DATABASE SYSTEMS GROUP

Ludwig-Maximilians-Universität München
Institut für Informatik
Lehr- und Forschungseinheit für Datenbanksysteme

LMU

Lecture notes

Knowledge Discovery in Databases

Summer Semester 2012

Lecture 7: Clustering I

Lecture: Dr. Eirini Ntoutsi
Tutorials: Erich Schubert

http://www.dbs.ifi.lmu.de/cms/Knowledge_Discovery_in_Databases_I_(KDD_I)

Sources

- Previous KDD I lectures on LMU (Johannes Aßfalg, Christian Böhm, Karsten Borgwardt, Martin Ester, Eshref Januzaj, Karin Kailing, Peer Kröger, Jörg Sander, Matthias Schubert, Arthur Zimek)

- Jiawei Han, Micheline Kamber and Jian Pei, *Data Mining: Concepts and Techniques, 3rd ed.,* Morgan Kaufmann, 2011.

- Margaret Dunham, Data Mining, *Introductory and Advanced Topics*, Prentice Hall, 2002.

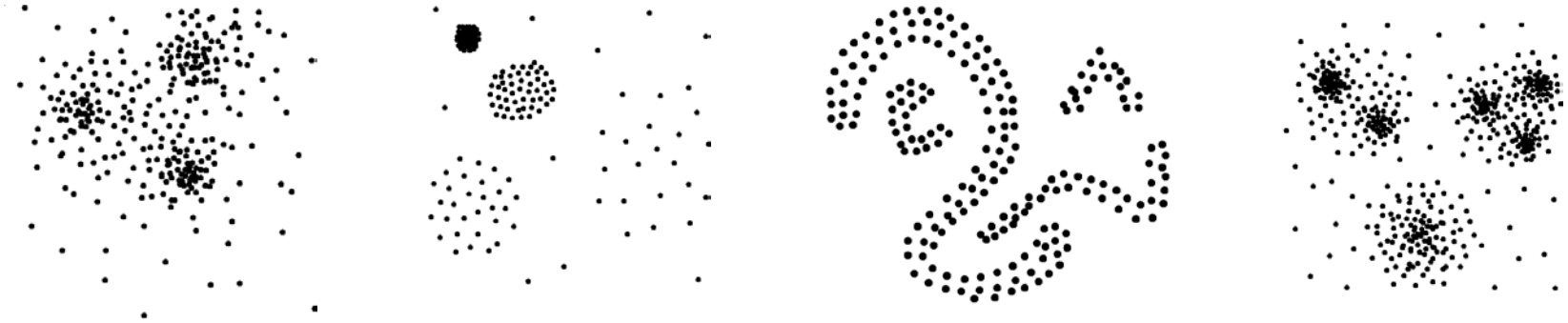
- Tan P.-N., Steinbach M., Kumar V., *Introduction to Data Mining*, Addison-Wesley, 2006

Outline

- Introduction
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Things you should know
- Homework/tutorial

What is cluster analysis?

- Cluster: a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters
- Cluster analysis
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters

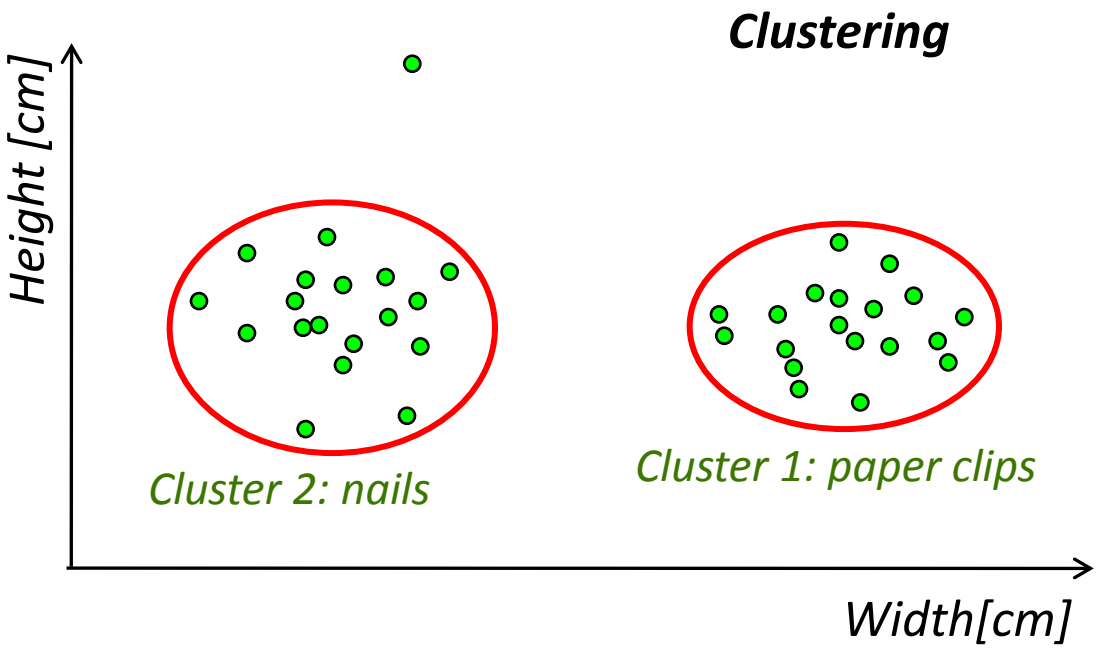


An unsupervised learning task

- Clustering is an unsupervised learning task
  - Given a set of measurements, observations, etc., the goal is to group the data into groups of similar data (clusters)
  - We are given a dataset as input which we want to cluster but there are no class labels
  - We don't know how many clusters exist in the data
  - We don't know the characteristics of the individual clusters

- In contrast to classification, which is a supervised learning task
  - Supervision: The training data (observations, measurements, etc.) are accompanied by *labels* indicating the *class* of the observations
  - New data is classified based on the training set

Unsupervised learning example



Question:
Is there any structure in data (based on their characteristics, i.e., width, height)?

