Clustering

Young-Rae Cho, Ph.D.
Associate Professor
Division of Software / Division of Digital Healthcare
Yonsei University – Mirae Campus

What Is Clustering?

- **Cluster**
  - A group of data objects
  - Similar (or related) to one another within the same group
  - Dissimilar (or unrelated) to the objects in different groups

- **Clustering (or Cluster Analysis)**
  - Finding clusters from data objects
  - Unsupervised learning: no pre-defined classes

- **Applications**
  - A stand-alone method for data analysis
  - A preprocessing step for other data analysis

Applications of Clustering

- **Business**
  - Grouping customers to promote sales
- **Economy**
  - Finding stocks with similar patterns for investment
- **IT**
  - Grouping web documents for information retrieval
- **Biology**
  - Grouping genes to predict their biological functions
- **Geography**
  - Finding areas with similar land use for city planning
- **Weather**
  - Finding similar climate patterns for weather forecast

Measuring Quality of Clustering

- **High-Quality Clusters have**
  - High intra-class similarity: cohesive within clusters
  - Low inter-class similarity: distinctive between clusters

- **Quality of Clustering Depends on**
  - Clustering methods
    - Handling both cohesiveness and distinctiveness
    - Ability to discover hidden patterns
    - Defining “similar enough” – problem of determining a threshold
  - Data sets
    - Amount of data
    - Complexity of data type
    - High dimensionality

Similarity / Dissimilarity Functions (1)

Numerical Attributes

- Minkowski distance,

$$d = \left(\sum_{i=1}^{n} | x_i - y_i |^p \right)^{1/p}$$

- Euclidean distance, when p=2
- Manhattan distance, when p=1

Binary Attributes

contingency table

	1	0	sum
1	q	r	q+r
0	s	t	s+t
sum	q+s	r+t	p

- If a binary variable is symmetric,
  - Dissimilarity $d = \dfrac{r+s}{q+r+s+t}$
- If a binary variable is asymmetric,
  - Dissimilarity $d = \dfrac{r+s}{q+r+s}$, similarity (Jaccard index) $s = \dfrac{q}{q+r+s}$

Similarity / Dissimilarity Functions (2)

Categorical Attributes

- Similarity (Jaccard index), $s(x,y)=\dfrac{|X\cap Y|}{|X\cup Y|}$

  where X: the set of variables for the object x

  Y: the set of variables for the object y

- Similarity (Geometric index), $s(x,y)=\dfrac{|X\cap Y|^2}{(|X|\cdot|Y|)}$

- Similarity (Dice index), $s(x,y)=\dfrac{2|X\cap Y|}{|X|+|Y|}$

Mixed Attributes

- Weighted combination

Issues in Clustering

- Ability to deal with **different types of data**
- **Scalability** (handling a very large amount of data)
- **High dimensionality**
- Insensitivity to **order of input records**
- Ability to deal with **noise and outliers**
- Ability to handle **dynamically changing data**
- Incorporation of **user-specified constraints**
- Interpretability and **usability**
- Discovery of clusters with **arbitrary shapes**
- Requirements of domain knowledge to determine **input parameters**

Categories of Clustering Methods

- **Partition-based Methods**
  - Finding the best partitions, e.g., k-means, k-medoids, CLARANS
- **Hierarchical Methods**
  - Hierarchically merging or dividing data, e.g., Agnes, Diana, BIRCH
- **Density-based Methods**
  - Finding densely populated groups of data, e.g., DBSCAN, OPTICS
- **Grid-based Methods**
  - Grouping data based on multi-level granularity, e.g., STING, CLIQUE
- **Model-based Methods**
  - Finding the best fit to models, e.g., EM, COBWEB, SOM
- **Pattern-based Methods**
  - Grouping data with similar patterns, e.g., p-Cluster
- **Constraint-based Methods**
  - Considering user-specified or application-specific constraints

Overview

1. Partition-Based Methods
2. Hierarchical Methods
3. Density-Based Methods
4. Grid-Based Methods
5. Pattern-Based Methods
6. Cluster Validation

Partition-based Methods

- **Main Idea**
  - Constructing a partition of the data with n objects into k clusters

- **Issues**
  - Finding a partition that optimize the clustering quality

    → high intra-class similarity and low inter-class similarity

- **Methods**
  - Brute force algorithms (or, Exhaustive search algorithms) ?
  - Heuristic algorithms?

    ex., k-means, k-medoids (PAM), CLARANS

k-Means

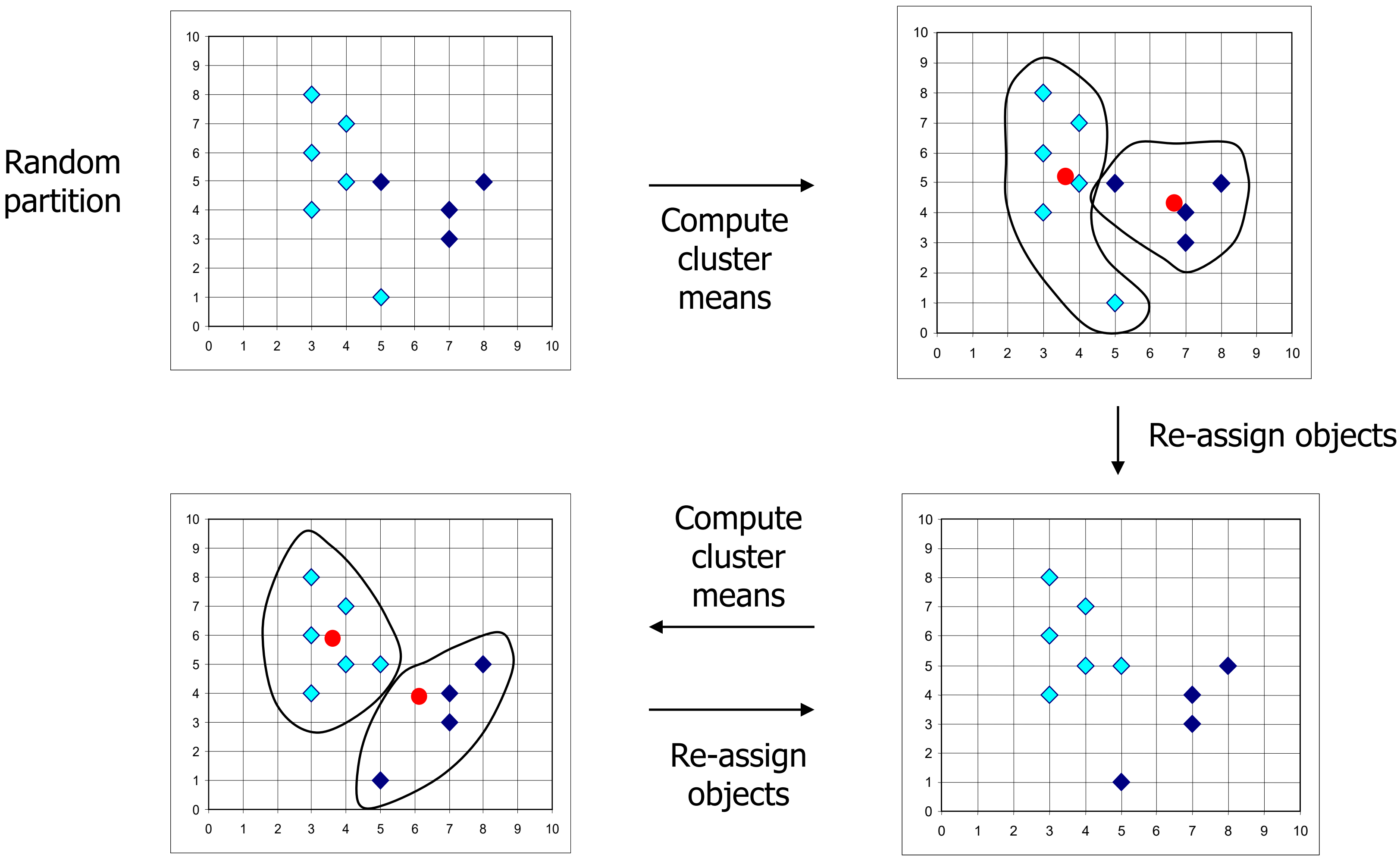
Main Idea

- Heuristic clustering algorithm
- Converges a local optimum quickly

Process

1) Partition objects randomly into k clusters.
2) Compute the mean point of the objects in each cluster as a centroid
3) Assign each object to the nearest centroid and generate k new clusters
4) Repeat (2) and (3) until there is no change of the objects in each cluster

