



PROGRAMA DE VERÃO DO LNCC

JORNADA DE CIÊNCIA DE DADOS

Clustering



Eduardo Ogasawara
<http://eic.cefet-rj.br/~eogasawara>

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

Applications of Cluster Analysis

- Data reduction
 - Summarization: Preprocessing for regression, PCA, classification, and association analysis
- Prediction based on groups
 - Cluster & find characteristics/patterns for each group
- Outlier detection: Outliers are often viewed as those “far away” from any cluster

Clustering: Application Examples

- Biology: taxonomy of living things: kingdom, 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
- 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
 - Discovery of clusters with arbitrary shape
 - Ability to deal with noisy data
 - Incremental clustering and insensitivity to input order
 - High dimensionality

Similarity and Dissimilarity

- Similarity
 - Numerical measure of how alike two data objects are
 - Value is higher when objects are more alike
 - Often falls in the range $[0,1]$
- Dissimilarity (e.g., distance)
 - Numerical measure of how different two data objects are
 - Lower when objects are more alike
 - Minimum dissimilarity is often 0
 - Upper limit varies
- Proximity refers to a similarity or dissimilarity

Data Matrix and Dissimilarity Matrix

- Data matrix

- n data points with p attributes
- Two modes: objects and attributes

$$\begin{bmatrix} x_{11} & \dots & x_{1f} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{if} & \dots & x_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nf} & \dots & x_{np} \end{bmatrix}$$

- Dissimilarity matrix

- n data points, but registers only the distance
- A triangular matrix
- Single mode: distances

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

Proximity Measure for Nominal Attributes

- Can take 2 or more states, e.g., red, yellow, blue, green (generalization of a binary attribute)
- Method 1: Simple matching
 - m: # of matches, p: total # of variables
 - $d(i, j) = \frac{p-m}{p}$
- Method 2: Use a large number of binary attributes
 - creating a new binary attribute for each of the M nominal states

Proximity Measure for Binary Attributes

- A contingency table for binary data
- Distance measure for symmetric binary variables

- $d(i, j) = \frac{r+s}{q+r+s+t}$

- Distance measure for asymmetric binary variables

- $d(i, j) = \frac{r+s}{q+r+s}$

- Jaccard coefficient (*similarity* measure for *asymmetric* binary variables):

- $d(i, j) = \frac{q}{q+r+s}$

		Object j		
		1	0	sum
Object i	1	q	r	q+r
	0	s	t	s+t
	sum	q+s	r+t	p

• Note: Jaccard coefficient is the same as "coherence"

Distance on Numeric Data: Minkowski Distance

- Minkowski distance: A popular distance measure

- $$d(i, j) = \sqrt[h]{|x_{i_1} - x_{j_1}|^h + |x_{i_2} - x_{j_2}|^h + \dots + |x_{i_p} - x_{j_p}|^h}$$

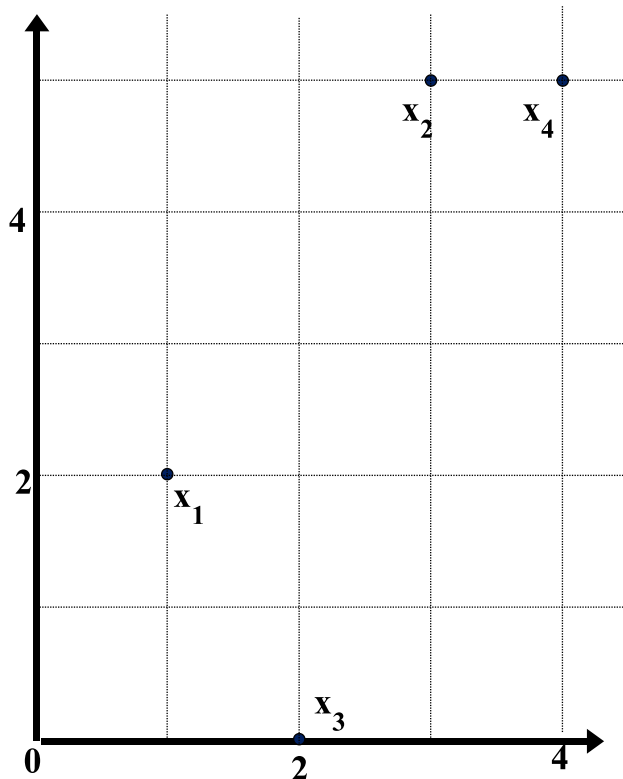
- where $i = (x_{i_1}, x_{i_2}, \dots, x_{i_p})$ and $j = (x_{j_1}, x_{j_2}, \dots, x_{j_p})$ are two p-dimensional data objects, and h is the order (the distance so defined is also called L-h norm)