Supervised learning example

Classification model

- *Screw*

- *Nails*

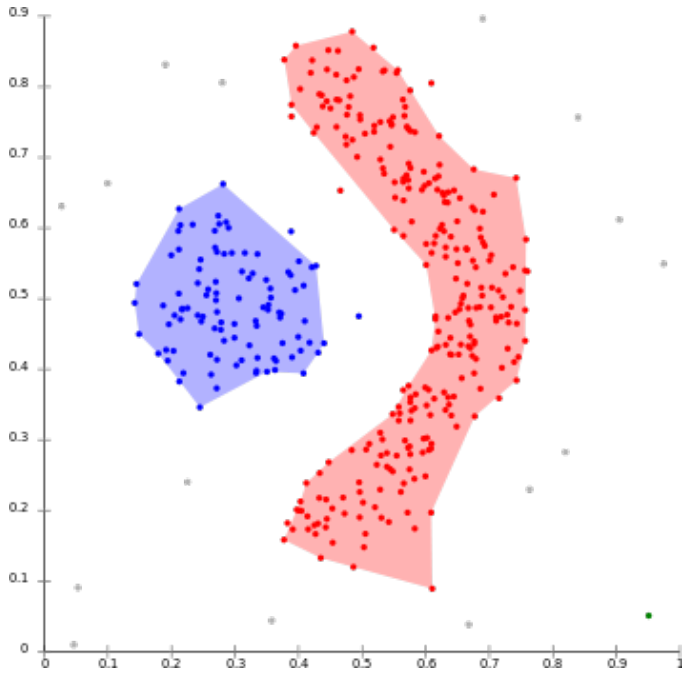
- *Paper clips*

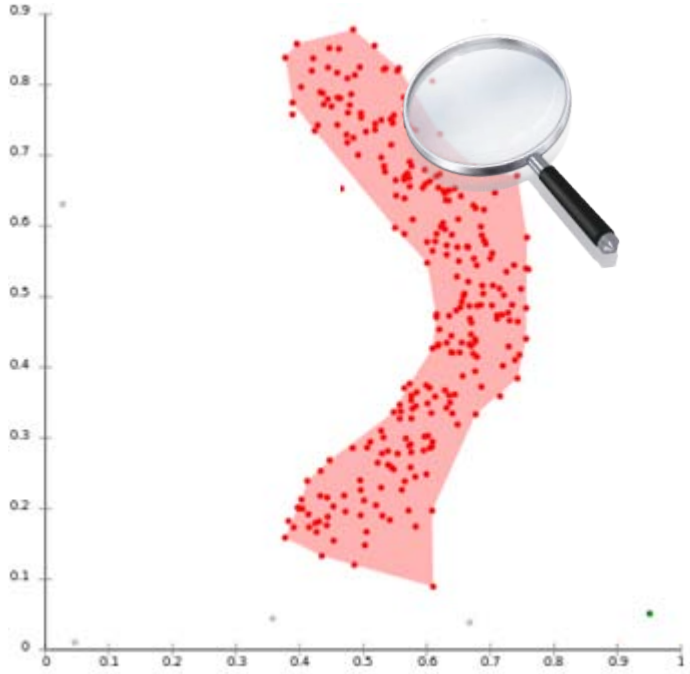
- *New object (unknown class)*

Question:
What is the class of a new object???
Is it a screw, a nail or a paper clip?

Why clustering?

- Clustering is widely used as:
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms





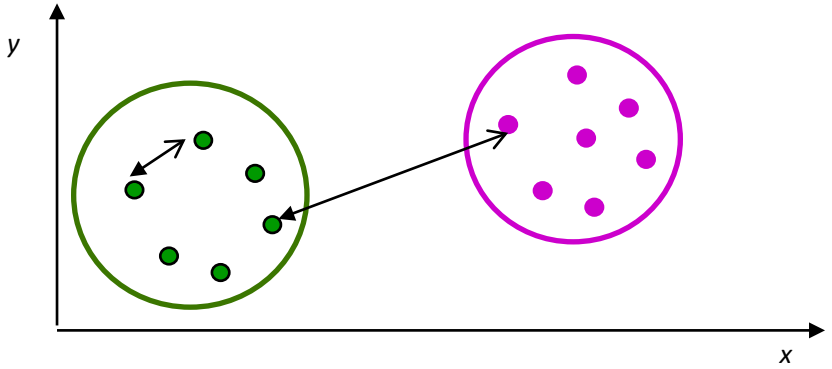
http://en.wikipedia.org/wiki/Cluster_analysis

Example applications

- Marketing:
  - Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Telecommunications:
  - Build user profiles based on usage and demographics and define profile specific tariffs and offers
- Land use:
  - Identification of areas of similar land use in an earth observation database
- City-planning:
  - Identifying groups of houses according to their house type, value, and geographical location
- Bioinformatics:
  - Cluster similar proteins together (similarity wrt chemical structure and/or functionality etc)
- Web:
  - Cluster users based on their browsing behavior
  - Cluster pages based on their content (e.g. News aggregators)

The clustering task I

- **Goal:** Group objects into groups so that the objects belonging in the same group are similar (high intra-class similarity), whereas objects in different groups are different (low inter-class similarity)
- A good clustering method will produce high quality clusters with
  - high intra-class similarity
  - low inter-class similarity