Example of k-Means



Summary of k-Means

- **Strength**
  - Relatively efficient
    - O(?) where n objects, k clusters and t iterations
    - Normally, t,k « n

- **Weakness**
  - Need to specify k, the number of clusters, in advance
  - Sensitive to noise and outliers
  - Applicable to only numeric data (when the mean is defined), not categorical data
  - Not suitable to detect clusters with non-convex shapes
  - Fall into local optimum, not identifying global optimum of clusters

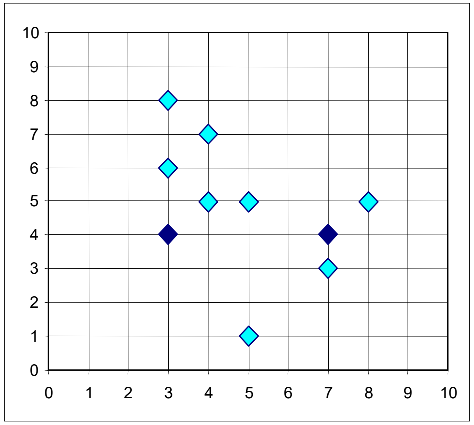
k-Medoids

- **Main Idea**
  - The same process to k-means
  - Instead of taking the mean of objects as a centroid for each cluster,
    use a medoid, the most centrally located object in a cluster

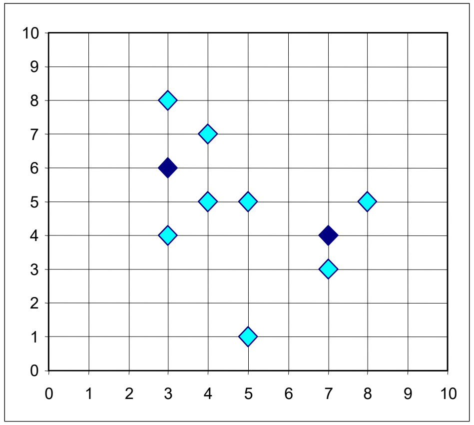
- **Process**
  1) Select k medoids randomly
  2) Compute the total cost as the sum of distance between each non-medoid and its nearest medoid
  3) Replace one medoid with one non-medoid if the swap decreases the total cost
  4) Repeat (3) until there is no change of the objects in each cluster

Example of k-Medoids

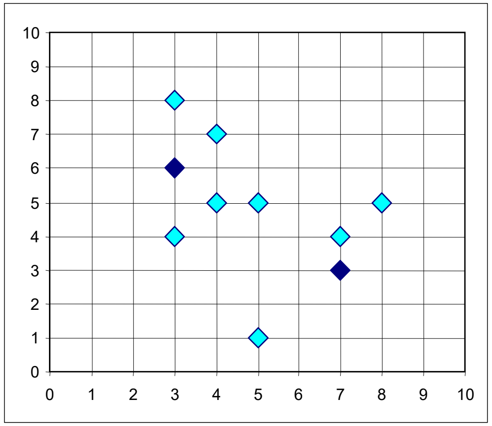
Random selection of k medoids & compute cost



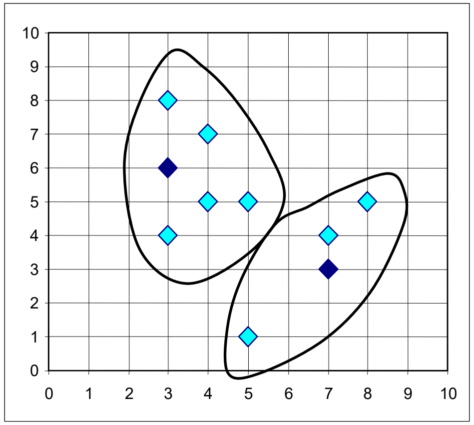
Swapping & compute cost



Swapping & compute cost

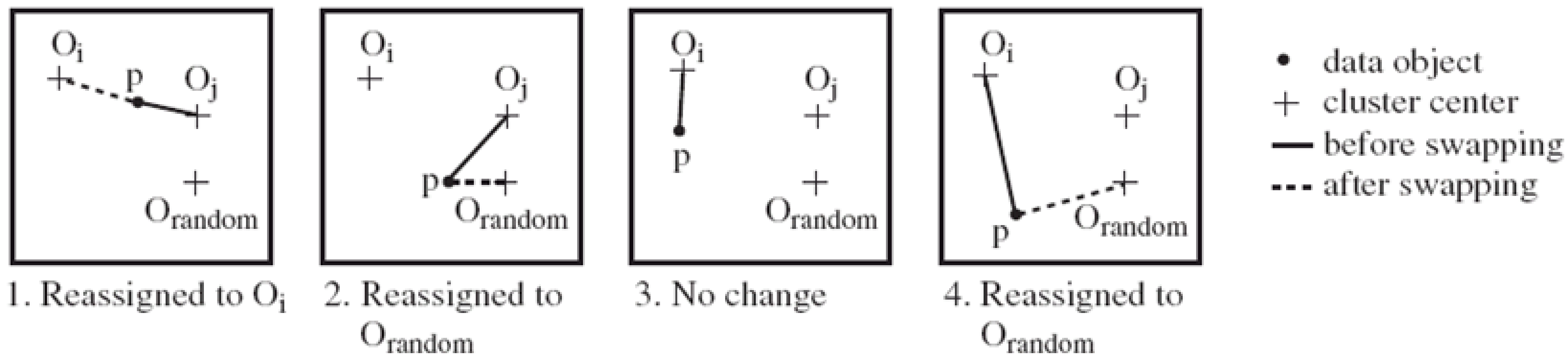


Stop swapping if the lowest cost



Iterative Clustering in k-Medoids

- **Clustering Step**
  - All non-medoids are assigned to the closest medoid to form a set of clusters for each iteration
  - Swapping changes cluster membership
    - Oi, Oj are original medoids
    - Oj is swaped with Orandom



Summary of k-Medoids

- **Strength**
  - Robust to noise and outliers comparing to k-Means

- **Weakness**
  - Not scalable to a large data set
  - O(?) where n objects and k clusters

    → How to improve efficiency and scalability?

Improved Fast k-Medoids

Main Idea

- k-Means: fast, but sensitive to outliers
- k-Medoids: less sensitive to outliers, but slow
- Combining k-means and k-medoids

Process

1) Choose k objects of the smallest sum of distance to the others as medoids
2) Assign each object to the nearest medoid and generate k initial clusters
3) Choose k object of the smallest sum of distance to the others within the cluster as medoids
4) Assign each object to the nearest medoid and generate k new clusters
5) Repeat (3) and (4) until there is no change of the objects in each cluster

Reference

- Park, H.-S. and Jun, C.-H., “A simple and fast algorithm for k-medoids clustering.” Expert Systems with Applications (2009)

CLARA (Clustering Large Applications)

- **Process**
  1) Draw multiple samples of the data set
  2) Apply PAM on each sample
  3) Measure the quality (total cost) of clusters in the entire data set
  4) Output the best clustering results

- **Strength**
  - Solve inefficiency of PAM in a large data set

- **Weakness**
  - Efficiency depends on the sample size.
  - The output does not represent the result from the whole data set if the sample is biased.

