



اصول و روش‌های داده‌کاوی (Data Mining)

درس ششم: روش‌های پایه خوشه‌بندی

مدرس: سمیرا لویمی

گروه مهندسی کامپیوتر - دانشگاه شهید چمران اهواز

Data Mining:

Concepts and Techniques


(3rd ed.)

— Chapter 10 —

Jiawei Han, Micheline Kamber, and Jian Pei
University of Illinois at Urbana-Champaign &
Simon Fraser University

©2011 Han, Kamber & Pei. All rights reserved.

Chapter 10. Cluster Analysis: Basic Concepts and Methods

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

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 research

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 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
 - Discovery of clusters with arbitrary shape
 - Ability to deal with noisy data
 - Incremental clustering and insensitivity to input order
 - High dimensionality


Major Clustering Approaches (I)

- 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

Major Clustering Approaches (II)

- Model-based:
 - A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other
 - Typical methods: EM, SOM, COBWEB
- Frequent pattern-based:
 - Based on the analysis of frequent patterns
 - Typical methods: p-Cluster
- User-guided or constraint-based:
 - Clustering by considering user-specified or application-specific constraints
 - Typical methods: COD (obstacles), constrained clustering
- Link-based clustering:
 - Objects are often linked together in various ways
 - Massive links can be used to cluster objects: SimRank, LinkClus

Chapter 10. Cluster Analysis: Basic Concepts and Methods

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

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)

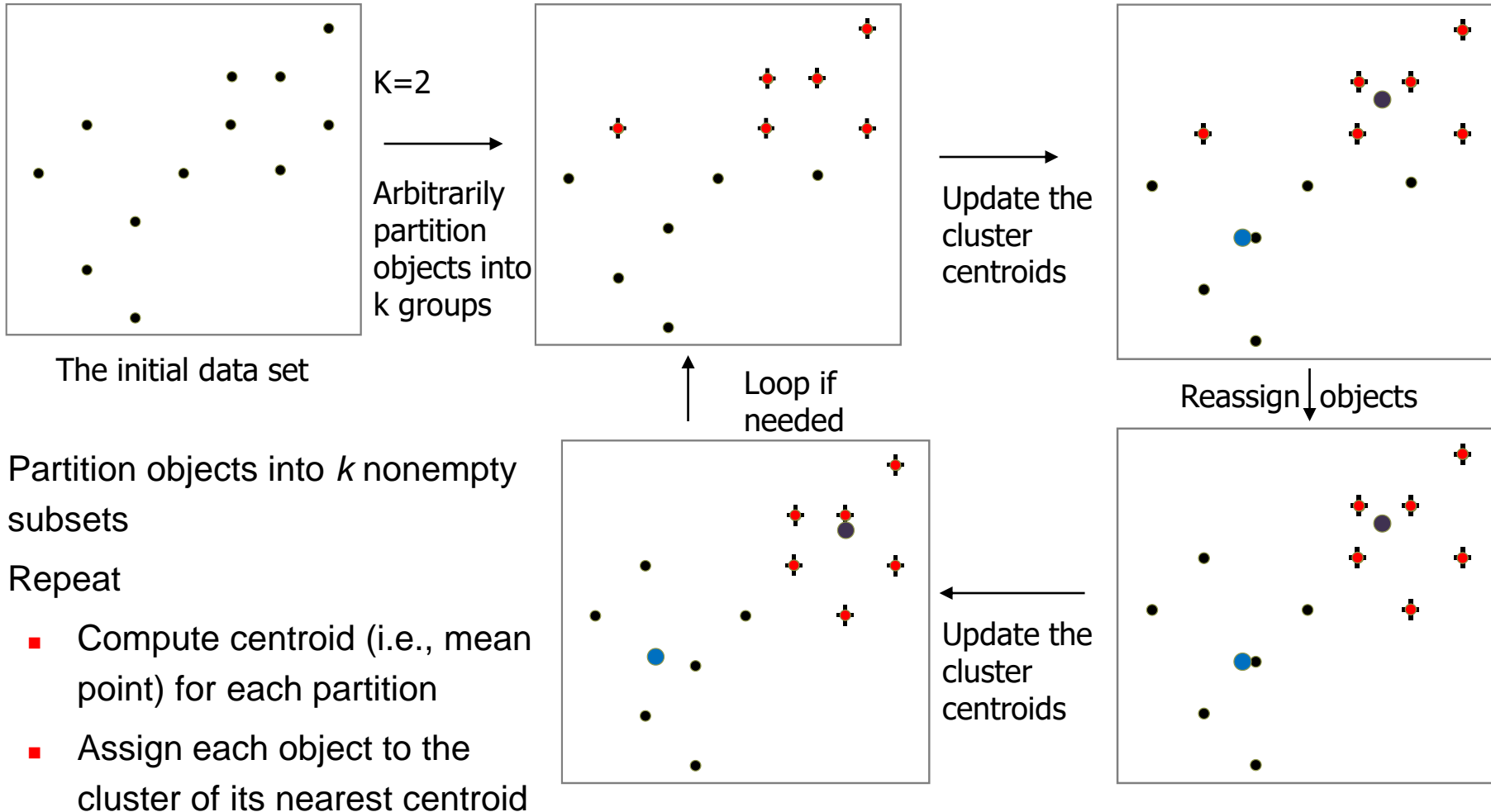
$$E = \sum_{i=1}^k \sum_{p \in C_i} (p - c_i)^2$$

- 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* (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 number of clusters, the K-Means clustering algorithm is outlined as follows
 - Select K points as initial centroids
 - Repeat
 - Form K clusters by assigning each point to its closest centroid
 - Re-compute the centroids (i.e., mean point) of each cluster
 - Until convergence criterion is satisfied
- ❖ Different kinds of measures can be used
 - Manhattan distance (L1 norm), Euclidean distance (L2 norm), Cosine similarity