- Properties

- $d(i, j) > 0$ if $i \neq j$, and $d(i, i) = 0$ (Positive definiteness)
 - $d(i, j) = d(j, i)$ (Symmetry)
 - $d(i, j) \leq d(i, k) + d(k, j)$ (Triangle Inequality)
- A distance that satisfies these properties is a metric

Example: Minkowski Distance

point	attribute 1	attribute 2
x1	1	2
x2	3	5
x3	2	0
x4	4	5



Dissimilarity Matrices

Manhattan (L_1)

L	x1	x2	x3	x4
x1	0			
x2	5	0		
x3	3	6	0	
x4	6	1	7	0

Euclidean (L_2)

L2	x1	x2	x3	x4
x1	0			
x2	3.61	0		
x3	2.24	5.1	0	
x4	4.24	1	5.39	0

Supremum

L_∞	x1	x2	x3	x4
x1	0			
x2	3	0		
x3	2	5	0	
x4	3	1	5	0

Ordinal Variables

- An ordinal variable can be discrete or continuous
- Order is important, e.g., rank
- Can be treated like interval-scaled
 - replace x_{if} by their rank $r_{if} \in \{1, \dots, M_f\}$
 - map the range of each variable onto $[0, 1]$ by replacing i -th object in the f -th variable by $z_{if} = \frac{r_{if}-1}{M_f-1} \{1, \dots, M_f\}$
 - compute the dissimilarity using methods for interval-scaled variables

Cosine Similarity

- A document can be represented by thousands of attributes, each recording the frequency of a particular word (such as keywords) or phrase in the document

<i>Document</i>	<i>teamcoach</i>	<i>hockey</i>	<i>baseball</i>	<i>soccer</i>	<i>penalty</i>	<i>score</i>	<i>win</i>	<i>loss</i>	<i>season</i>	
Document1	5	0	3	0	2	0	0	2	0	0
Document2	3	0	2	0	1	1	0	1	0	1
Document3	0	7	0	2	1	0	0	3	0	0
Document4	0	1	0	0	1	2	2	0	3	0

- Other vector objects: gene features in micro-arrays, ...
- Applications: information retrieval, biologic taxonomy, gene feature mapping, ...
- Cosine measure: If d_1 and d_2 are two vectors (e.g., term-frequency vectors), then
 - $\cos(d_1, d_2) = \frac{(d_1 \cdot d_2)}{|d_1||d_2|}$,
- where \cdot indicates vector dot product, $|d|$: the length of vector d

Example: Cosine Similarity

- $\cos(d1, d2) = (d1 \bullet d2) / (||d1|| \cdot ||d2||)$,
 - where \bullet indicates vector dot product, $||d||$: the length of vector d
- Ex: Find the similarity between documents 1 and 2.
 - $d1 = (5, 0, 3, 0, 2, 0, 0, 2, 0, 0)$
 - $d2 = (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)$
 - $d1 \bullet d2 = 5 \cdot 3 + 0 \cdot 0 + 3 \cdot 2 + 0 \cdot 0 + 2 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 + 2 \cdot 1 + 0 \cdot 0 + 0 \cdot 1 = 25$
 - $||d1|| = (5^2 + 0^2 + 3^2 + 0^2 + 2^2 + 0^2 + 0^2 + 2^2 + 0^2 + 0^2)^{0.5} = (42)^{0.5} = 6.481$
 - $||d2|| = (3^2 + 0^2 + 2^2 + 0^2 + 1^2 + 1^2 + 0^2 + 1^2 + 0^2 + 1^2)^{0.5} = (17)^{0.5} = 4.12$
 - $\cos(d1, d2) = 0.94$

Combining Mixed Types

- A database may contain all attribute types
 - Nominal, symmetric binary, asymmetric binary, numeric, ordinal
- One may use a weighted formula to combine their effects

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

- $\delta_{ij}^f = 0$

- if (1) either x_{if} or x_{jf} is missing
- or $x_{if} = x_{jf} = 0$ and attribute is binary asymmetric

- $\delta_{ij}^f = 1$, otherwise

- f is binary or nominal: $d_{ij}^f = 0$ if $x_{if} = x_{jf}$, or $d_{ij}^f = 1$ otherwise
- f is numeric: use a normalized distance
- f is ordinal: convert to ranks r_{if} and compute z_{if}