CLARANS (Clustering Algorithm with Randomized Search)

❑ Main Idea

- Select a non-medoid to replace with a medoid in a sample (like CLARA)
- Not restrict the medoid search in a particular sample (unlike CLARA)
- Dynamically change the sample in every step of medoid search
- Not confine the search in a localized area

❑ Strength

- More efficient than PAM and more accurate than CLARA

❑ Reference

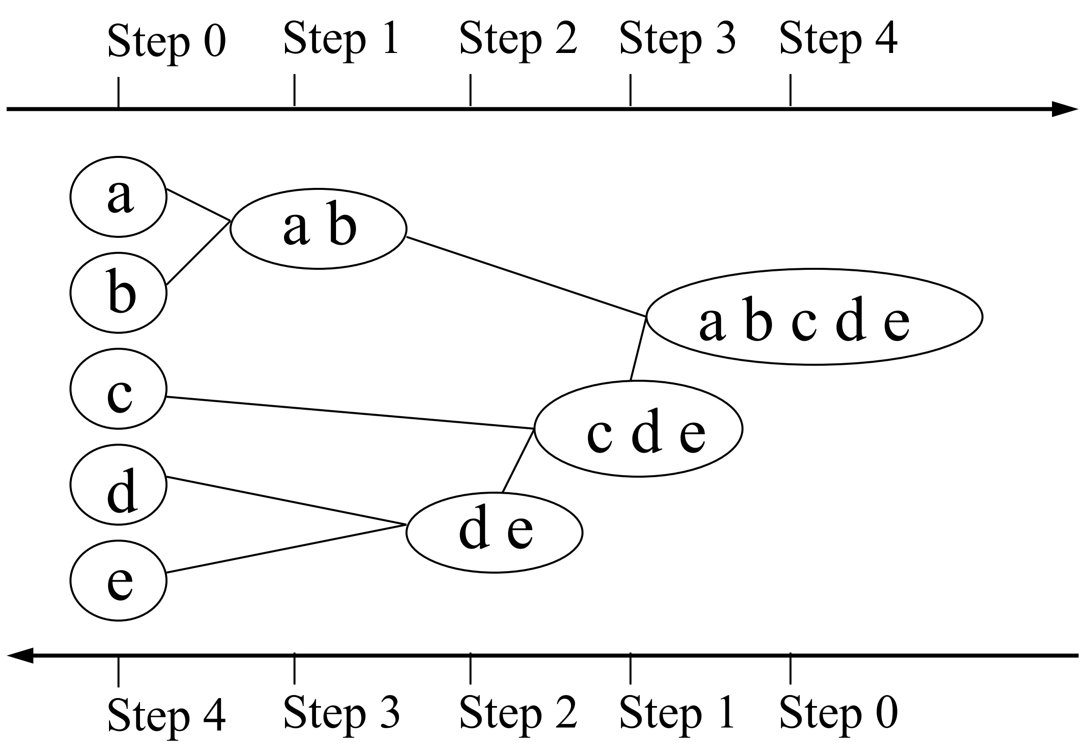
- Ng, R. and Han, J., “CLARANS: A Method for Clustering Objects for Spatial Data Mining.” IEEE Transactions on Knowledge and Data Engineering (2002)

Overview

1. **Partition-Based Methods**
2. **Hierarchical Methods**
3. **Density-Based Methods**
4. **Grid-Based Methods**
5. **Pattern-Based Methods**
6. **Cluster Validation**

Hierarchical Methods

- **Main Idea**
  - Decomposing data objects into several levels of nested partitioning (tree of clusters)



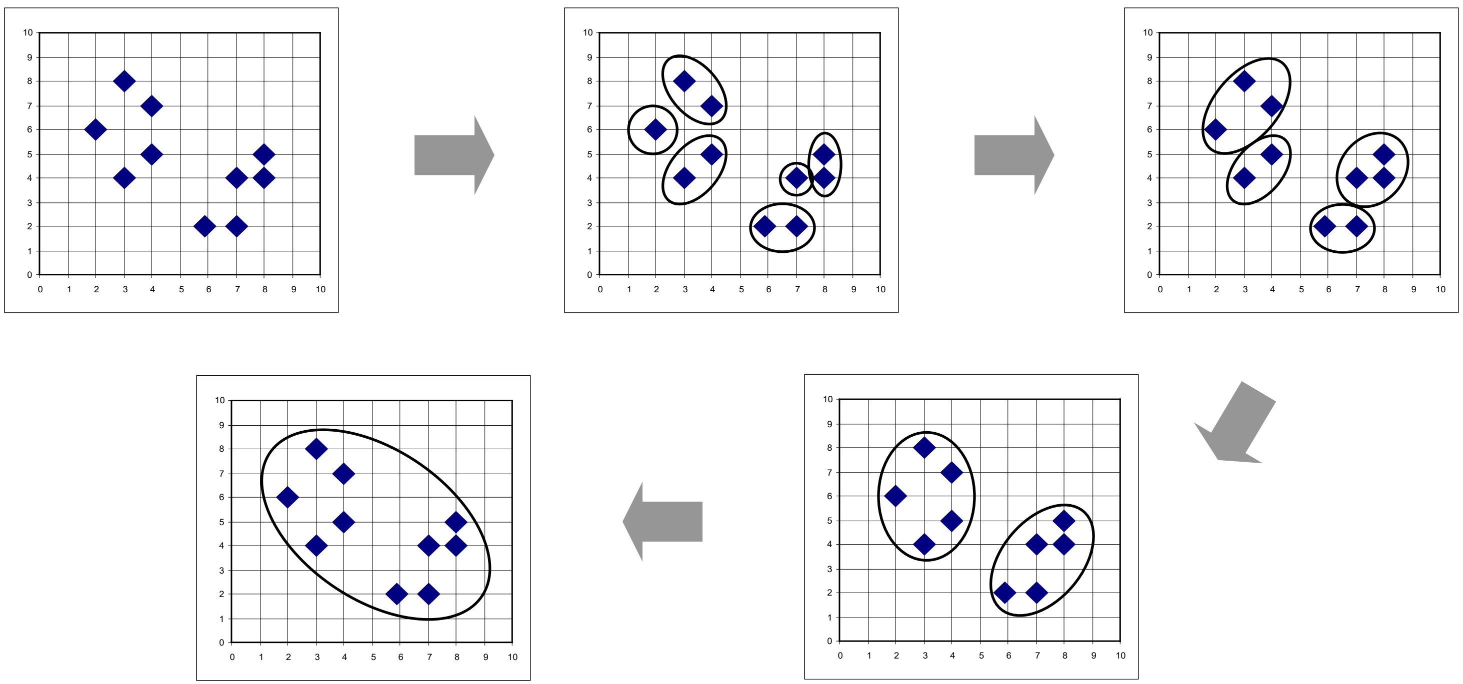
agglomerative (AGNES):
bottom-up approach

divisive (DIANA):
top-down approach

AGNES (Agglomerative Nesting)

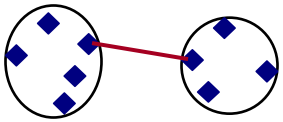
❑ Process

1) Start with all single-node clusters
2) Iteratively merge the closest (the most similar) clusters
3) Eventually, all nodes belong to one cluster.

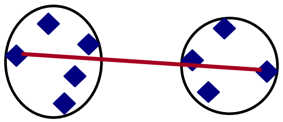


Distance Measures between Clusters

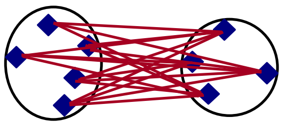
- **Single-Link Distance:** $$d(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y)$$



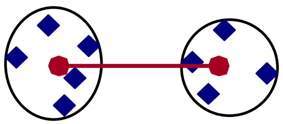
- **Complete-Link Distance:** $$d(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y)$$



- **Average-Link Distance:** $$d(C_i, C_j) = \frac{1}{n_i n_j} \sum_{x \in C_i} \sum_{y \in C_j} d(x, y)$$



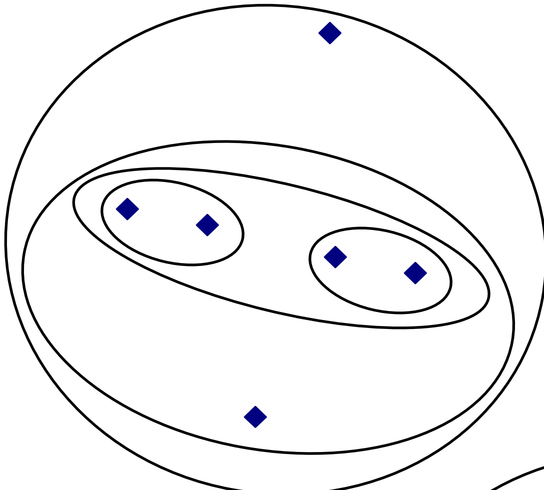
- **Centroid Distance:** $$d(C_i, C_j) = d(m_i, m_j)$$