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

Comments on the *K-Means* Method

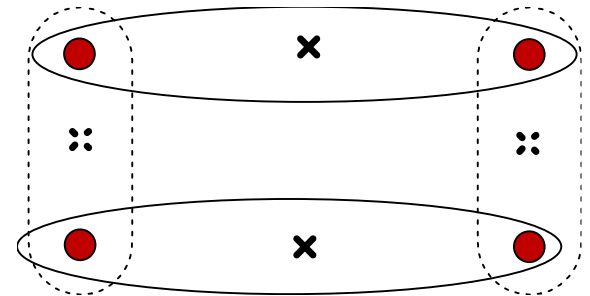
- Strength: *Efficient*. $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, $k, t \ll n$.
 - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$
- Comment: Often terminates at a *local optimal*.
- Weakness
 - Applicable only to objects in a continuous n -dimensional 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 (see Hastie et al., 2009))
 - Sensitive to noisy data and *outliers*
 - Not suitable to discover clusters with *non-convex shapes*

Variations of K-Means

- Choosing better initial centroid estimates
 - K-means++, Intelligent K-Means, Genetic K-Means
- Choosing different representative prototypes for the clusters
 - K-Medoids, K-Medians, K-Modes
- Applying feature transformation techniques
 - Weighted K-Means, Kernel K-Means

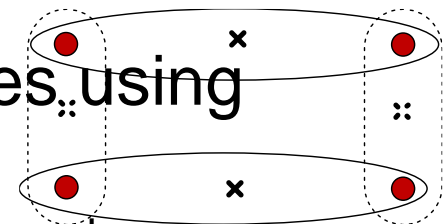
Variations of the *K-Means* Method

- Most of the 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*
 - 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

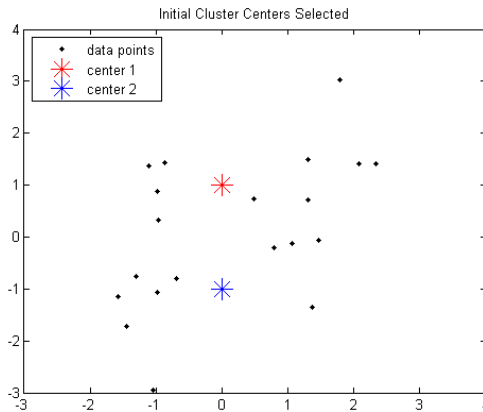


Initialization of K-Means

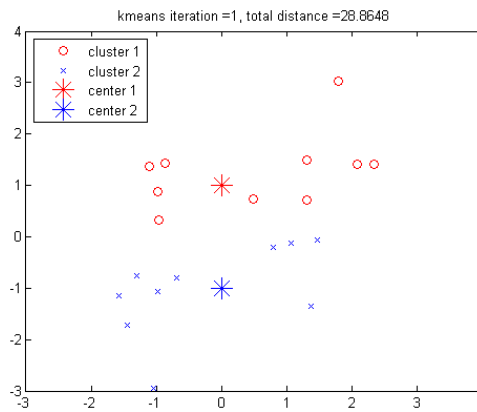
- Different initializations may generate rather different clustering results (some could be far from optimal)
- Original proposal (MacQueen'67): Select K seeds randomly
 - Need to run the algorithm multiple times using different seeds
- There are many methods proposed for better initialization of k seeds
- ***K-Means++*** (Arthur & Vassilvitskii'07):
 - The first centroid is selected at random
 - The next centroid selected is the one that is farthest from the currently selected (selection is based on a weighted probability score)
 - The selection continues until K centroids are obtained



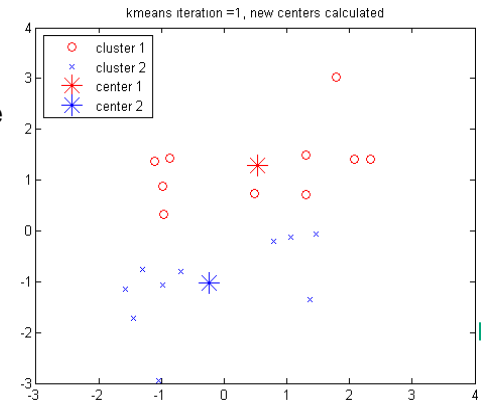
Example: Poor Initialization May Lead to Poor Clustering



Assign points to clusters

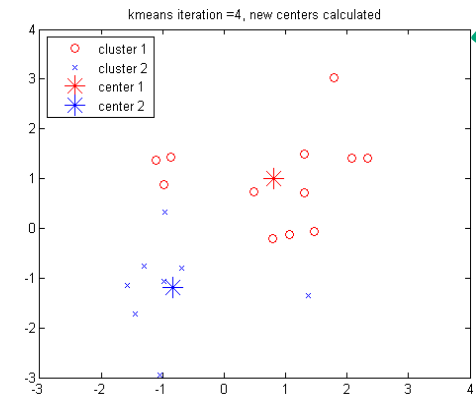


Recompute cluster centers



Another random selection of k centroids for the same data points

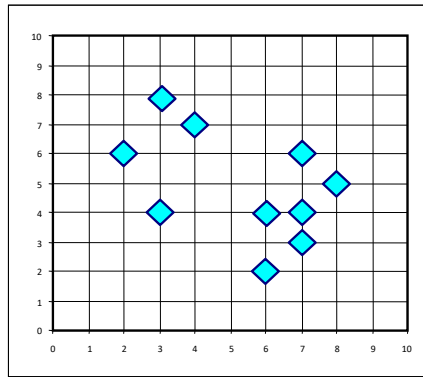
- ❑ Rerun of the *K-Means* using another random K seeds
- ❑ This run of *K-Means* generates a poor quality clustering