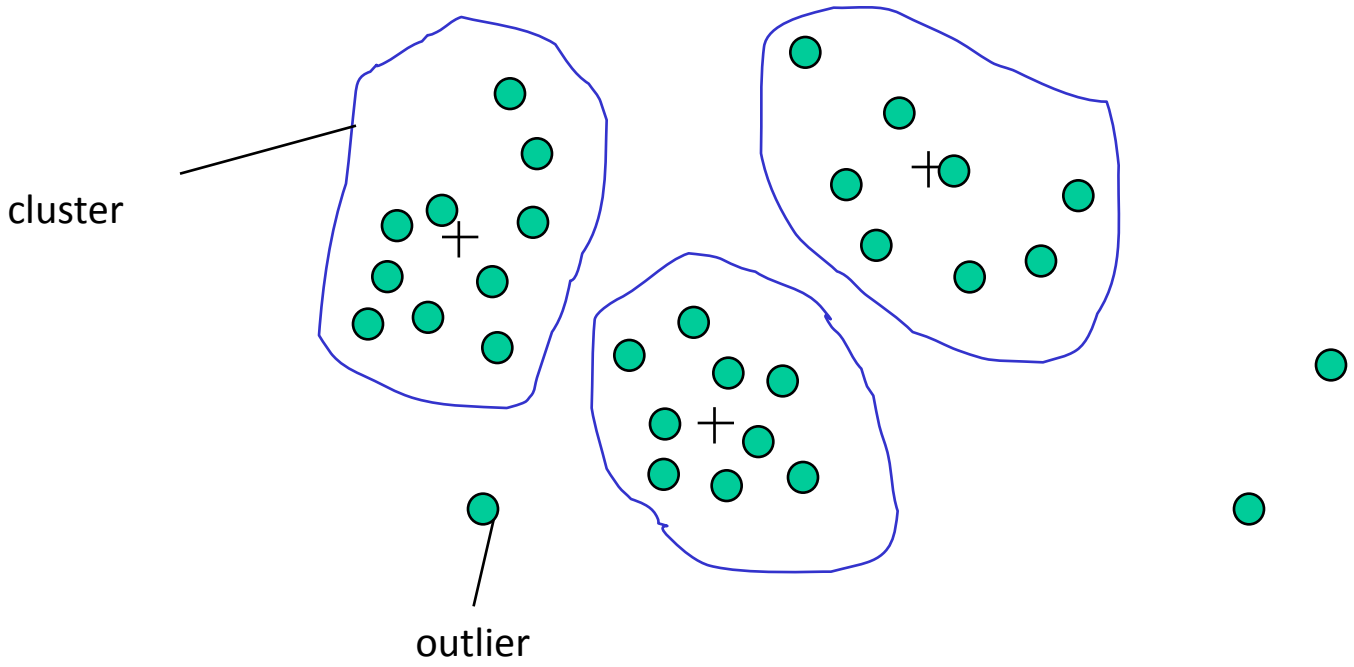
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation

Requirements of clustering in Data Mining

- Scalability
- Ability to deal with different types of attributes
- Ability to handle dynamic data
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- High dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability

Outliers

- There might be objects that do not belong to any cluster



- There are cases where we are interested in detecting outliers not clusters
- More on outlier analysis in an upcoming lecture

Outline

- Introduction
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Things you should know
- Homework/tutorial

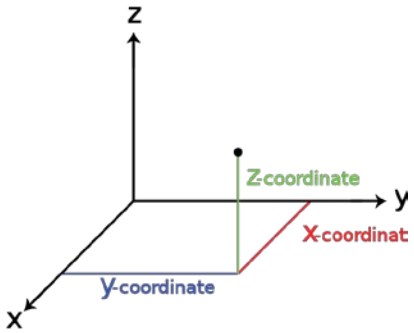
Type of data in clustering analysis

- Real-value attributes
  - e.g., salary, height
- Binary/ Dichotomous attributes
  - e.g., gender (M/F), has_cancer(T/F)
- Categorical/ Nominal attributes
  - e.g., religion (Christian, Muslim, Buddhist, Hindu, etc.)
- Ordinal/Ranked attributes
  - e.g., military rank (soldier, sergeant, lutenant, captain, etc.)
- Variables of mixed types
  - multiple attributes with various types

Features/ Attributes/ Dimensions

- Objects are described through features/ attributes/ variables/ dimensions
  - e.g., ("J. Smith", 20K, 30, 5)

- If all d dimensions are real-valued then we can visualize each data point as points in a d-dimensional space



http://en.wikipedia.org/wiki/Three-dimensional_space

- If all d dimensions are binary then we can think of each data point as a binary vector

0	1	0	0

Distance functions

- The distance d(x, y) between two objects x and y is a metric if
  - d(i, j)≥0 *(non-negativity)*
  - d(i, i)=0 *(isolation)*
  - d(i, j)= d(j, i) *(symmetry)*
  - d(i, j) ≤ d(i, h)+d(h, j) *(triangular inequality)*
- The definitions of distance functions are usually different for real, boolean, categorical, and ordinal variables.
- Weights may be associated with different variables based on applications and data semantics.

Data structures

- Data matrix
  - Rows: objects
  - Columns: dimensions

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

- Dissimilarity matrix
  - Rows: objects
  - Columns: objects

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

Dissimilarity between binary attributes (from Lecture 2)

- A binary attribute has two states: 0 (absence), 1 (presence)
- A contingency table for binary data

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

- Simple matching coefficient

  (for symmetric binary variables)

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

- for asymmetric binary variables:

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

- Jaccard coefficient

  (for *asymmetric* binary variables)

$$sim_{Jaccard}(i,j) = \frac{q}{q+r+s}$$

Example

Name	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	1	0	1	0	0	0
Mary	1	0	1	0	1	0
Jim	1	1	0	0	0	0

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

Dissimilarity measures between categorical/ nominal attributes (from Lecture 2)

- A categorical/ nominal attribute has >2 states (generalization of a binary attribute)
  - e.g. color={red, blue, green}
- Method 1: Simple matching
  - m: # of matches, p: total # of variables

$$d(i, j) = \frac{p - m}{p}$$

- Method 2: Map it to binary variables
  - create a new binary attribute for each of the *M* nominal states of the attribute

Dissimilarity between real-valued attributes (from Lecture 2)

- L_p norms or Minkowski distance: $L_p = (|p_1-q_1|^p + |p_2-q_2|^p + ...+|p_d-q_d|)^{1/p}$
  - where p is a positive integer

- If p = 1 → Manhattan-norm (or city block) distance: $L_1= |p_1-q_1|+|p_2-q_2|+... |p_d-q_d|$
  - The sum of the absolute differences of their coordinates