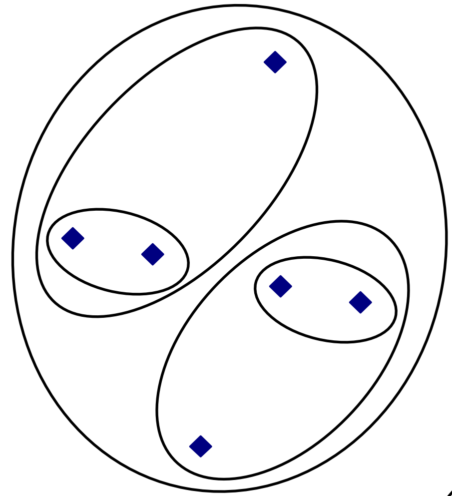
where m_i and m_j are means of C_i and C_j

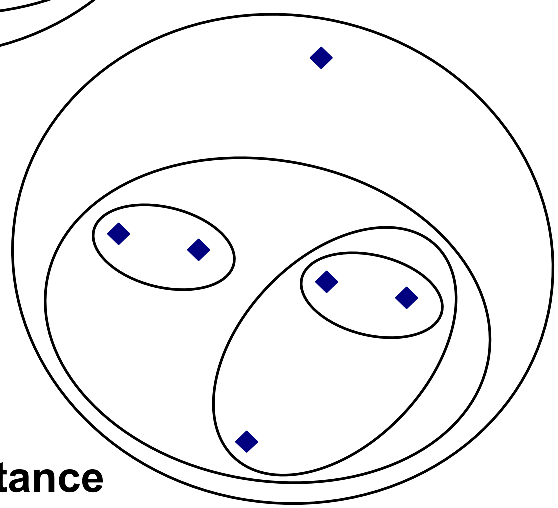
Comparison of Distance Measures

Single-Link Distance

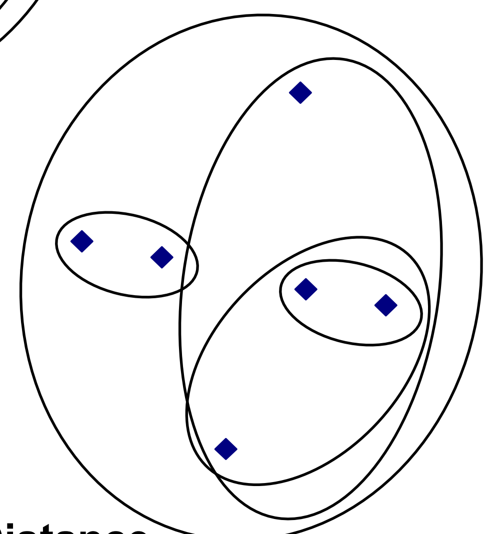


Complete-Link Distance





Average-Link Distance



Centroid Distance

Comparison between Single-Link and Complete-Link

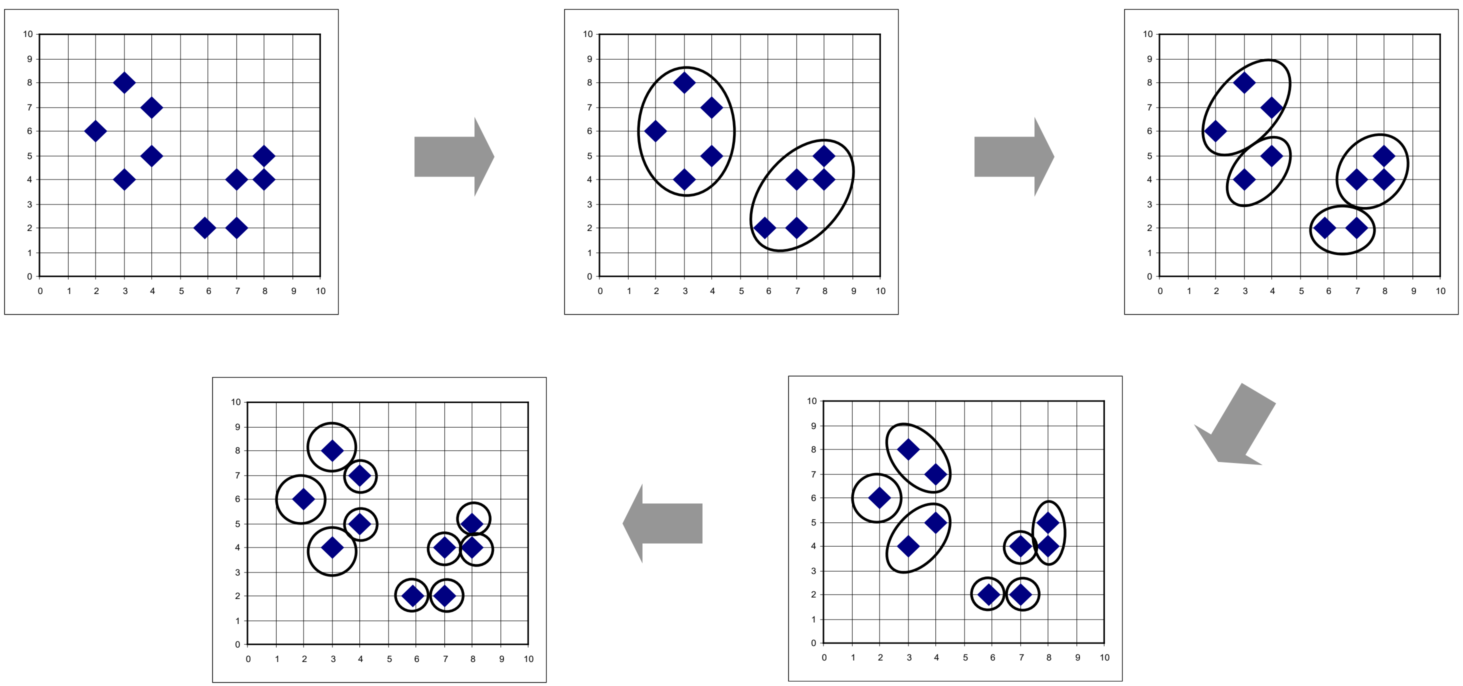
- **Single-Link Distance**
  - Strength
    - Handles non-sphere shape clusters
  - Weakness
    - Sensitive to noise

- **Complete-Link Distance**
  - Strength
    - Less sensitive to noise
  - Weakness
    - Biased large-sized clusters (or uneven-sized clusters)

DIANA (Divisive Analysis)

❑ Process

1) Start with one single clusters with all nodes
2) Iteratively divide the farthest (the most dissimilar) clusters
3) Eventually, all clusters have a single node.



Summary of Hierarchical Methods

- **Strength**
  - Not require the number of clusters, k, in advance
  - Reveal a hierarchical structure of clusters

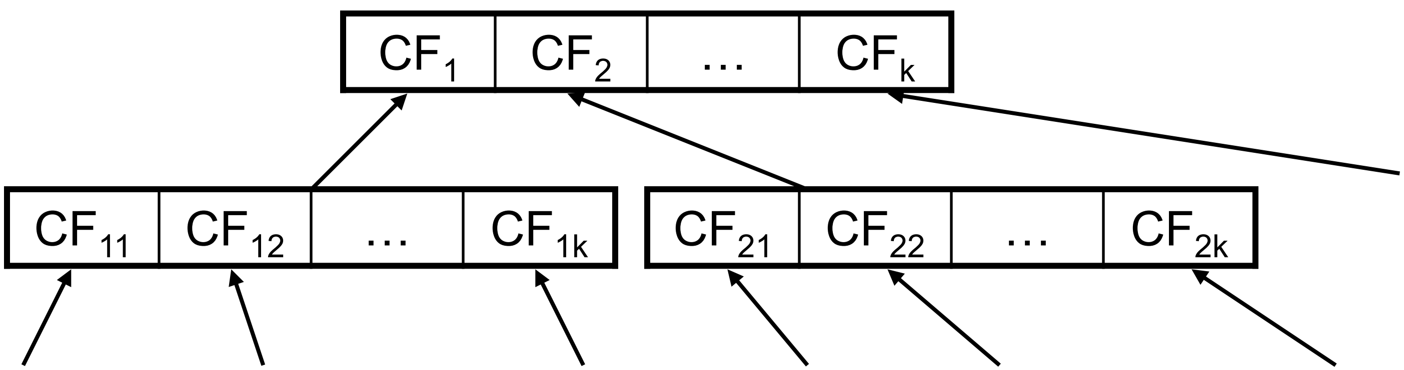
- **Weakness**
  - Require the stopping condition
  - Sensitive to noise
  - Not able to undo what was done previously
  - Not scalable, at least $O(n^2)$ where n objects

    → How to improve efficiency and scalability?