Handling Outliers: From K-Means to K-Medoids

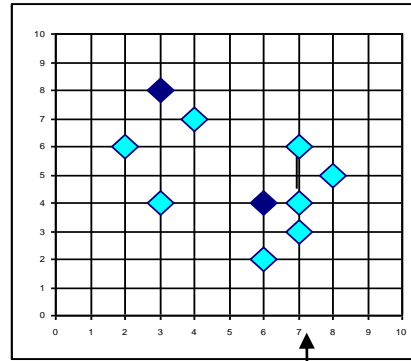
- The K-Means algorithm is sensitive to outliers
 - An object with an extremely large value may substantially distort the distribution of the data
- K-Medoids: Instead of taking the mean value of the object in a cluster as a reference point, medoids can be used, which is the most centrally located object in a cluster
- The K-Medoids clustering algorithm:
 - Select K points as the initial representative objects (i.e., as initial K medoids)
 - Repeat
 - Assigning each point to the cluster with the closest medoid
 - Randomly select a non-representative object o_i
 - Compute the total cost S of swapping the medoid m with o_i
 - If $S < 0$, then swap m with o_i to form the new set of medoids
 - Until convergence criterion is satisfied

PAM: A Typical K-Medoids Algorithm

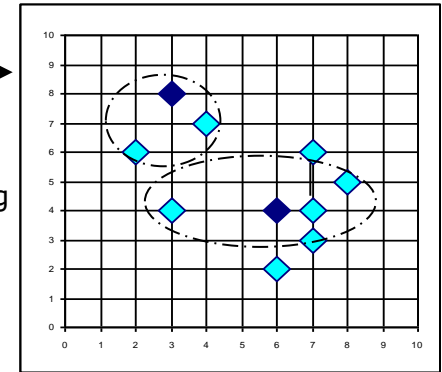


$K = 2$

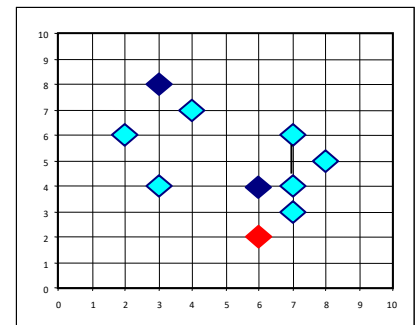
Arbitrary
choose K
object as
initial
medoids



Assign
each
remaining
object to
nearest
medoids

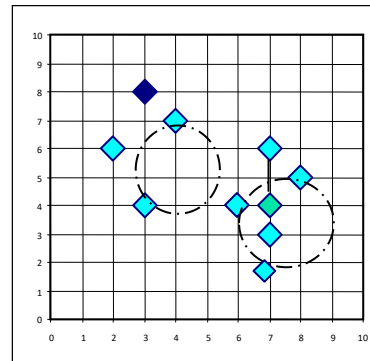


Randomly select a non-
medoid object, O_{random}



Compute
total cost of
swapping

Swapping O
and O_{random}
If quality is
improved



Select initial K medoids randomly

Repeat

Object re-assignment

Swap medoid m with o_i if it
improves the clustering quality

Until convergence criterion is satisfied

Discussion on K-Medoids Clustering

- K-Medoids Clustering: Find representative objects (medoids) in clusters
- PAM (Partitioning Around Medoids: Kaufmann & Rousseeuw 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 sum of the squared errors (SSE) 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)
 - Computational complexity: PAM: $O(K(n - K)^2)$ (quite expensive!)
- Efficiency improvements on PAM
 - CLARA (Kaufmann & Rousseeuw, 1990):
 - PAM on samples; $O(Ks^2 + K(n - K))$, s is the sample size
 - CLARANS (Ng & Han, 1994): Randomized re-sampling, ensuring efficiency + quality

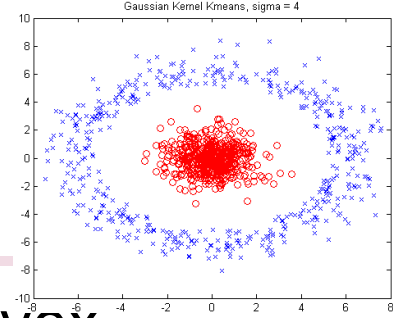
K-Medians: Handling Outliers by Computing Medians

- Medians are less sensitive to outliers than means
 - Think of the median salary vs. mean salary of a large firm when adding a few top executives!
- K-Medians: Instead of taking the mean value of the object in a cluster as a reference point, medians are used (L1-norm as the distance measure)
- The criterion function for the K-Medians algorithm
$$S = \sum_{k=1}^K \sum_{x_i \in C_k} |x_{ij} - med_{kj}|$$
- The K-Medians clustering algorithm:
 - Select K points as the initial representative objects (i.e., as initial K medians)
 - Repeat
 - Assign every point to its nearest median
 - Re-compute the median using the median of each individual feature
 - Until convergence criterion is satisfied

K-Modes: Clustering Categorical Data

- K-Means cannot handle non-numerical (categorical) data
 - Mapping categorical value to 1/0 cannot generate quality clusters for high-dimensional data
- K-Modes: An extension to K-Means by replacing means of clusters with modes
- Dissimilarity measure between object X and the center of a cluster Z
 - $\Phi(x_j, z_j) = 1$ when $x_j \neq z_j$; and 0 otherwise
- This dissimilarity measure (distance function) is frequency-based
- Algorithm is still based on iterative object cluster assignment and centroid update
- A fuzzy K-Modes method is proposed to calculate a fuzzy cluster membership value for each object to each cluster
- A mixture of categorical and numerical data: Using a K-Prototype method

