizing Feature Maps

- Clustering is also performed by having several units competing for the current object
- The unit whose weight vector is closest to the current object wins
- The winner and its neighbors learn by having their weights adjusted
- SOMs are believed to resemble processing that can occur in the brain
- Useful for visualizing high-dimensional data in 2- or 3-D space

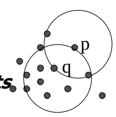
Density-Based Clustering

- 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
- Representative algorithms:
  - DBSCAN (Ester et al., 1996)
  - DENCLUE (Hinneburg & Keim, 1998)

Definitions (I)

- Two parameters:
  - ***Eps****:* Maximum radius of neighborhood
  - ***MinPts****:* Minimum number of points in an Eps-neighborhood of a point
- $N_{Eps}(p) = \{q \in D \mid dist(p,q) <= Eps\}$
- Directly density-reachable**:** A point ***p*** is directly density-reachable from a point ***q*** wrt. ***Eps***, ***MinPts*** iff
  - 1) ***p*** belongs to $N_{Eps}(q)$
  - 2) ***q*** is a core point:

    $|N_{Eps}(q)| >= MinPts$

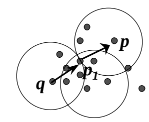


MinPts = 5

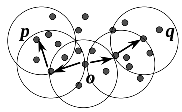
Eps = 1 cm

Definitions (II)

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

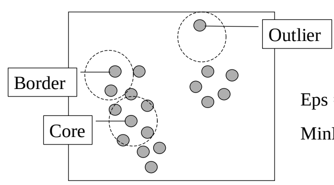


- Density-connected
  - A point *p* is density-connected to a point *q* wrt. *Eps*, *MinPts* if there is a point *o* such that both, *p* and *q* are density-reachable from *o* wrt. *Eps* and *MinPts*.



DBSCAN: Density Based Spatial Clustering of Applications with Noise

- Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points
- Discovers clusters of arbitrary shape in spatial databases with noise



Eps = 1cm

MinPts = 5

DBSCAN: The Algorithm

- Arbitrarily select a point ***p***
- Retrieve all points density-reachable from ***p*** wrt ***Eps*** and ***MinPts***.
- If ***p*** is a core point, a cluster is formed.
- If ***p*** is a border point, no points are density-reachable from ***p*** and DBSCAN visits the next point of the database.
- Continue the process until all of the points have been processed.

DENCLUE: Using Density Functions

- DENsity-based CLUstEring (Hinneburg & Keim, 1998)
- Major features
  - Good for data sets with large amounts of noise
  - Allows a compact mathematical description of arbitrarily shaped clusters in high-dimensional data sets
  - Significantly faster than other algorithms (faster than DBSCAN by a factor of up to 45)
  - But needs a large number of parameters

DENCLUE

- Uses grid cells but only keeps information about grid cells that do actually contain data points and manages these cells in a tree-based access structure.
- Influence function: describes the impact of a data point within its neighborhood.
- Overall density of the data space can be calculated as the sum of the influence function of all data points.
- Clusters can be determined mathematically by identifying density attractors.
- Density attractors are local maxima of the overall density function.

Influence Functions

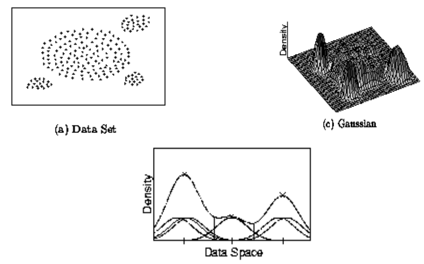
- Example

$$f_{Gaussian}(x,y) = e^{-\frac{d(x,y)^2}{2\sigma^2}}$$

$$f_{Gaussian}^{D}(x) = \sum_{i=1}^{N} e^{-\frac{d(x,x_i)^2}{2\sigma^2}}$$

$$\nabla f_{Gaussian}^{D}(x,x_i) = \sum_{i=1}^{N} (x_i - x) \cdot e^{-\frac{d(x,x_i)^2}{2\sigma^2}}$$

Density Attractors



Center-Defined & Arbitrary Clusters

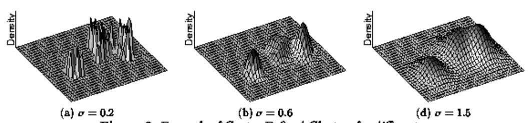


Figure 3: Example of Center-Defined Clusters for different σ

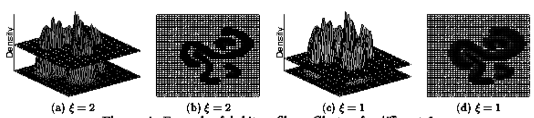


Figure 4: Example of Arbitray-Shape Clusters for different ξ

Clustering: Summary

- Introduction
- Partitioning methods
- Hierarchical methods
- Model-based methods
- Density-based methods