- If p=2→ Euclidean Norm (L_2): $L_2= ((p_1-q_1)^2+(p_2-q_2)^2+...)^{1/2}$
  - The length of the line segment connecting p and q

- One can use weighted distance also
  - Weighted $L_1 = w_1|p_1-q_1|+w_2|p_2-q_2|+...+w_d|p_d-q_d|$
  - Weighted $L_2 = (w_1(p_1-q_1)^2+w_2 (p_2-q_2)^2+... +w_d (p_d-q_d)^2)^{1/2}$

Normalization (from Lecture 2)

- Attributes with large ranges outweigh ones with small ranges
  - e.g. income [10K-100K]; age [10-100]
- To balance the "contribution" of each attribute in the resulting distance, the attributes are scaled to fall within a small, specified range
- min-max normalization: to [new_min_A , new_max_A]

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

  - e.g. normalize age=30 in [0-1], when min=10,max=100. new_age=(30-10)/(100-10)=2/9
- z-score normalization

$$v' = \frac{v - mean_A}{stand\_dev_A}$$

e.g. normalize 70,000 iff μ=50,000, σ=15,000.
new_value = (70,000-50,000)/15,000=1.33

- Normalization by decimal scaling

$$v' = \frac{v}{10^j}$$

j is the smallest integer such that Max(|v'|) < 1

e.g., if v lies in [1, 999], if we set j=3 then v' will lie in [0.001, 0,999]

Vector objects

- Vector objects: keywords in documents, gene features in micro-arrays, etc.
- Broad applications: information retrieval, biologic taxonomy, etc.
- Cosine measure

$$s(\vec{X}, \vec{Y}) = \frac{\vec{X}^t \cdot \vec{Y}}{|\vec{X}||\vec{Y}|},$$

$\vec{X}^t$ is a transposition of vector $\vec{X}$, $|\vec{X}|$ is the Euclidean normal of vector $\vec{X}$, $|\mathbf{x}|_2 = |\mathbf{x}| = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}$.

- A variant: Tanimoto coefficient

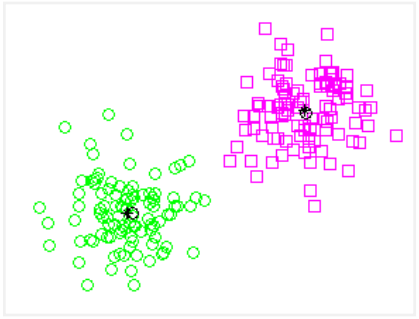
$$s(\vec{X}, \vec{Y}) = \frac{\vec{X}^t \cdot \vec{Y}}{\vec{X}^t \cdot \vec{X} + \vec{Y}^t \cdot \vec{Y} - \vec{X}^t \cdot \vec{Y}},$$

Outline

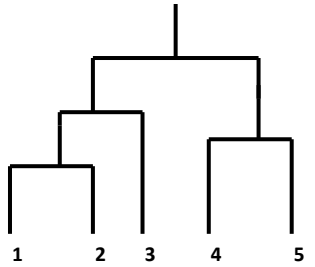
- Introduction
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Things you should know
- Homework/tutorial

Major clustering methods 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



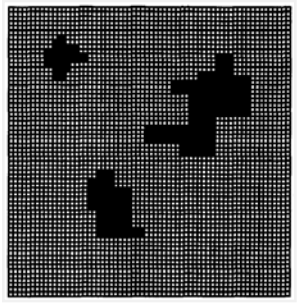
- Hierarchical approach:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, ROCK, CHAMELEON



- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSCAN, OPTICS, DenClue

Major clustering methods II

- Grid-based approach:
  - based on a multiple-level granularity structure
  - Typical methods: STING, WaveCluster, CLIQUE



- 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: pCluster
- User-guided or constraint-based:
  - Clustering by considering user-specified or application-specific constraints
  - Typical methods: COD (obstacles), constrained clustering

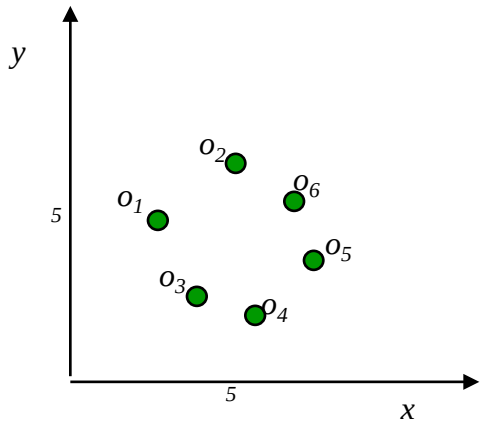
Cluster descriptors (numerical data)

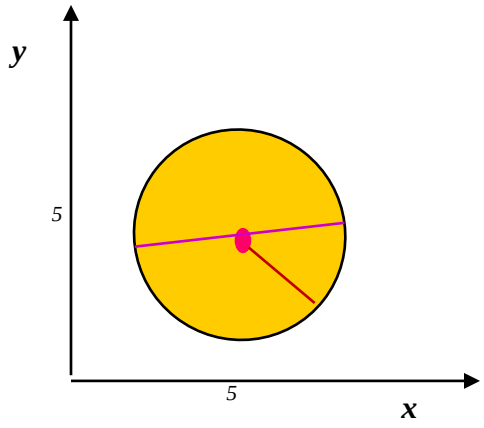
- Centroid: the "middle" of a cluster
- Radius: square root of average distance from any point of the cluster to its centroid
- Diameter: square root of average mean squared distance between all pairs of points in the cluster

$$c_m = \frac{\sum_{i=1}^{n} p_i}{n}$$

$$r_m = \sqrt{\frac{\sum_{i=1}^{n} (p_i - c_m)^2}{n}}$$

$$d_m = \sqrt{\frac{\sum_{i=1}^{n}\sum_{i=1}^{n} (p_i - p_j)^2}{n(n-1)}}$$





Typical alternatives to calculate the distance between clusters

- Single link: smallest distance between an element in one cluster and an element in the other, i.e.,