Kernel K-Means Clustering



- Kernel K-Means can be used to detect non-convex clusters
 - K-Means can only detect clusters that are linearly separable
- Idea: Project data onto the high-dimensional kernel space, and then perform K-Means clustering
 - Map data points in the input space onto a high-dimensional feature space using the kernel function
 - Perform K-Means on the mapped feature space
- Computational complexity is higher than K-Means
 - Need to compute and store $n \times n$ kernel matrix generated from the kernel function on the original data
- The widely studied spectral clustering can be considered as a variant of Kernel K-Means clustering

Kernel Functions and Kernel K-Means Clustering

- Typical kernel functions:
 - Polynomial kernel of degree h : $K(\mathbf{X}_i, \mathbf{X}_j) = (\mathbf{X}_i \mathbf{X}_j + 1)^h$
 - Gaussian radial basis function (RBF) kernel: $K(\mathbf{X}_i, \mathbf{X}_j) = e^{-\frac{\|\mathbf{X}_i - \mathbf{X}_j\|^2}{2\sigma^2}}$
 - Sigmoid kernel: $K(\mathbf{X}_i, \mathbf{X}_j) = \tanh(\kappa \mathbf{X}_i \cdot \mathbf{X}_j - \delta)$
- The formula for kernel matrix K for any two points $\mathbf{X}_i, \mathbf{X}_j \in C_k$ is $K(\mathbf{X}_i, \mathbf{X}_j) = \phi(\mathbf{X}_i) \cdot \phi(\mathbf{X}_j)$
- The SSE criterion of kernel K-means: $SSE(C) = \sum_{k=1}^K \sum_{x_i \in C_k} |\phi(x_i) - c_k|^2$
 - The formula for the cluster centroid: $c_k = \frac{\sum_{x_i \in C_k} \phi(x_i)}{|C_k|}$
- Clustering can be performed without the actual individual projections $\phi(x_i)$ and $\phi(x_j)$ for the data points $x_i, x_j \in C_k$

Example: Kernel Functions and Kernel K-Means Clustering

- Gaussian radial basis function (RBF) kernel:

$$K(\mathbf{X}_i, \mathbf{X}_j) = e^{-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{2\sigma^2}}$$

- Suppose there are 5 original 2-dimensional points:

$$x_1(0, 0), x_2(4, 4), x_3(-4, 4), x_4(-4, -4), x_5(4, -4)$$

Example: Kernel Functions and Kernel K-Means Clustering

- If we set σ to 4, we will have the following points in the kernel space
 - E.g., $\|x_1 - x_2\|^2 = (0 - 4)^2 + (0 - 4)^2 = 32$,
therefore, $K(x_1, x_2) = e^{-\frac{32}{2 \cdot 4^2}} = e^{-1}$

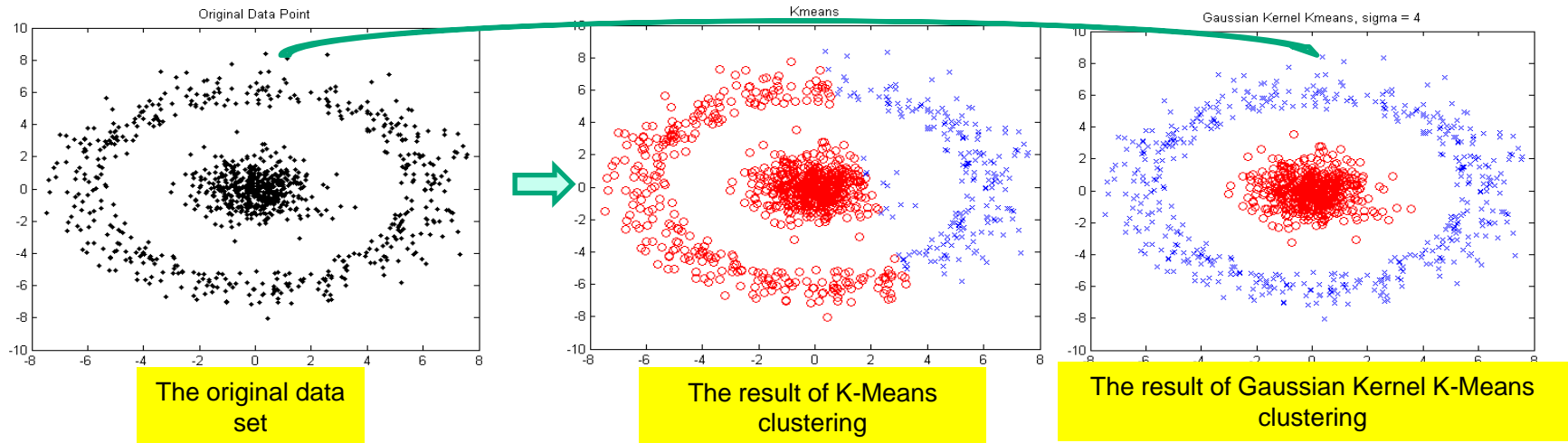
Original
Space

	x	y
x_1	0	0
x_2	4	4
x_3	-4	4
x_4	-4	-4
x_5	4	-4

RBF Kernel Space ($\sigma =$
4)


$K(x_i, x_1)$	$K(x_i, x_2)$	$K(x_i, x_3)$	$K(x_i, x_4)$	$K(x_i, x_5)$
1	$e^{-\frac{4^2+4^2}{2 \cdot 4^2}} = e^{-1}$	e^{-1}	e^{-1}	e^{-1}
e^{-1}	1	e^{-2}	e^{-4}	e^{-2}
e^{-1}	e^{-2}	1	e^{-2}	e^{-4}
e^{-1}	e^{-4}	e^{-2}	1	e^{-2}
e^{-1}	e^{-2}	e^{-4}	e^{-2}	1