BIRCH (Balanced Iterative Reducing & Clustering in Hierarchies)

- **Main Idea**
  - Building CF tree, a hierarchical data structure for multi-phase clustering

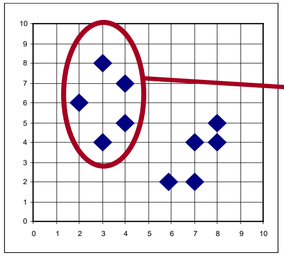
- **Process**
  1) Scan DB to build an initial in-memory CF tree
  2) Iteratively build the higher-level of the CF tree by grouping nodes



CF (Clustering Features)

- **Clustering Features (CF)**
  - Three-dimensional vector summarizing information of data objects for a cluster
  - **CF** = < n, **LS**, **SS** >
    - n is the number of data points
    - **LS** is the linear sum of n points $\sum_{i=1}^{n} x_i$
    - **SS** is the square sum of n points $\sum_{i=1}^{n} x_i^2$

- **Example**



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

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

Implementation with Clustering Features

Similarity Computation between Clusters

- Average-link distance between two clusters, C_i and C_j ,

$$d(C_i, C_j) = \sqrt{\frac{\sum_{x \in C_i} \sum_{y \in C_j} (x-y)^2}{n_i n_j}}$$

can be calculated using the components in their clustering feature vectors.

Merging Clusters

- CF_1 = < n_1, LS_1, SS_1 >
- CF_2 = < n_2, LS_2, SS_2 >
- CF_1 + CF_2 = < n_1+n_2, LS_1+LS_2, SS_1+SS_2 >

Summary of BIRCH

- **Strength**
  - Linearly scalable, ~ O(n) where n is the number of objects
  - Efficient memory usage

- **Weakness**
  - Only find spherical clusters by the distance measure
  - Only applicable to numeric attributes

- **Reference**
  - Zhang, T., Ramakrishnan, R. and Livny, M., “BIRCH: An Efficient Data Clustering Method for Very Large Databases”, In Proceeding of SIGMOD (1996)

ROCK (Robust Clustering using Links)

❑ Main Idea

- Clustering categorical data
- Previous approaches
  - Similarity: Jaccard index (ratio of common attributes)
  - But, not reflect distribution patterns of attributes in the datasets
- This approach
  - Using link (a new definition) instead of similarity
  - Reflect distribution patterns of attributes in the datasets

❑ Definitions

- **Neighbors**: two objects, a and b, are neighbors if $sim(a, b) > \theta$
- **Link**: the number of common neighbors between two objects

Examples of Link

- Measuring similarity between **{a,b,c}** and **{c,d,e}**

- **Example 1**
  - DB: {a,b,c}, {a,b,d}, {a,b,f}, {a,c,d}, {a,d,e}, {b,c,e}, {b,f,g}, {c,d,e}
  - $\theta = 0.5$
  - Neighbors of {a,b,c}: {a,b,d}, {a,b,f}, {a,c,d}, {b,c,e}
  - Neighbors of {c,d,e}: {a,c,d}, {a,d,e}, {b,c,e}
  - Link({a,b,c}, {c,d,e}) = 2 (Jaccard({a,b,c}, {c,d,e}) = 0.2)

- **Example 2**
  - DB: {a,b,c}, {c,d,e}, {a,e,f}, {b,d,g}, {c,e,g}
  - $\theta = 0.5$
  - Neighbors of {a,b,c}: None
  - Neighbors of {c,d,e}: {c,e,g}
  - Link({a,b,c}, {c,d,e}) = 0 (Jaccard({a,b,c}, {c,d,e}) = 0.2)

Summary of ROCK

- **Process**
  1) Compute similarity matrix using link
  2) Run hierarchical (bottom-up) clustering

- **Strength**
  - Results depend on the other data objects
  - Guarantee high intra-class similarity within a cluster

- **Weakness**
  - Not guarantee low inter-class similarity between clusters

- **Reference**

  - Guha, S., Rastogi, R. and Shim, K., “ROCK: An Robust Clustering Algorithm for Categorical Attributes”, In Proceeding of ICDE (1999)

Overview

1. Partition-Based Methods
2. Hierarchical Methods
3. <u>Density-Based Methods</u>
4. Grid-Based Methods
5. Pattern-Based Methods
6. Cluster Validation

Density-based Methods

- **Main Idea**
  - Clustering data objects located densely
  - Use density as the local clustering criterion
- **Issues**
  - Find clusters of arbitrary shapes
  - Handle outliers
  - Determine density parameters as termination condition
- **Methods**
  - DBSCAN
  - OPTICS

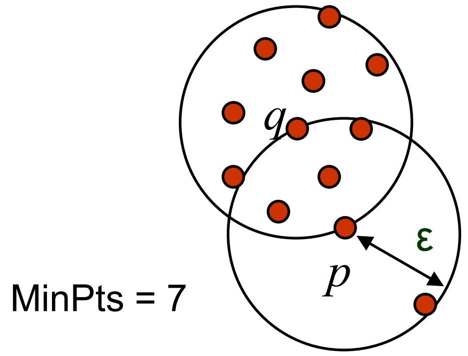
DBSCAN

❑ DBSCAN

- Density-based Spatial Clustering of Applications with Noise
- Typical density-based clustering method

❑ Basic Terms

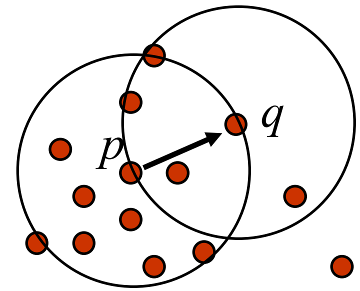
- **ε-neighborhood**
  - ε: minimum radius of neighborhood
  - ε-neighborhood of a data point p, $N_\varepsilon(p)=\{q \in D \mid dist(p,q) \leq \varepsilon, p \neq q\}$
- **Core / Border**
  - MinPts: minimum number of data points of an ε-neighborhood
  - Core: if $|N_\varepsilon(p)| \geq MinPts$, p is a core
  - Border: if $|N_\varepsilon(p)| < MinPts$, p is a border



Density-Reachability

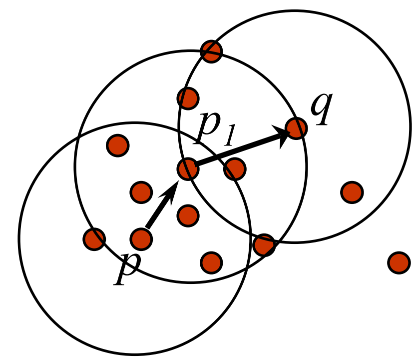
- **Direct Density-Reachable**
  - An object q is directly density-reachable from the object p, if p is a core and q is in ε-neighborhood of p.
  - Symmetric or Asymmetric (?)

MinPts = 6

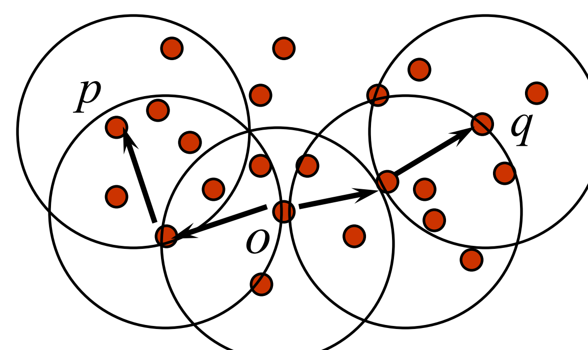


- **Density-Reachable**
  - An object q is (indirectly) density-reachable from the object p, if there is a chain of points, p_1, p_2, … p_n, such that p_1=p, p_n=q, and $p_{(i+1)}$ is direct density-reachable from p_i.
  - Symmetric or Asymmetric (?)

MinPts = 6



- **Density-Connected**
  - An object p is density-connected to an object q, if there is an object o, and both p and q are density-reachable from o.
  - Symmetric or Asymmetric (?)