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: DBSCAN, OPTICS, DenClue

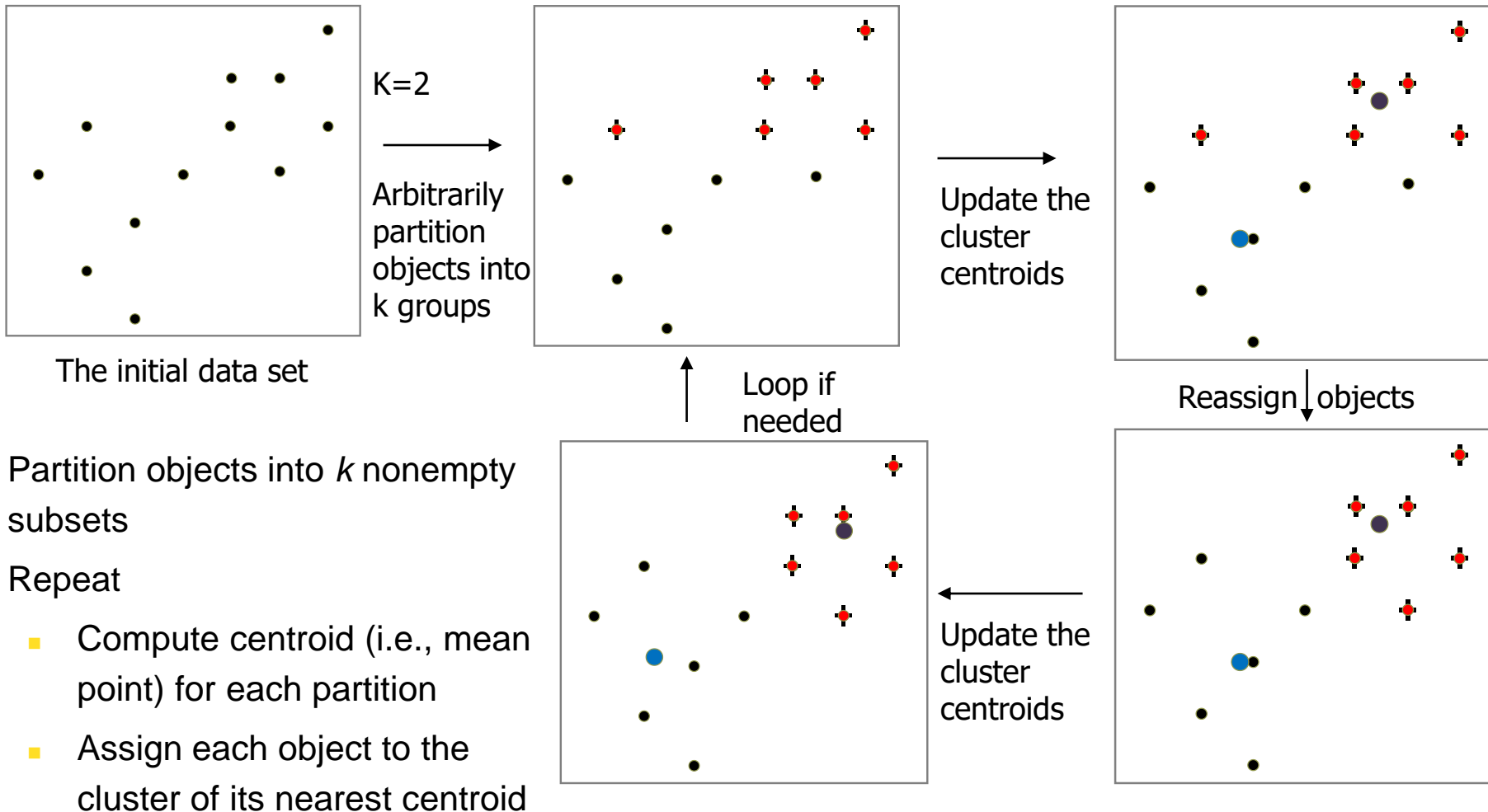
Partitioning Algorithms: Basic Concept

- 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)
- Given k , find a partition of k clusters that optimizes the chosen partitioning criterion
 - $E = \sum_{i=1}^k \sum_{p \in C_i} d(p, c_i)^2$
 - 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:
 - [1] Partition objects into k nonempty subsets
 - [2] 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)
 - [3] Assign each object to the cluster with the nearest seed point
 - [4] Go back to Step 2, stop when the assignment does not change