Example: Kernel K-Means Clustering



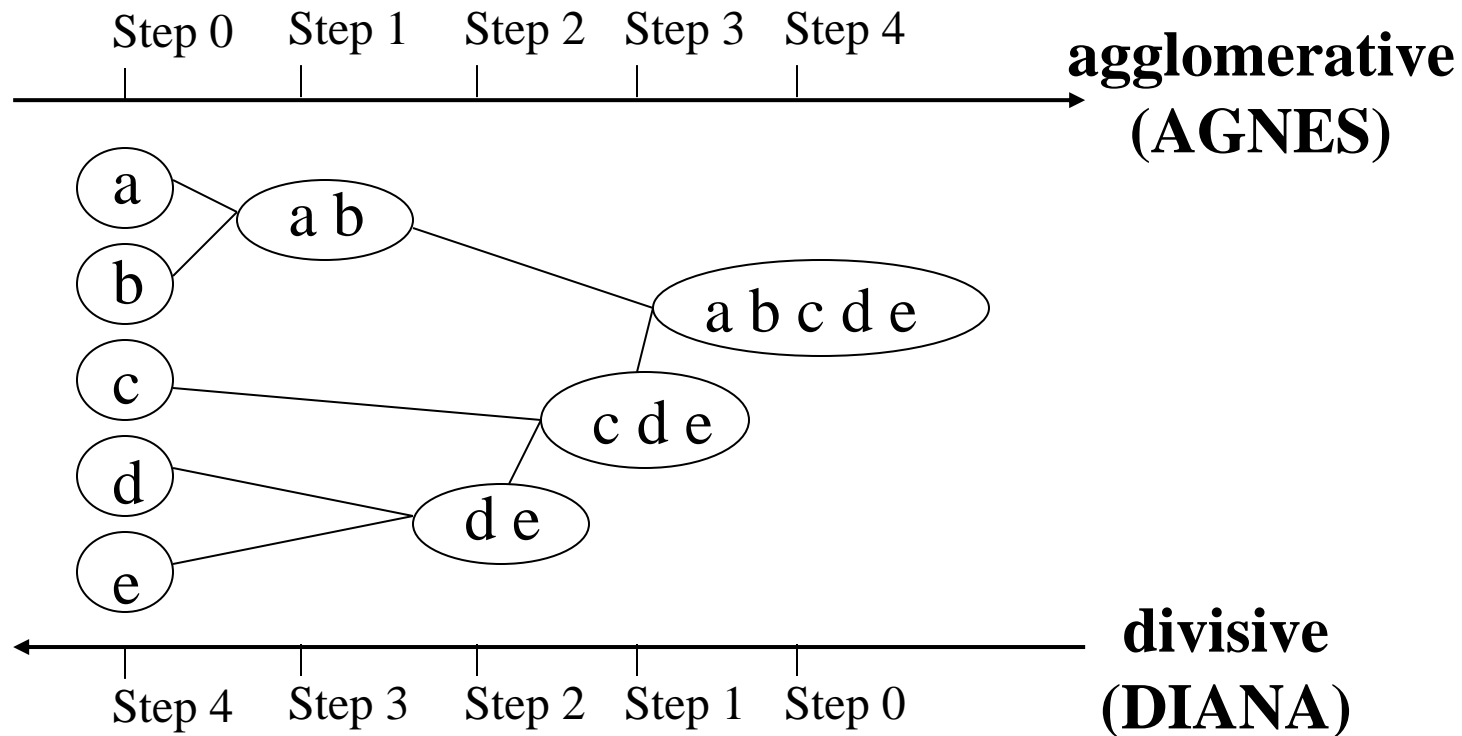
- ❑ The above data set cannot generate quality clusters by K-Means since it contains non-convex clusters
- ❑ Gaussian RBF Kernel transformation maps data to a kernel matrix K for any two points x_i, x_j : $K_{x_i x_j} = \phi(x_i) \bullet \phi(x_j)$ and Gaussian kernel: $K(X_i, X_j) = e^{-\|X_i - X_j\|^2 / 2\sigma^2}$
- ❑ K-Means clustering is conducted on the mapped data, generating quality clusters

Chapter 10. Cluster Analysis: Basic Concepts and Methods

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

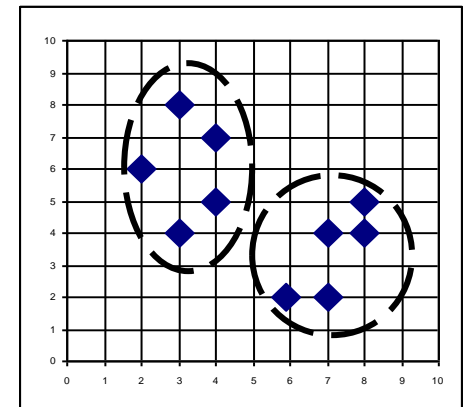
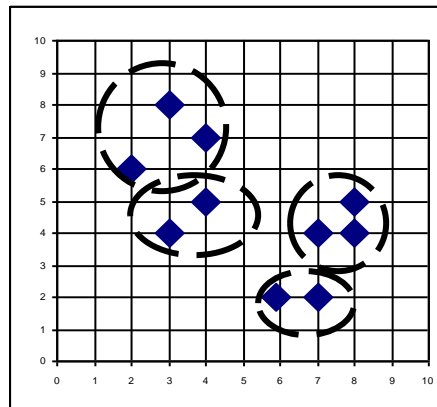
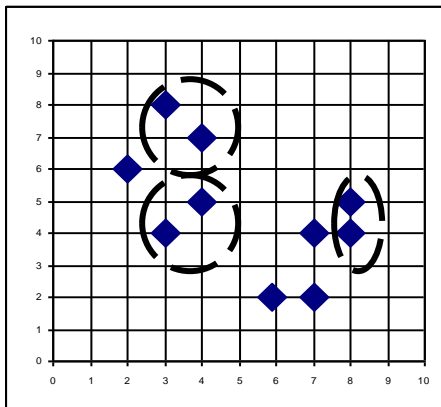
Hierarchical Clustering