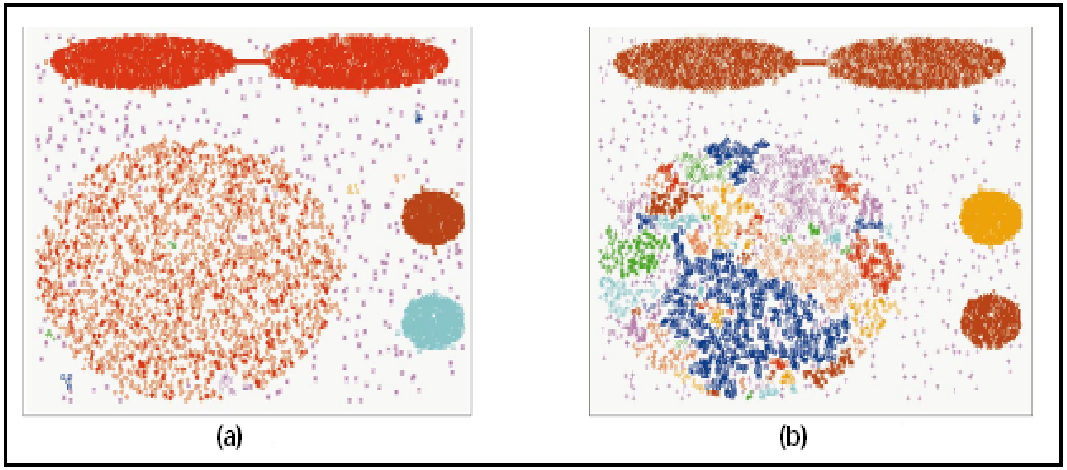
Clustering by DBSCAN

- **Cluster**
  - A maximal set of density-connected points

- **Algorithm**
  1) Select an arbitrary data point p
  2) If p is a core, retrieve all data points density-reachable from p as a cluster
  3) Repeat (1) and (2) until there is no more data point to be selected

- **Results**

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



Summary of DBSCAN

- **Strength**
  - Robust to noise
  - Find arbitrary shapes and sizes

- **Weakness**
  - Cannot handle varying densities
  - Sensitive to parameters, ε and MinPts

    → Need a density-based algorithm without pre-setting parameters, e.g., OPTICS

- **Reference**

  - Ester, M., et al., “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”, In Proceedings of KDD (1996)

OPTICS

- **OPTICS**
  - Ordering Points to Identify Clustering Structures
  - Density-based and hierarchical clustering method

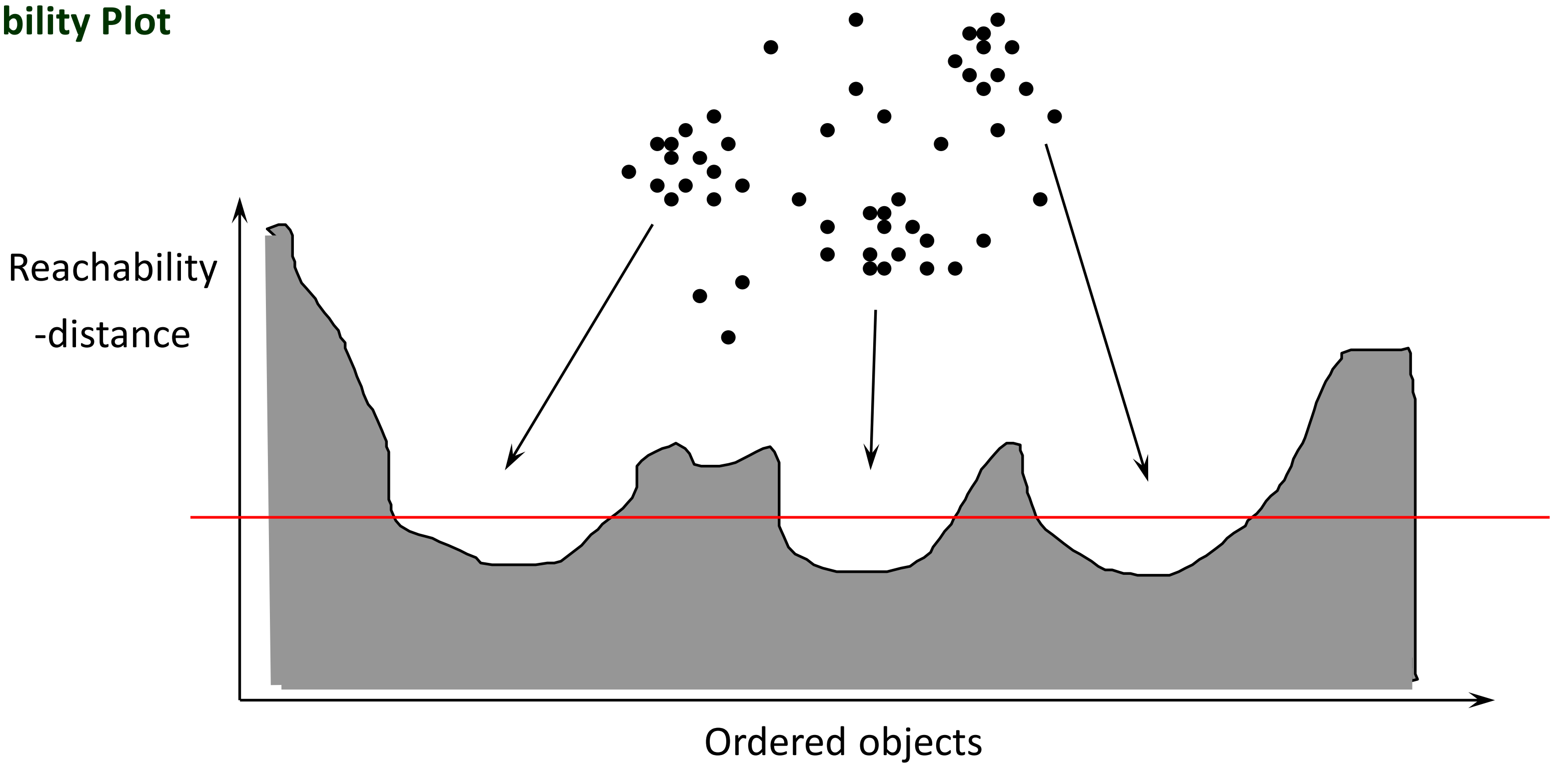
- **Main Idea**
  - Clustering data points with varying density
  - Produce a specific order of data points
  - Provide a hierarchical structure of density-based clusters

- **Basic Terms**
  - **Core-Distance**
    - Core-distance of p: distance between p and MinPts'th closest point if $|N_\varepsilon(p)| \geq MinPts$
  - **Reachability-Distance**
    - Reachability-distance of q from p: max (core-distance(p), distance(p,q))

Graphic View of OPTICS Results

- **Algorithm**
  1) Select an arbitrary data point p
  2) If p is a core, compute reachability-distance to all data points and select the closest point
  3) Repeat (2) for ordering data points until there is no more data point to be selected

- **Reachability Plot**



Summary of OPTICS

- **Strength**
  - Able to visualize graphically
  - Find density-based hierarchical clusters
  - Allow interactive clustering analysis

- **Weakness**
  - May not cover all data points

- **Reference**
  - Ankerst, M., et al., "OPTICS: Ordering Points to Identify the Clustering Structure", In Proceedings to SIGMOD (1999)

Overview

1. Partition-Based Methods
2. Hierarchical Methods
3. Density-Based Methods
4. <u>Grid-Based Methods</u>
5. Pattern-Based Methods
6. Cluster Validation

Grid-based Methods

Main Idea

- Use multi-resolution grid data structure
- Fast processing time, independent of the number of data objects

Methods

- STING
- CLIQUE

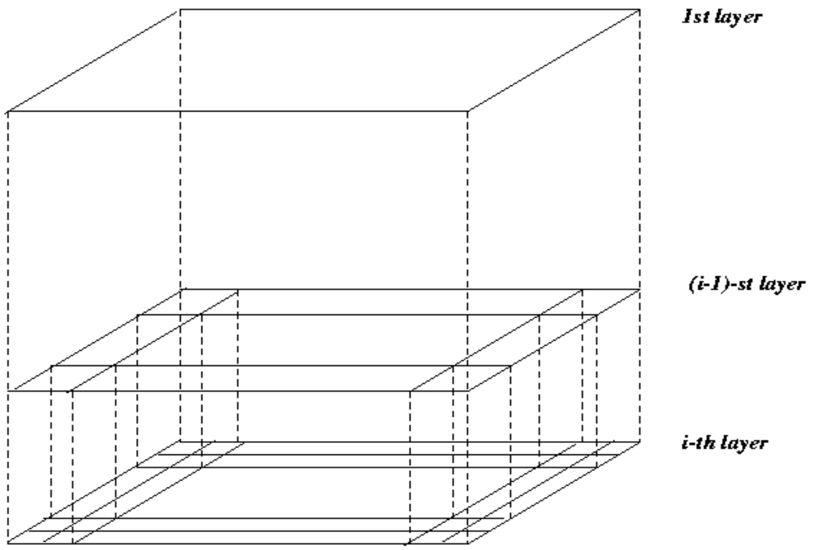
STING (Statistical Information Grid approach)