An Example of K-Means Clustering



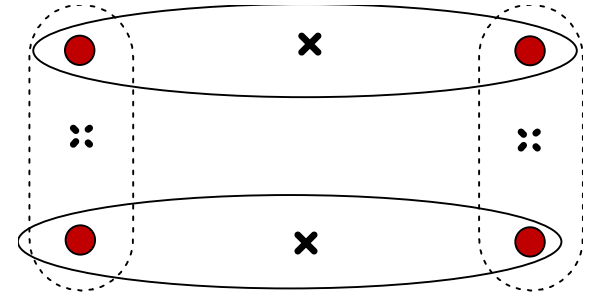
- Partition objects into k nonempty subsets
- Repeat
 - Compute centroid (i.e., mean point) for each partition
 - Assign each object to the cluster of its nearest centroid
- Until no change

Comments on the K-Means Method

- Strength: Efficient: $O(tkn)$, where n is number objects, k is number of clusters, and t is number of 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
 - Sensitive to noisy data and outliers
 - Not suitable to discover clusters with non-convex shapes

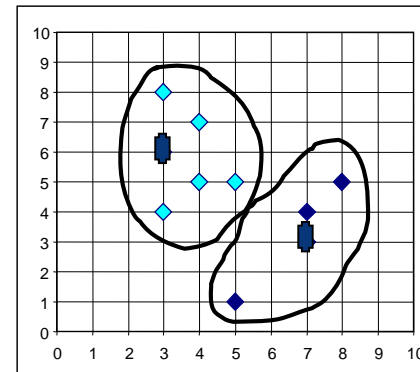
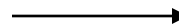
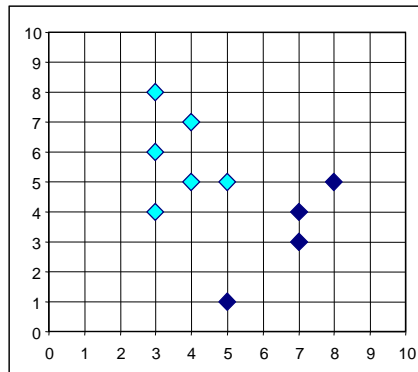
Variations of the K-Means Method

- 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



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

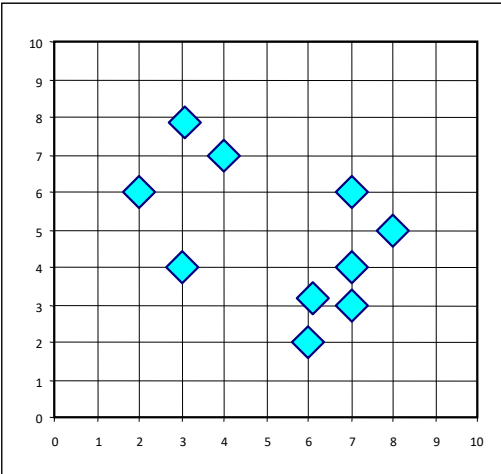


The K-Medoid Clustering Method

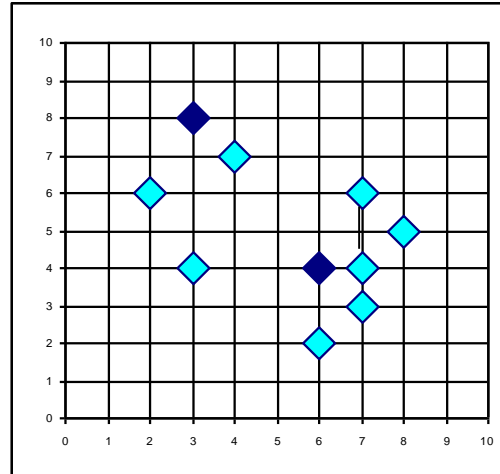
- K-Medoids Clustering
 - Find representative objects (medoids) in clusters
- PAM
 - Starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
 - PAM works effectively for small data sets, but does not scale well for large data sets (due to the computational complexity)
- Efficiency improvement on PAM
 - CLARA : PAM on samples
 - CLARANS: Randomized re-sampling