- 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)

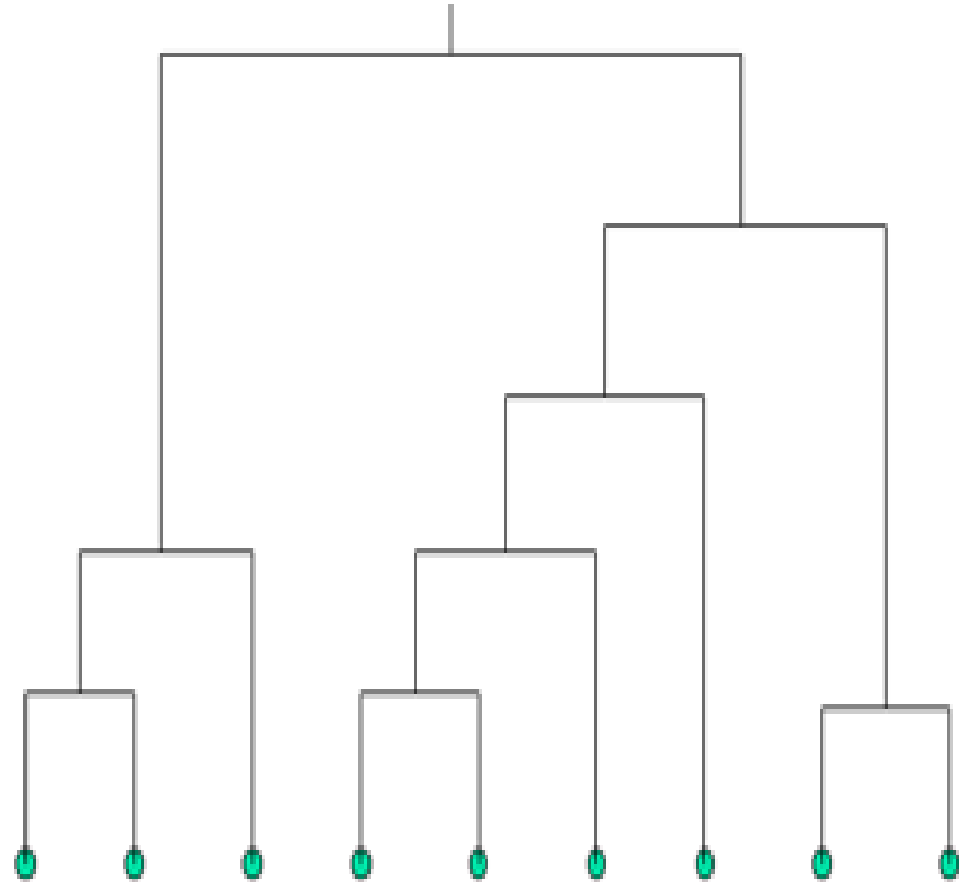
- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical packages, e.g., Splus
- Use the **single-link** method and the dissimilarity matrix
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



Dendrogram: Shows How Clusters are Merged

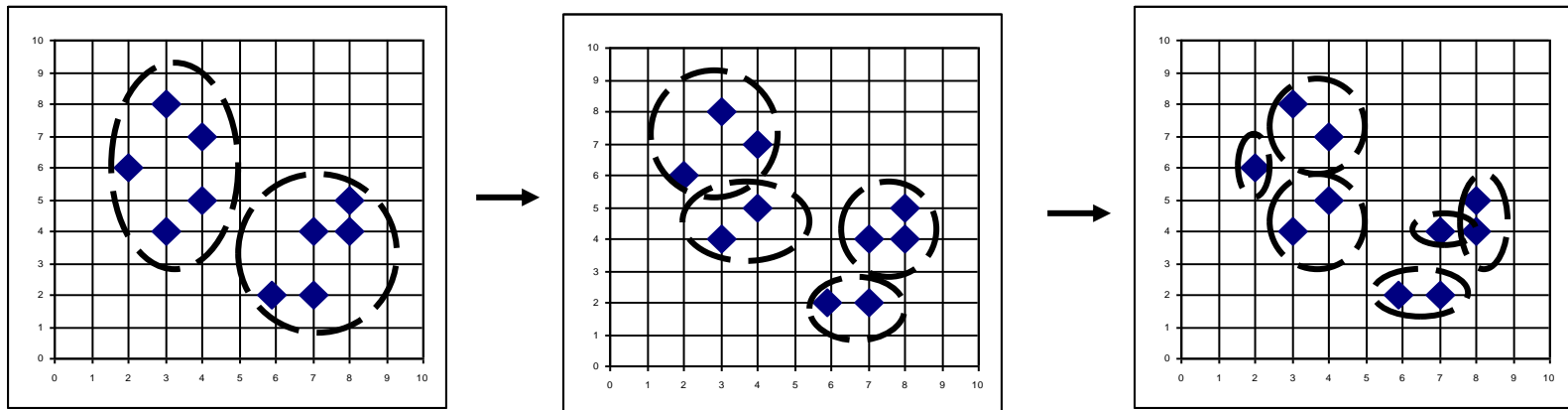
Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram

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



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

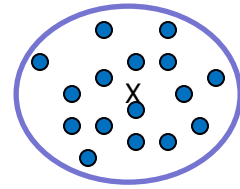
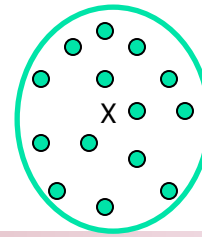
Minimum distance: $dist_{min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} \{|p - p'|\}$

Maximum distance: $dist_{max}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} \{|p - p'|\}$

Mean distance: $dist_{mean}(C_i, C_j) = |m_i - m_j|$

Average distance: $dist_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p \in C_i, p' \in C_j} |p - p'|$

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

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

- Radius: square root of average distance from any point of the cluster to its centroid

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

- Diameter: square root of average mean squared distance between all pairs of points in the cluster

$$D_m = \sqrt{\frac{\sum_{i=1}^N \sum_{q=1}^N (t_{ip} - t_{iq})^2}{N(N-1)}}$$

BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies)

- Zhang, Ramakrishnan & Livny, SIGMOD'96
- Incrementally construct a CF (Clustering Feature) tree, a hierarchical data structure for multiphase clustering
 - Phase 1: scan DB to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data)
 - 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
- *Weakness*: handles only numeric data, and sensitive to the order of the data record

Clustering Feature Vector in BIRCH

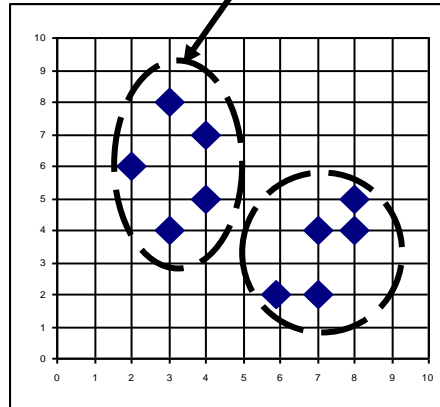
Clustering Feature (CF): $CF = (N, LS, SS)$

N : Number of data points

LS : linear sum of N points: $\sum_{i=1}^N X_i$

SS : square sum of N points

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



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

(3,4)

(2,6)

(4,5)

(4,7)

(3,8)

Generative Model

- Given a set of 1-D points $X = \{x_1, \dots, x_n\}$ for clustering analysis & assuming they are generated by a Gaussian distribution:

$$\mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- The probability that a point $x_i \in X$ is generated by the model

$$P(x_i|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

- The likelihood that X is generated by the model:

$$L(\mathcal{N}(\mu, \sigma^2) : X) = P(X|\mu, \sigma^2) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

- The task of learning the generative model: find the parameters μ and σ^2 such that

$$\mathcal{N}(\mu_0, \sigma_0^2) = \arg \max \{L(\mathcal{N}(\mu, \sigma^2) : X)\}$$

the maximum likelihood

A Probabilistic Hierarchical Clustering Algorithm

- For a set of objects partitioned into m clusters C_1, \dots, C_m , the quality can be measured by,

$$Q(\{C_1, \dots, C_m\}) = \prod_{i=1}^m P(C_i)$$

where $P()$ is the maximum likelihood

- Distance between clusters C_1 and C_2 : $dist(C_i, C_j) = -\log \frac{P(C_1 \cup C_2)}{P(C_1)P(C_2)}$
- Algorithm: Progressively merge points and clusters

Input: $D = \{o_1, \dots, o_n\}$: a data set containing n objects

Output: A hierarchy of clusters

Method

Create a cluster for each object $C_i = \{o_i\}$, $1 \leq i \leq n$;


For $i = 1$ to n {

Find pair of clusters C_i and C_j such that

$C_i, C_j = \operatorname{argmax}_{i \neq j} \{\log (P(C_i \cup C_j) / (P(C_i)P(C_j)))\}$;

If $\log (P(C_i \cup C_j) / (P(C_i)P(C_j))) > 0$ then merge C_i and C_j }

Chapter 10. Cluster Analysis: Basic Concepts and Methods

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

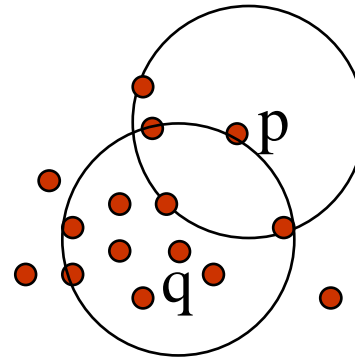
Density-Based Clustering Methods

- 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) (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
- $N_{Eps}(p)$: $\{q \text{ belongs to } D \mid \text{dist}(p,q) \leq Eps\}$
- **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:

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



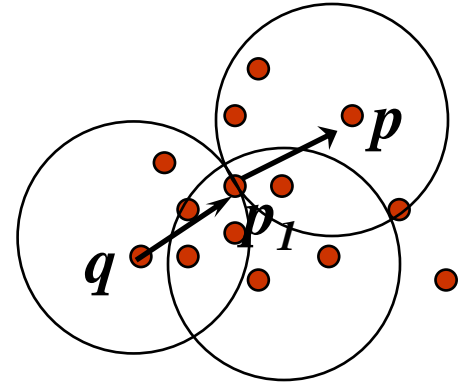
MinPts = 5

Eps = 1 cm

Density-Reachable and Density-Connected

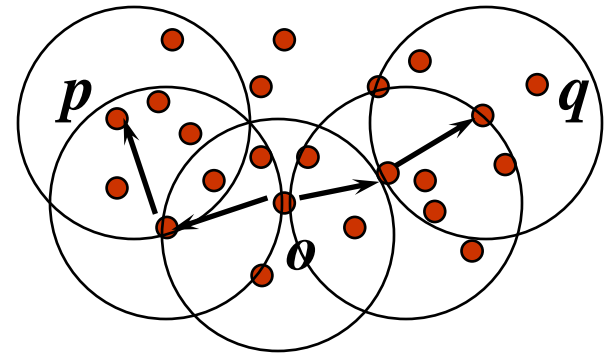
- Density-reachable:

- 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



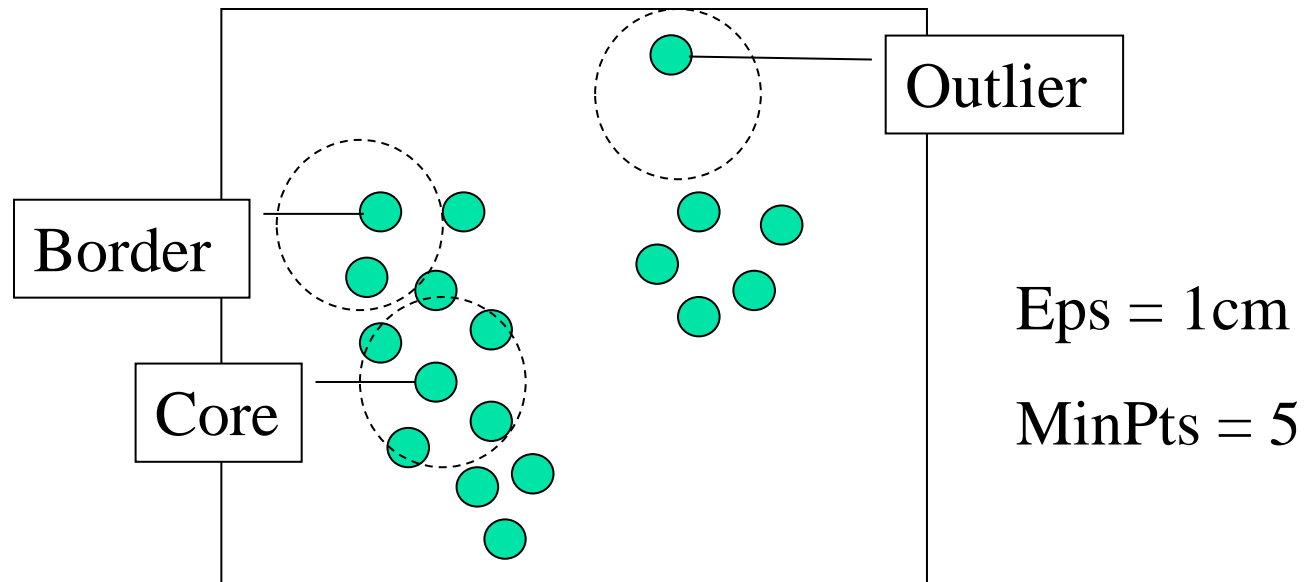
- Density-connected

- 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


- 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



DBSCAN: The Algorithm

- Arbitrary select a point p
- Retrieve all points density-reachable from p w.r.t. 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

Chapter 10. Cluster Analysis: Basic Concepts and Methods

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

Assessing Clustering Tendency

- Assess if non-random structure exists in the data by measuring the probability that the data is generated by a uniform data distribution
- Test spatial randomness by statistic test: Hopkins Static
 - Given a dataset D regarded as a sample of a random variable o , determine how far away o is from being uniformly distributed in the data space
 - Sample n points, p_1, \dots, p_n , uniformly from D . For each p_i , find its nearest neighbor in D : $x_i = \min\{\text{dist}(p_i, v)\}$ where v in D
 - Sample n points, q_1, \dots, q_n , uniformly from D . For each q_i , find its nearest neighbor in $D - \{q_i\}$: $y_i = \min\{\text{dist}(q_i, v)\}$ where v in D and $v \neq q_i$
 - Calculate the Hopkins Statistic:
$$H = \frac{\sum_{i=1}^n y_i}{\sum_{i=1}^n x_i + \sum_{i=1}^n y_i}$$
 - If D is uniformly distributed, $\sum x_i$ and $\sum y_i$ will be close to each other and H is close to 0.5. If D is highly skewed, H is close to 0

Determine the Number of Clusters

- Empirical method
 - # of clusters $\approx \sqrt{n/2}$ for a dataset of n points
- Elbow method
 - Use the turning point in the curve of sum of within cluster variance w.r.t the # of clusters
- 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 fits the data the best

Measuring Clustering Quality

- Two methods: extrinsic vs. intrinsic
- Extrinsic: supervised, i.e., the ground truth is available
 - Compare a clustering against the ground truth using certain clustering quality measure
 - Ex. BCubed precision and recall metrics
- Intrinsic: unsupervised, i.e., the ground truth is unavailable
 - Evaluate the goodness of a clustering by considering how well the clusters are separated, and how compact the clusters are
 - Ex. Silhouette coefficient

Measuring Clustering Quality: Extrinsic Methods

- Clustering quality measure: $Q(C, C_g)$, for a clustering C given the ground truth C_g .
- 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

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
- **K-means** and **K-medoids** algorithms are popular partitioning-based clustering algorithms
- **Birch** and **Chameleon** are interesting hierarchical clustering algorithms, and there are also probabilistic hierarchical clustering algorithms
- **DBSCAN**, **OPTICS**, and **DENCLU** are interesting density-based algorithms
- **STING** and **CLIQUE** are grid-based methods, where CLIQUE is also a subspace clustering algorithm
- Quality of clustering results can be evaluated in various ways