$$dis_{sl}(C_i, C_j) = \min_{x,y}\{d(x,y) \mid x \in C_i, y \in C_j\}$$

- Complete link: largest distance between an element in one cluster and an element in the other, i.e.,

$$dis_{cl}(C_i, C_j) = \max_{x,y}\{d(x,y) \mid x \in C_i, y \in C_j\}$$

- Average: avg distance between an element in one cluster and an element in the other, i.e.,

$$dis_{avg}(C_i, C_j) = \frac{\sum_{x \in C_i, y \in C_j} d(x,y)}{|C_i||C_j|}$$

- Centroid: distance between the centroids of two clusters, i.e.,

$$dis_{centroids}(C_i, C_j) = d(c_i, c_j)$$

- Medoid: distance between the medoids of two clusters, i.e., $dis(K_i, K_j) = dis(M_i, M_j)$
  - Medoid: one chosen, centrally located object in the cluster

Outline

- Introduction
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Things you should know
- Homework/tutorial

Partitioning methods idea

- Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters
  - Each object belongs to exactly one cluster (hard clustering)
  - The number of clusters k is given in advance

- The partition should optimize the chosen partitioning criterion
  - e.g., minimize the intra-cluster variance, i.e., the sum of the squared distances from each data point to its cluster center.
  - Possible solutions:
    - Global optimal: exhaustively enumerate all partitions
    - Heuristic methods: k-means and k-medoids algorithms
    - k-means (MacQueen'67): 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 problem

- Given a set X of n points in a d-dimensional space and an integer k

- Task: choose a set of k points $\{c_1, c_2, \ldots, c_k\}$ in the d-dimensional space to form clusters $\{C_1, C_2, \ldots, C_k\}$ such that

$$Cost(C) = \sum_{i=1}^{k} \sum_{x \in C_i} (x - c_i)^2$$

Cluster cost

Clustering cost

is minimized.

- Some special cases: k = 1, k = n

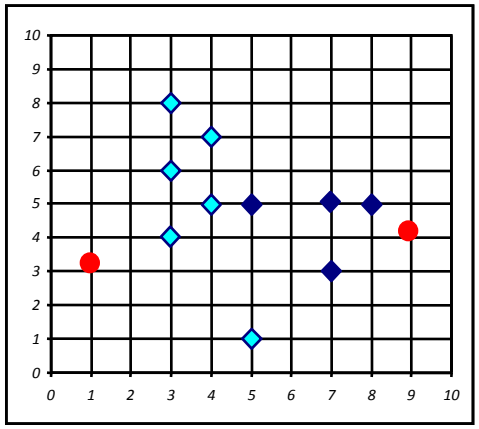
The k-Means algorithm

- Given k, the k-Means algorithm is implemented in four steps:
  - Randomly pick k objects as cluster centers $\{c_1,\ldots,c_k\}$
  - Assign the rest of the points to their closest cluster centers.
  - Update the center of each cluster based on the new point assignments.
  - Repeat until convergence

- Complexity
  - Relatively efficient: O(tkn), where n is # objects, k is # clusters, and t is # iterations. Normally, k, t << n.

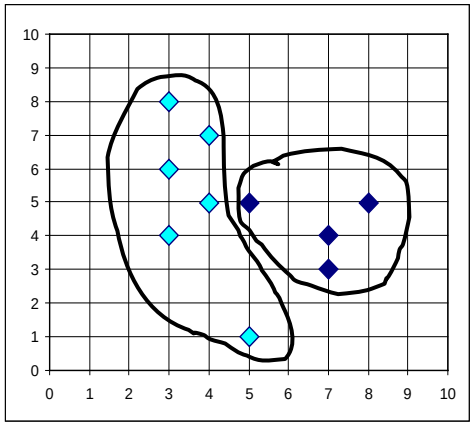
k-Means example

k=2

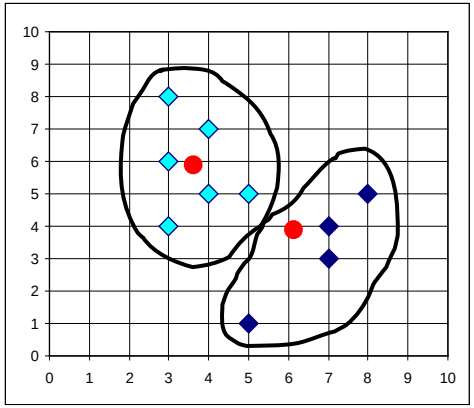


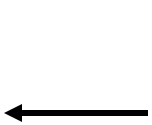
Arbitrarily choose K objects as initial cluster center