PAM: A Typical K-Medoids Algorithm

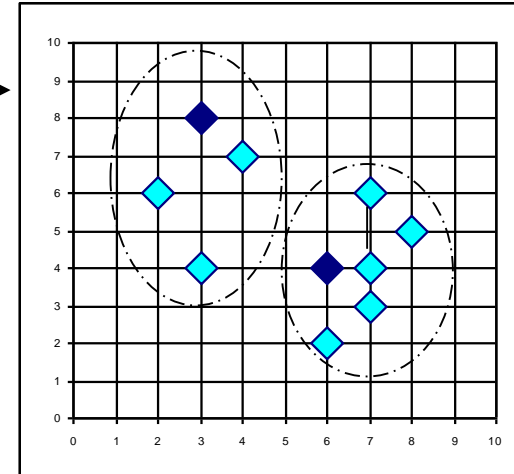
K=2



Arbitrary
choose k
object as
initial
medoids



Assign
each
remainin
g object
to
nearest
medoids



Total Cost = 20

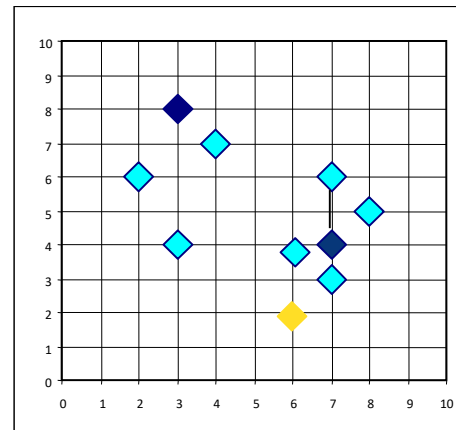
Randomly select a
nonmedoid object, O_{random}

Do loop

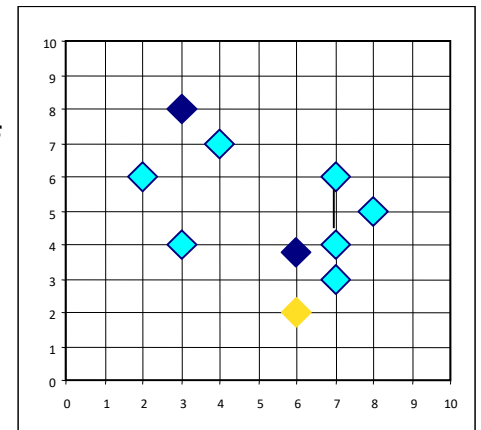
Until no change

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

Total Cost = 26



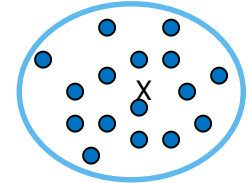
Compute
total cost of
swapping



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

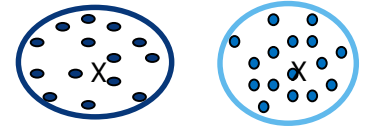
- Diameter:

- square root of average mean squared distance between all

pairs of points in the cluster $D_m = \sqrt{\frac{\sum_{i=1}^n \sum_{i=1}^n (t_{ip} - t_{iq})^2}{n(n-1)}}$

Distance between Clusters

- Single link:



- smallest distance between an element in one cluster and an element in the other, i.e., $dist(K_i, K_j) = \min(t_{ip}, t_{jq})$

- Complete link:

- largest distance between an element in one cluster and an element in the other, i.e., $dist(K_i, K_j) = \max(t_{ip}, t_{jq})$

- Average:

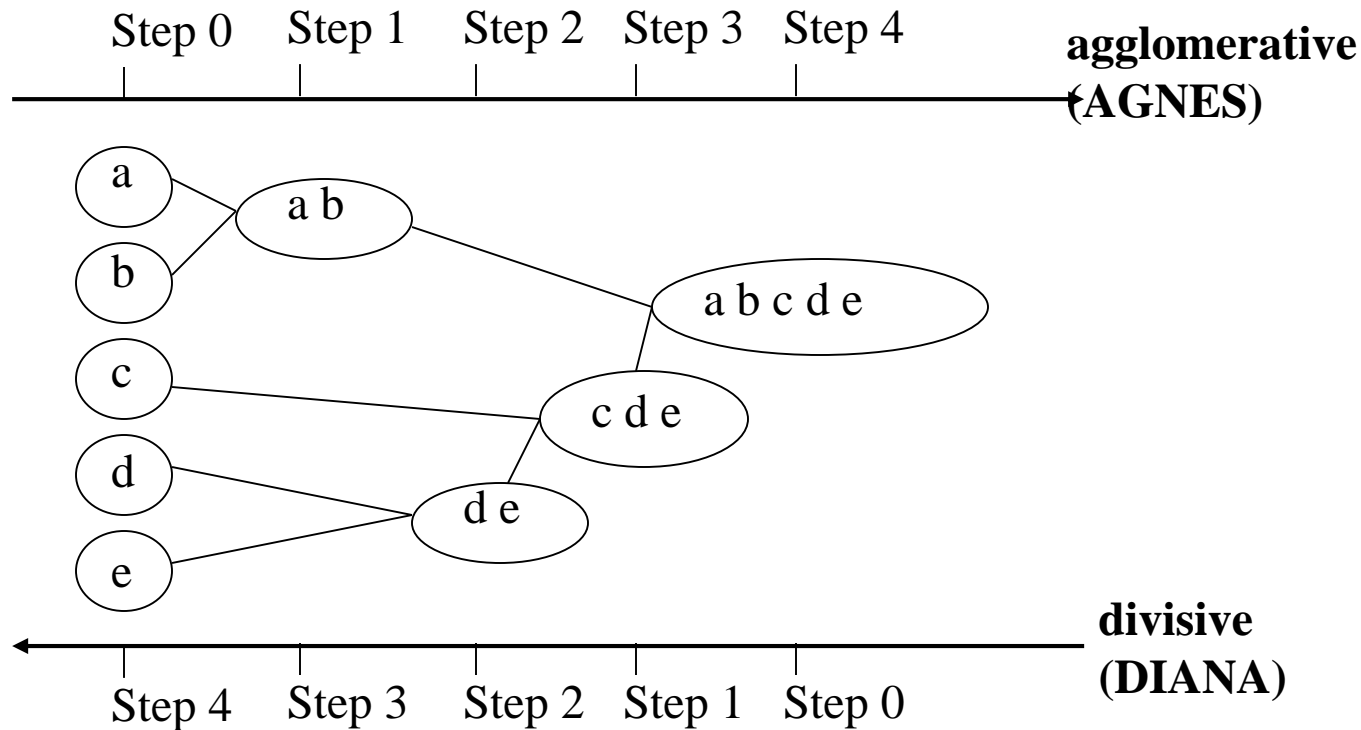
- avg distance between an element in one cluster and an element in the other, i.e., $dist(K_i, K_j) = \text{mean}(t_{ip}, t_{jq})$

- Centroid/Medoid:

- distance between the centroids/medoids of two clusters, i.e., $dist(K_i, K_j) = \text{mean}(c_i, c_j)$

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

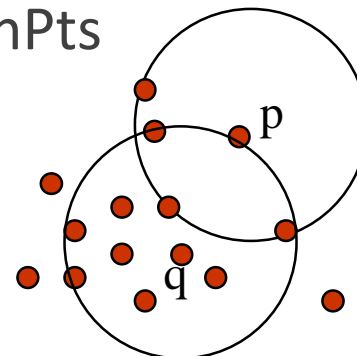


Density-Based Clustering Methods

- 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
 - OPTICS
 - DENCLUE
 - CLIQUE

Density-Based Clustering: Basic Concepts

- Two parameters:
 - *Eps*: Maximum radius of the neighborhood
 - *MinPts*: Minimum number of points in an Eps-neighborhood of that point
- $N_{Eps}(q): \{p \text{ belongs to } D \mid \text{dist}(p, q) \leq Eps\}$
- 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: $|N_{Eps}(q)| \geq \text{MinPts}$

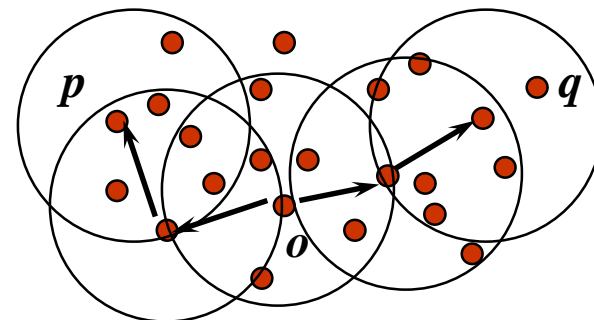
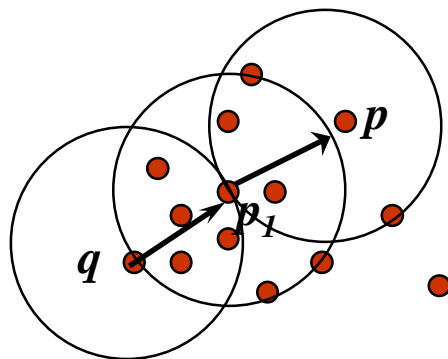