- **Main Idea**
  - Grid-based, density-based, and hierarchical clustering (using a multi-layer grid structure)
  - Top-down approach

- **Process**
  1) Initially compute statistical parameters for each cell in the bottom level
  2) Go to the top level
  3) Remove irrelevant cells, and go to the next lower level
  4) Repeat (3) until the bottom level is reached

Grid Structure

- **Multi-Layer Grids**
  - The spatial area of data points is divided into rectangular cells
  - There are several different levels of the cells
  - Each cell is partitioned into some smaller cells in the next lower level
  - Statistical parameters, such as count, mean, stdev, min, and max, are pre-computed and stored in each cell in the lowest level
  - Relevance of the cells in higher levels is determined using the statistical parameter values



Summary of STING

- **Strength**
  - Efficient
    - – use only the statistical information in the bottom-level cells after the first scan of DB
  - Able to parallelize
- **Weakness**
  - Inaccurate
  - Cluster boundaries are always horizontal and vertical
  - Only applicable to numeric attributes
- **Reference**
  - Wang, W., Yang, J. and Muntz, R., “STING: A Statistical Information Grid Approach to Spatial Data Mining” In Proceedings of VLDB (1997)

CLIQUE (Clustering in Quest)

- **Main Idea**
  - Grid-based and density-based clustering
  - Applicable to **high-dimensional data** such that each dimension has a different data type
  - Partition each dimension into cells (units)
  - Find sub-dimensional spaces with high-density units
  - A cluster represents a maximal set of connected dense units within a sub-dimensional space

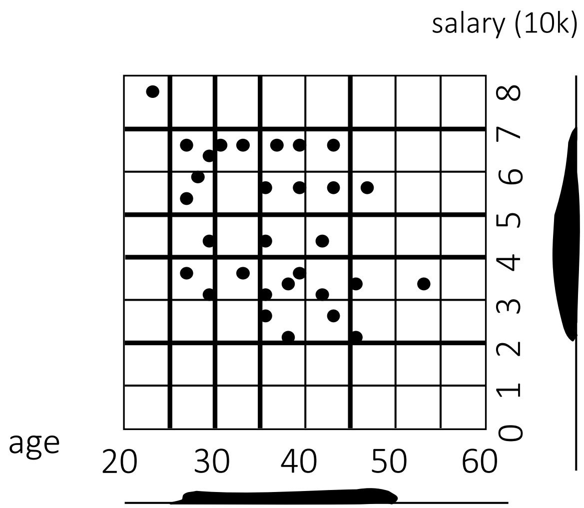
    → **Subspace Clustering**

Subspace Clustering Algorithm

❑ Process

1) Partition each dimension into the same number of equal-length units
2) Identify subspaces that contain dense regions using the Apriori-like algorithm
   → Iterative increment of sub-dimensional spaces with high density
3) Determine dense regions on maximal dimension spaces
4) Combine connecting regions

❑ Example



Summary of CLIQUE

- **Strength**
  - Discovery of informative subspaces in high dimensionality
  - Scalable and efficient in high dimensional data space
- **Weakness**
  - Accuracy depends on the grid size and density threshold
- **Reference**
  - Agrawal, R., et al., “Automatics Subspace Clustering of High Dimensional Data for Data Mining Applications” In Proceedings of SIGMOD (1998)

Overview

1. **Partition-Based Methods**
2. **Hierarchical Methods**
3. **Density-Based Methods**
4. **Grid-Based Methods**
5. **Pattern-Based Methods**
6. **Cluster Validation**

Pattern-based Methods

- **Main Idea**
  - Clustering data objects having similar patterns across dimensions

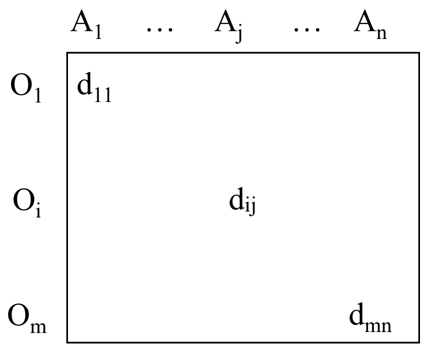
- **Issue**
  - Find clusters in high-dimensional space (Subspace Clustering)
  - Find hidden patterns on a sub-dimensional space

- **Methods**
  - p-Clustering
  - OP-Clustering
  - MAPLE

p-Clustering (Pairwise Clustering)

- **Main Idea**
  - Devised to apply for gene expression data clustering

    e.g., Data with thousands of genes (dimensions)
  - **Bi-clustering**
    - Extension of subspace clustering

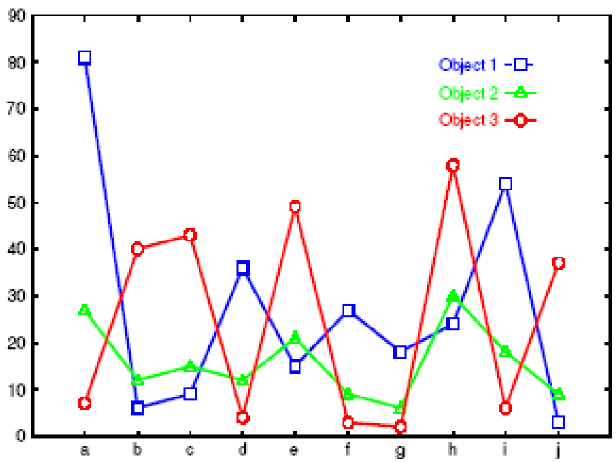


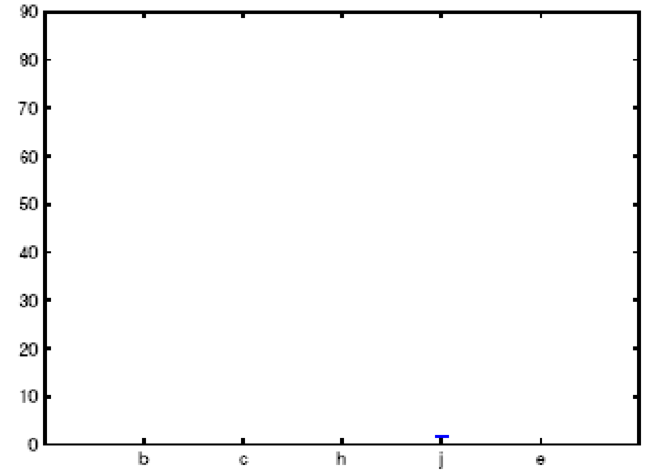
  - Finding similar data patterns (not similar data values)

    e.g., shifting and scaling patterns
  - Can we use the typical Euclidean distance to find similar patterns?