Assign each objects to most similar center

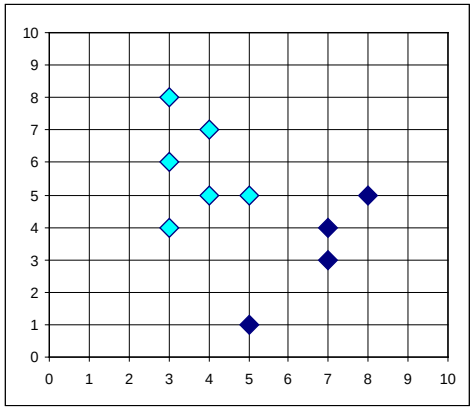


reassign

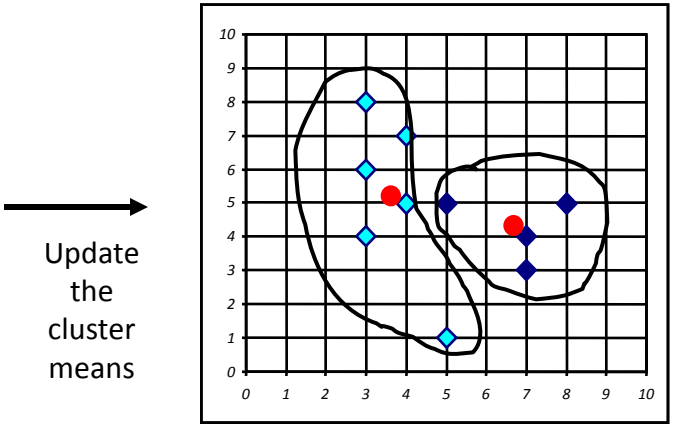




Update the cluster means



Reassign

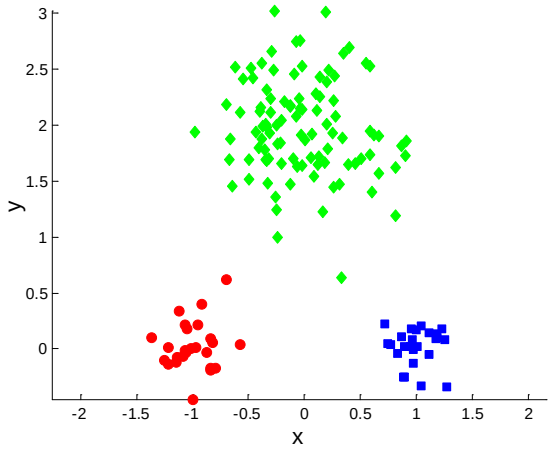


Update the cluster means

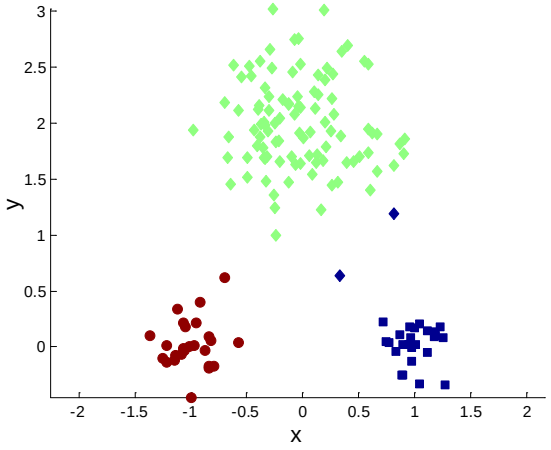
k-Means overview

- Relatively efficient: O(tkn), where n is # objects, k is # clusters, and t is # iterations. Normally, k, t << n.
  - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$
- Finds a local optimum
  - The global optimum may be found using techniques such as: deterministic annealing and genetic algorithms
- The choice of initial points can have large influence in the result
- Weakness
  - Need to specify k, the number of clusters, in advance
  - Unable to handle noisy data and outliers
  - Not suitable to discover clusters with non-convex shapes
  - Applicable only when mean is defined, then what about categorical data?

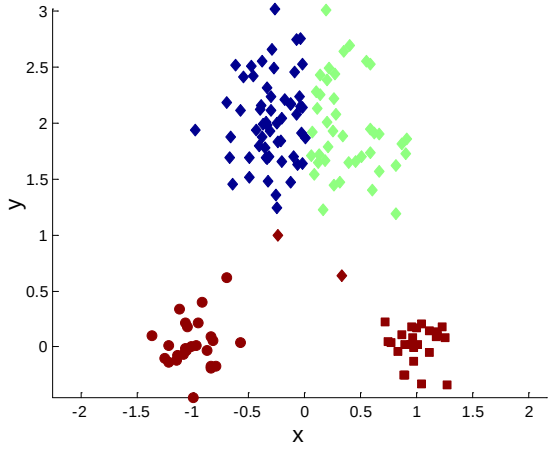
Two different K-means clusterings



Original Points



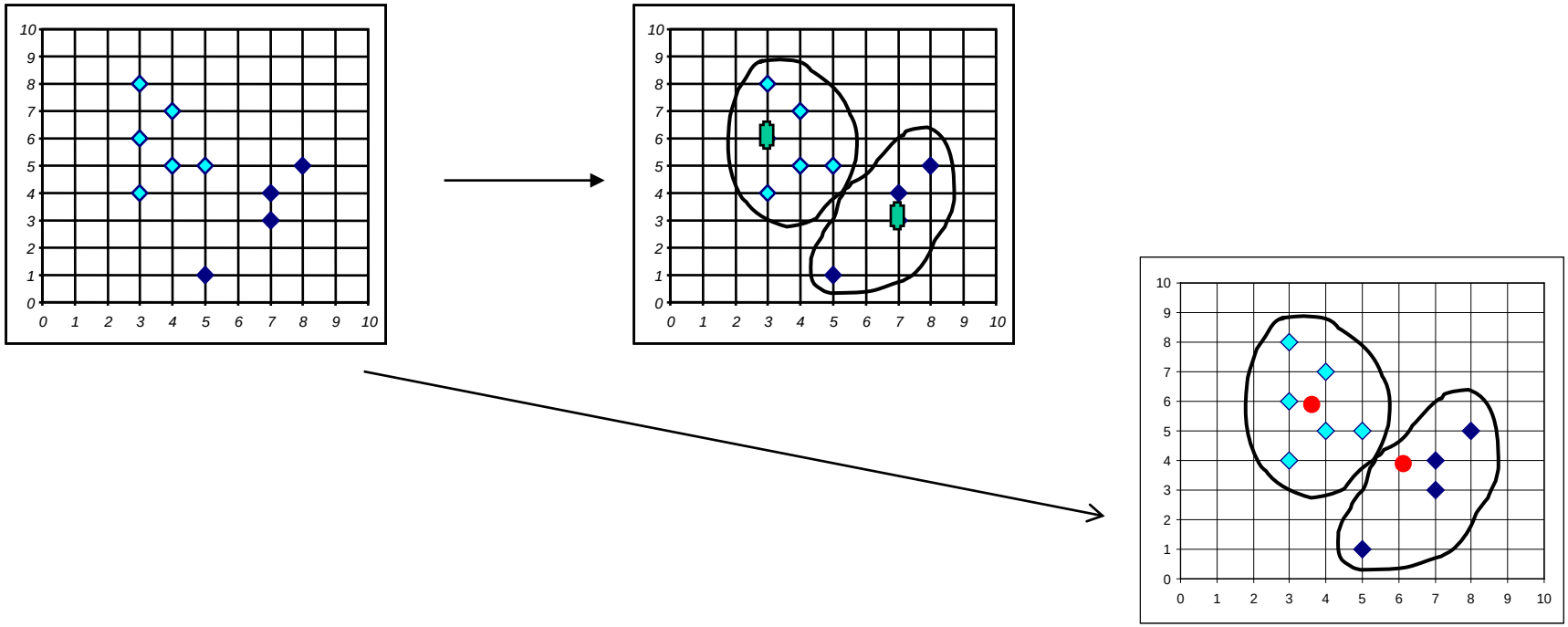
Optimal Clustering



Sub-optimal Clustering

The problem with outliers

- 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.