MinPts = 5

Eps = 1 cm

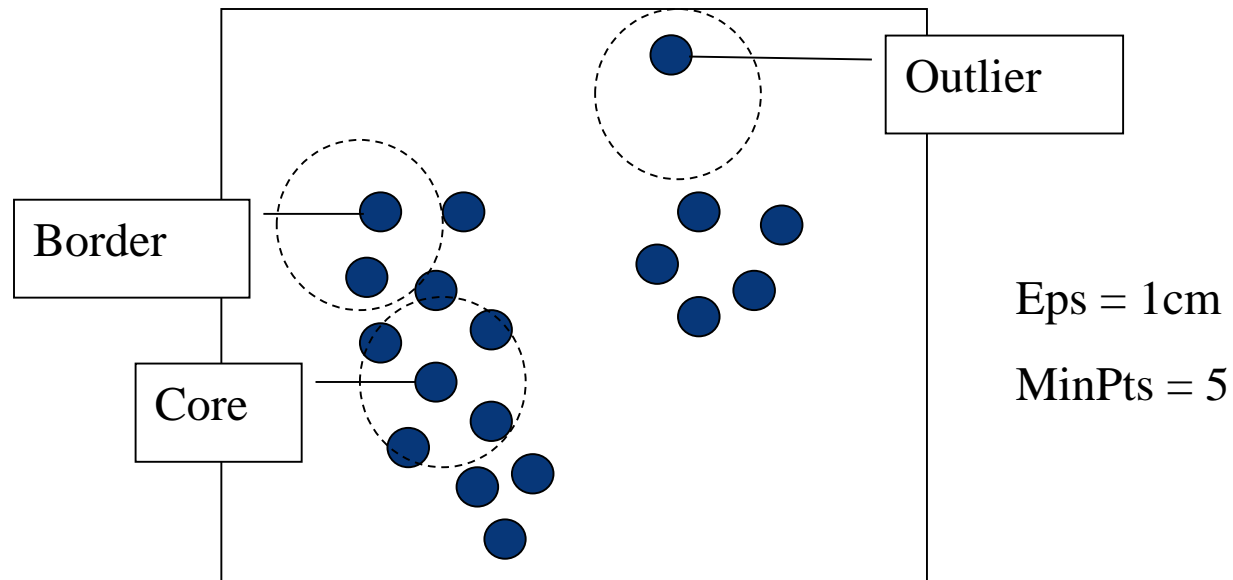
Density-Reachable and Density-Connected

- 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
- If a spatial index is used, the computational complexity of DBSCAN is $O(n \cdot \log(n))$, where n is the number of database objects. Otherwise, the complexity is $O(n^2)$

Measuring Clustering Quality

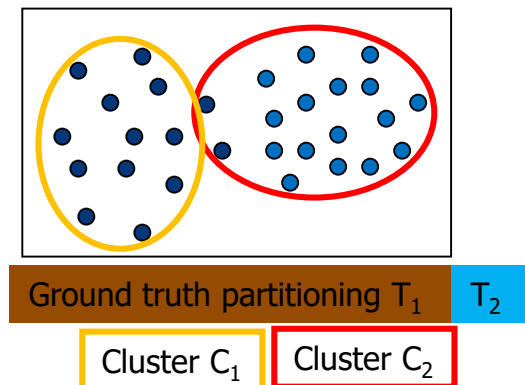
- Three kinds of measures:
 - External
 - Internal
 - Relative
- External: supervised, employ criteria not inherent to the dataset
 - Compare a clustering against prior or expert-specified knowledge (i.e., the ground truth) using certain clustering quality measure
- Internal: unsupervised, criteria derived 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
- Relative: directly compare different clustering, usually those obtained via different parameter settings for the same algorithm

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 essential criteria
 - Cluster homogeneity: the purer, the better

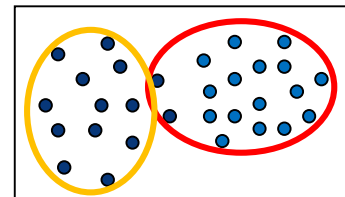
Some Commonly Used External Measures

- Matching-based measures
 - Purity, maximum matching, F-measure
- Entropy-Based Measures
 - Conditional entropy
 - Normalized mutual information (NMI)
 - Variation of information
- Pair-wise measures
 - Four possibilities: TP, FN, FP, TN
 - Jaccard coefficient