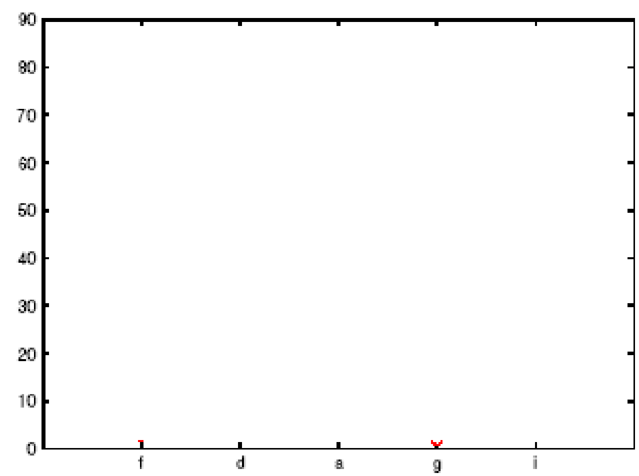
Pattern Examples

❑ **Example**





Shifting pattern



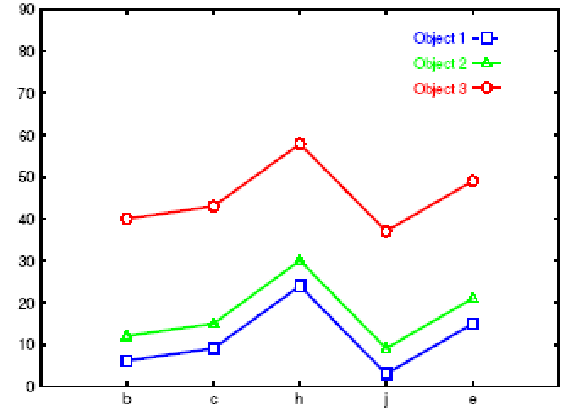
Scaling pattern

Pattern Discovery

❑ Pattern Detection Model

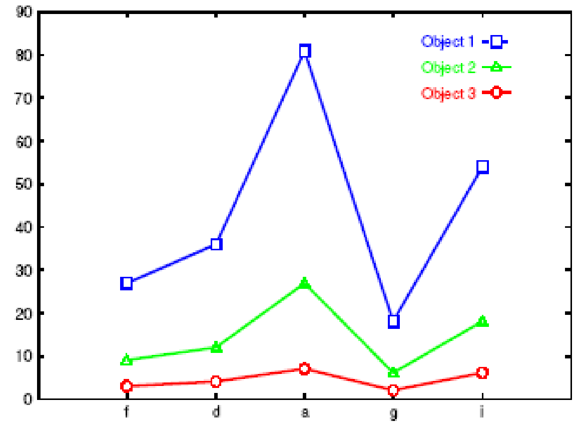
- To detect shifting patterns,

$$pScore\left(\begin{bmatrix} d_{xa} & d_{xb} \\ d_{ya} & d_{yb} \end{bmatrix}\right) = |(d_{xa} - d_{xb}) - (d_{ya} - d_{yb})| \le \delta$$



- To detect scaling patterns,

$$pScore\left(\begin{bmatrix} d_{xa} & d_{xb} \\ d_{ya} & d_{yb} \end{bmatrix}\right) = \frac{d_{xa} / d_{ya}}{d_{xb} / d_{yb}} \le \delta$$

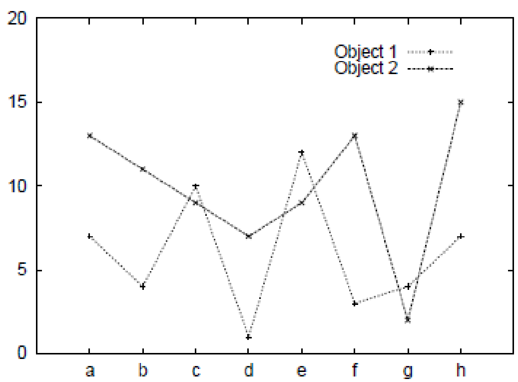


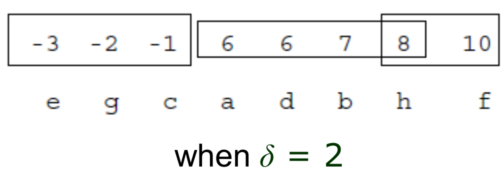
- δ is a user-specified parameter

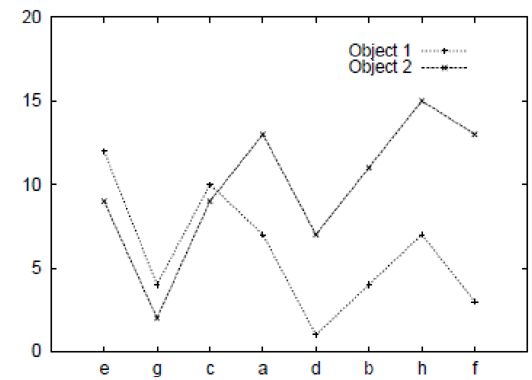
Summary of p-Clustering

Process

- Iterative pairwise clustering
- For each pair of objects,
  1) Sort dimensions in an ascending order of (dx – dy)
  2) Detect maximal size patterns in a maximal dimension space







Reference

- Wang, H., et al, "Clustering by Pattern Similarity in Large Data Sets", In Proceedings of SIGMOD (2002)

Overview

1. **Partition-Based Methods**
2. **Hierarchical Methods**
3. **Density-Based Methods**
4. **Grid-Based Methods**
5. **Pattern-Based Methods**
6. **Cluster Validation**

Cluster Validation

Definition

- Assessing the quality of clustering results

Why Validating?

- To avoid finding clusters formed by chance
- To compare clustering algorithms
- To choose clustering parameters

Methods

- External index: when “ground truth” is available
- Internal index: when “ground truth” is unavailable

Internal Index

- **Error Measures**
  - Absolute error = $|x_i - x_i'|$
  - Squared error = $(x_i - x_i')^2$

- **Sum of Squared Error (SSE)**
  - Measure of cohesiveness by within-cluster sum of squared error

$$WSS = \sum_i \sum_{x \in C_i} (x - m_i)^2$$

  - Measure of separability by between-cluster sum of squared error

$$BSS = \sum_i |C_i| \cdot (m - m_i)^2$$

  - Relationship between WSS and BSS ?

Internal Index

Error Measures

- Absolute error = $|x_i - x_i'|$
- Squared error = $(x_i – x_i')^2$

Sum of Squared Error (SSE)

- Measure of cohesiveness by within-cluster sum of squared error

$$WSS = \sum_i \sum_{x \in C_i} (x - m_i)^2$$

- Measure of separability by between-cluster sum of squared error

$$BSS = \sum_i |C_i| \cdot (m - m_i)^2$$

- Relationship between WSS and BSS ?

External Index – Incident Matrix

- **Notation**
  - N: the total number of data objects
  - $\boldsymbol{C} = \{C_1, C_2, \ldots, C_n\}$: the set of clusters reported by a clustering algorithm
  - $\boldsymbol{P} = \{P_1, P_2, \ldots, P_m\}$: the set of "ground truth" clusters

- **Incident Matrix**
  - $(N \times N)$ matrix
  - $C_{ij} = 1$ if two data objects O_i and O_j belong to the same cluster in $\boldsymbol{C}$

    $C_{ij} = 0$ otherwise
  - $P_{ij} = 1$ if O_i and O_j belong to the same "ground truth" cluster in $\boldsymbol{P}$

    $P_{ij} = 0$ otherwise

External Index – Incident Matrix – Cont'

- **Result Categories**
  - SS: $C_{ij} = 1$ and $P_{ij} = 1$ (agree)
  - DD: $C_{ij} = 0$ and $P_{ij} = 0$ (agree)
  - SD: $C_{ij} = 1$ and $P_{ij} = 0$ (disagree)
  - DS: $C_{ij} = 0$ and $P_{ij} = 1$ (disagree)

- **Validation**
  - Rand Index

$$\text{Accuracy} = \frac{|SS| + |DD|}{|SS| + |DD| + |SD| + |DS|}$$

  - Jaccard Index

$$\text{Accuracy} = \frac{|SS|}{|SS| + |SD| + |DS|}$$

External Index – *f*-Measure

❑ Recall & Precision

- Comparison between an output cluster and a ground-truth cluster
- Let an output cluster X, and a ground-truth cluster Y
- Recall (Sensitivity or True positive rate) = $\frac{|X \cap Y|}{|Y|}$
- Precision (Positive predictive value) = $\frac{|X \cap Y|}{|X|}$

❑ f-Score / f-Measure

- f-score: harmonic mean of Recall and Precision
- f-measure = 2 × (Recall × Precision) / (Recall + Precision)

External Index – Statistical P-Value

P-value of Hyper-Geometric Distribution

- Let the set of all data objects, N
- Let an output cluster X, and a ground-truth cluster Y
- Probability that at least k data objects in X are included in Y
- $$P = 1 - \sum_{i=0}^{k-1} \frac{\binom{|Y|}{i}\binom{|N|-|Y|}{|X|-i}}{\binom{|N|}{|X|}}$$ where k = |X∩Y|
- A low P-value indicates it is less probable that the cluster X is produced by chance
- \- log(P) is usually used for clustering evaluation

Questions?

- Lecture Slides on the Course Website, “https://ads.yonsei.ac.kr/faculty/data_mining”