k-Means variations

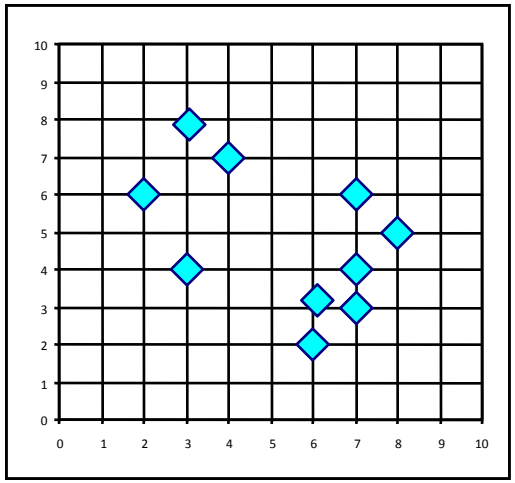
- A few variants of the *k-means* which differ in
  - Selection of the initial *k* means
    - Multiple runs
    - Not random selection of centers. e.g., pick the most distant (from each other) points as cluster centers (kMeans++ algorithm)
  - Dissimilarity calculations
  - Strategies to calculate cluster means
- Handling categorical data: *k-modes* (Huang'98)
  - 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

The k-Medoids clustering algorithm

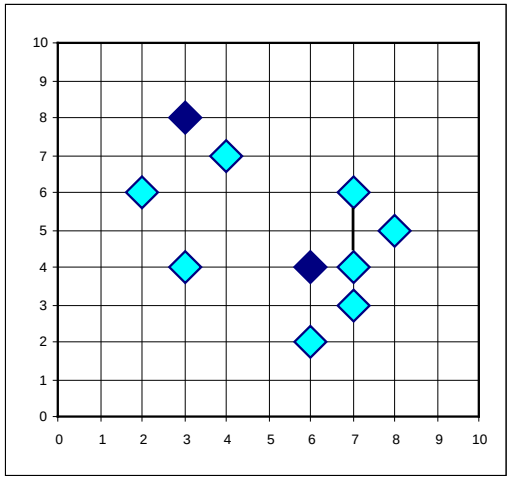
- Clusters are represented by real objects called medoids.
- PAM (Partitioning Around Medoids, Kaufman and Rousseeuw, 1987)
  - starts from an initial set of k medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total clustering cost
- Pseudocode:
  - Select k representative objects arbitrarily
  - Representative replacement: For each pair of non-selected object h and selected object i, check whether i could be replaced by h
  - Replacement is possible if the total clustering cost is improving
  - Repeat until no improvements can be achieved by any replacement

PAM example

K=2

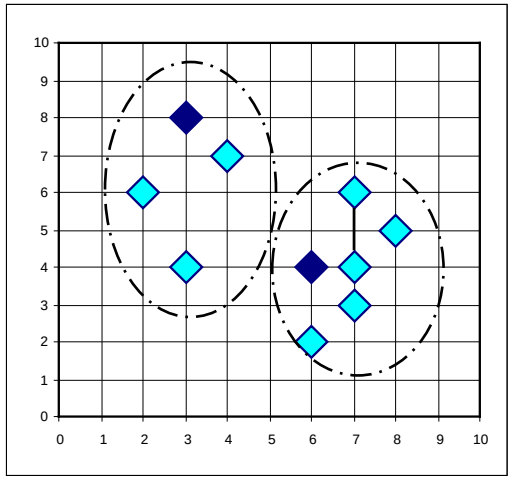


Arbitrary choose k object as initial medoids

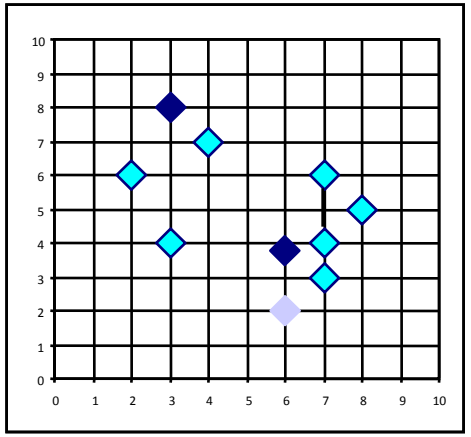


Assign each remaining object to nearest medoids

Total Cost = 20

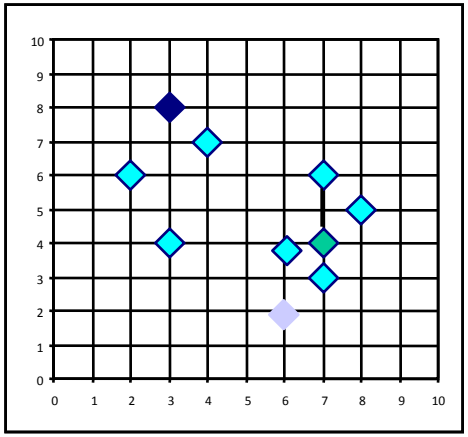


Randomly select a nonmedoid object, O_{ramdom}



Compute total cost of swapping

Total Cost = 26



Swapping O and O_{ramdom}

If quality is improved.