Entropy-Based Measure (I): Conditional Entropy

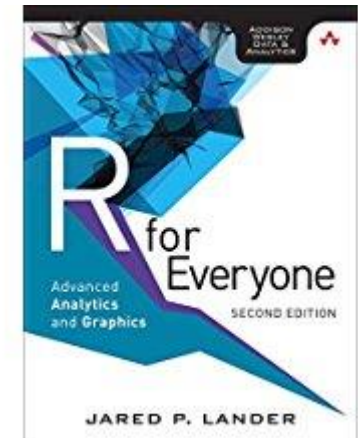
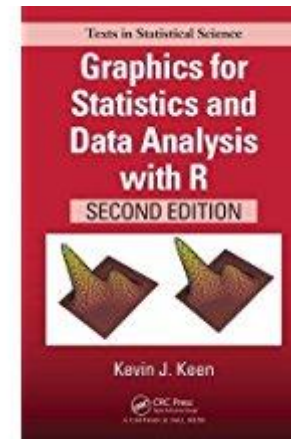
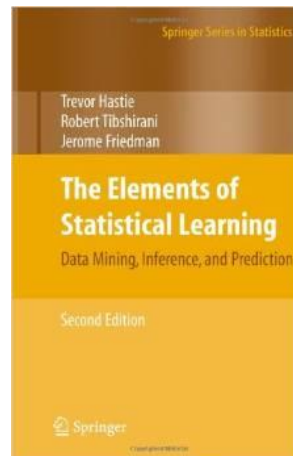
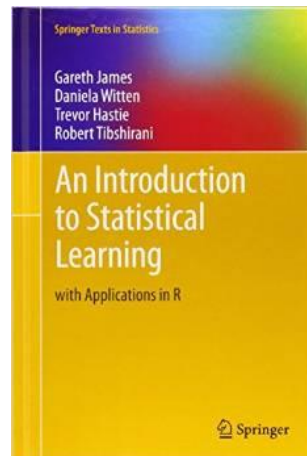
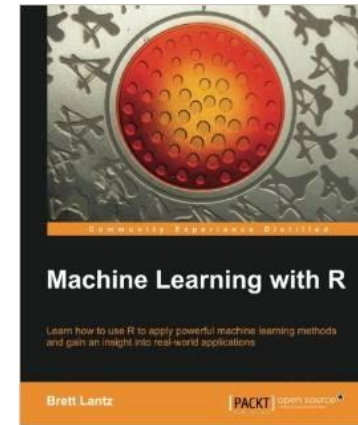
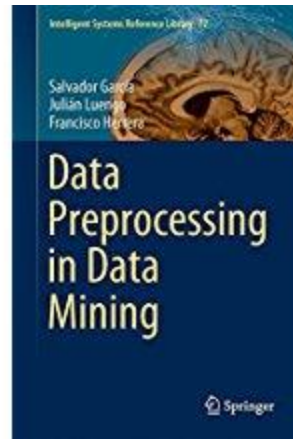
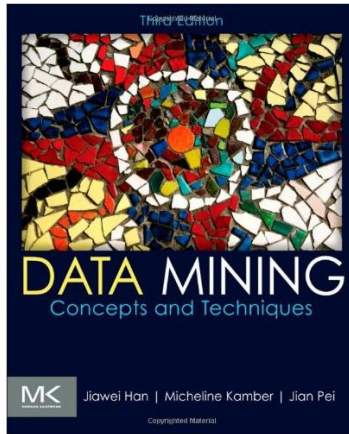
- Entropy of clustering C : $H(C) = - \sum_{i=1}^r p_{C_i} \cdot \log(p_{C_i})$
 - $p_{C_i} = \frac{n_i}{n}$, the probability of cluster C_i
- Entropy of partitioning T : $H(T) = - \sum_{i=1}^r p_{T_i} \cdot \log(p_{T_i})$
- Entropy of T w.r.t. cluster C_i : $H(T|C_i) = - \sum_{j=1}^k \left(\frac{n_{ij}}{n_i}\right) \log\left(\frac{n_{ij}}{n_i}\right)$
- Conditional entropy of T w.r.t. clustering C :
 - $H(T|C) = - \sum_{j=1}^k \left(\frac{n_i}{n}\right) H(T|C_i) = - \sum_{i=1}^r \sum_{j=1}^k p_{ij} \cdot \log\left(\frac{p_{ij}}{p_{C_i}}\right)$
 - $H(T|C) = - \sum_{i=1}^r \sum_{j=1}^k p_{ij} \cdot \log(p_{ij}) + \sum_{i=1}^r (\log(p_{C_i}) \cdot \sum_{j=1}^k p_{ij})$
 - $= - \sum_{i=1}^r \sum_{j=1}^k p_{ij} \cdot \log(p_{ij}) + \sum_{i=1}^r (p_{C_i} \cdot \log(p_{C_i})) = H(C, T) - H(C)$
 - 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 $\log k$



Practicing

- Take some time to practice the examples
 - <https://nbviewer.jupyter.org/github/eogasawara/mylibrary/blob/master/myClustering.ipynb>

Main References



Most of the slides were extracted from
Data Mining Concepts and Techniques