Do loop

Until no change

PAM overview

- Very similar to k-Means
- PAM is more robust than k-Means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean
- PAM works efficiently for small data sets but does not scale well for large data sets.
  - $O(k(n-k)^2)$ for each iteration

  where n is # of data,k is # of clusters

➔ Sampling based method,

CLARA(Clustering LARge Applications)

CLARA (Clustering Large Applications)

- CLARA (Kaufmann and Rousseeuw in 1990)
- It draws multiple samples of the data set, applies PAM on each sample, and gives the best clustering as the output
- Strength: deals with larger data sets than PAM
- Weakness:
  - Efficiency depends on the sample size
  - A good clustering based on samples will not necessarily represent a good clustering of the whole data set if the sample is biased

CLARANS ("Randomized" CLARA) (1994)

- *CLARANS* (A Clustering Algorithm based on Randomized Search), Ng and Han'94
- CLARANS draws sample of neighbors dynamically
- The clustering process can be presented as searching a graph where every node is a potential solution, that is, a set of *k* medoids
- If the local optimum is found, *CLARANS* starts with new randomly selected node in search for a new local optimum
- It is more efficient and scalable than both *PAM* and *CLARA*
- Focusing techniques and spatial access structures may further improve its performance (Ester et al.'95)

What is the right number of clusters I

- The number of clusters k is required as input by the partitioning algorithms
- Silhouette coefficient (Kaufman & Rousseeuw 1990)
  - Let a(o) the distance of an object o to the representative of its cluster and b(o) the distance to the representatives of its "second best" cluster
  - Silhouette *s*(*o*) of an object *o:*

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

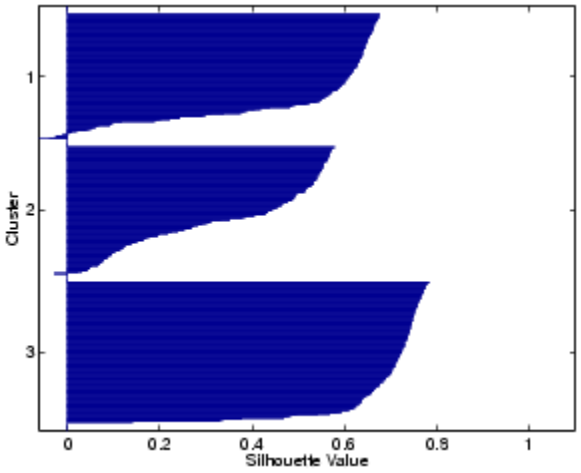
-1 ≤ s(o) ≤ +1

s(o) ~ -1 / 0 / +1 : bad / indifferent / good assignment

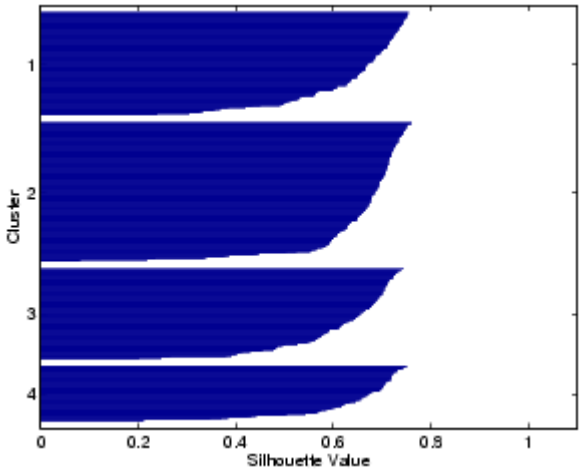
  - s(o)~1→ *a*(*o*) < < *b*(*o*). Small *a*(*o*) means it is well matched to its own cluster. Large b(o) means is badly matched to its neighbouring cluster.
  - s(o)~-1→ the neighbor cluster seems more appropriate
  - s(o)~0→ in the border between two natural clusters

What is the right number of clusters II

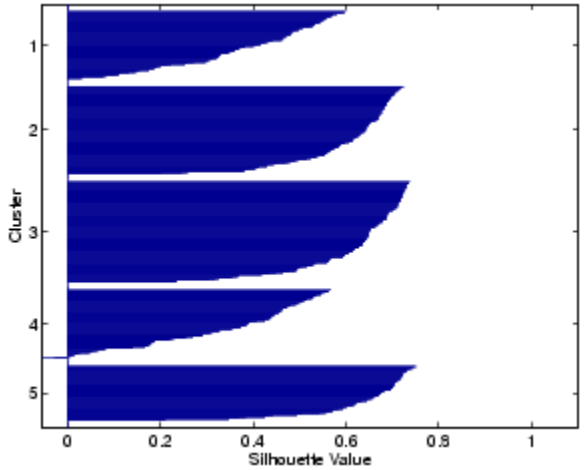
- The Silhouette coefficient of a cluster is the avg silhouette of all its objects
  - Is a measure of how tightly grouped all the data in the cluster are.
  - > 0,7: strong structure, > 0,5: usable structure ….
- The Silhouette coefficient of a clustering is the avg silhouette of all objects
  - is a measure of how appropriately the dataset has been clustered



K=3



K=4



K=5

Outline

- Introduction
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Things you should know
- Homework/tutorial

Things you should know

- What is clustering
- Unsupervised vs supervised
- Dissimilarity measures
- Main cluster descriptors (for numerical data)
- Dissimilarity between clusters
- Main clustering methods
- Partitioning clustering methods
  - k-Means
  - k-Medoids

Homework/ Tutorial

Tutorial: Tutorial this Thursday on classification

Homework:

- Run kMeans clustering in Elki, Weka on a numerical dataset
- Play with the number of clusters k
- Implement your own k-Means or k-Medoids method

Suggested reading:

- Han J., Kamber M., Pei J. *Data Mining: Concepts and Techniques 3rd ed., Morgan Kaufmann, 2011* (Chapter 10)
- Tan P.-N., Steinbach M., Kumar V., *Introduction to Data Mining*, Addison-Wesley, 2006 (Chapter 8).