# Data Mining lecture: Clustering 1

Sebastian Risi

# Chapter 7. Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorisation of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Clustering High-Dimensional Data
7. Summary

# 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
- Unsupervised learning: no predefined classes
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

# Examples of Clustering Applications

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Land use: Identification of areas of similar land use in an earth observation database
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost
- 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
- Games: identify player groups / archetypes

# Quality: What Is Good Clustering?

- A good clustering method will produce high quality clusters with
  - high intra-class similarity
  - low inter-class similarity
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation
- The quality of a clustering method is also measured by 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)
- There is a separate "quality" function that measures the "goodness" of a cluster.
- The definitions of distance functions are usually very different for interval-scaled, boolean, categorical, ordinal ratio, and vector variables.
- Weights should be associated with different variables based on applications and data semantics.
- It is hard to define "similar enough" or "good enough"
  - the answer is typically highly subjective.

# 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

# What we're looking for: dissimilarity

- Many clustering algorithms work exclusively with the dissimilarity between different data points
- We need data structures optimized for this

# Data Structures

- Data matrix

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

- Dissimilarity matrix

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

# Type of data in clustering analysis

- Numerical (interval-scaled) variables
- Binary variables
- Nominal, ordinal, and ratio variables
- Variables of mixed types

# Interval-valued variables

- Interval-scaled variables are continues measurements of roughly linear scale (e.g. weight, height, etc.)

It is very important to normalize data before clustering!

- Calculate the standardized measurement (z-score)

$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

Mean absolute deviation

# Similarity and Dissimilarity Between Objects

- Distances are normally used to measure the similarity or dissimilarity between two data objects
- Some popular ones include: Minkowski distance:

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

  where i = ($x_{i1}$, $x_{i2}$, …, $x_{ip}$) and j = ($x_{j1}$, $x_{j2}$, …, $x_{jp}$) are two p-dimensional data objects, and q is a positive integer

- If q = 1, d is Manhattan distance

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

# Similarity and Dissimilarity Between Objects (Cont.)

- If q = 2, d is Euclidean distance:

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

- Also, one can use weighted distance, parametric Pearson product moment correlation, or other dissimilarity measures

# Binary Variables

- A contingency table for binary data

| | | Object $j$ | | |
|---|---|---|---|---|
| | | 1 | 0 | $sum$ |
| Object $i$ | 1 | $a$ | $b$ | $a+b$ |
| | 0 | $c$ | $d$ | $c+d$ |
| | $sum$ | $a+c$ | $b+d$ | $p$ |

- Distance measure (**Hamming**) for symmetric binary variables: *simply count the differences*

$$d(i,j) = \frac{b+c}{a+b+c+d}$$

- Distance measure for asymmetric binary variables:

$$d(i,j) = \frac{b+c}{a+b+c}$$

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

$$sim_{Jaccard}(i,j) = \frac{a}{a+b+c}$$

# Nominal Variables

- A generalization of the binary variable in that it can take more than 2 states, e.g., red, yellow, blue, green
- 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 variables
  - creating a new binary variable for each of the M nominal states

# Variables of Mixed Types

- A database may contain all the six types of variables
  - symmetric binary, asymmetric binary, nominal, ordinal, interval and ratio
- One may use a weighted formula to combine their effects

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

  - f is binary or nominal:

    $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$ , or $d_{ij}^{(f)} = 1$ otherwise
  - f is interval-based: use the normalized distance
  - f is ordinal or ratio-scaled
    - compute ranks $r_{if}$ and
    - and treat $z_{if}$ as interval-scaled

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

# 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, ROCK, CAMELEON
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSACN, OPTICS, DenClue

# Major Clustering Approaches (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

# Partitioning Algorithms: Basic Concept

- Partitioning method: Construct a partition of a database **D** of **n** objects into a set of **k** clusters, s.t., min sum of squared distance

$$\Sigma_{m=1}^{k}\Sigma_{t_{mi}\in Km}(C_m - t_{mi})^2$$

- Given a k, find a partition of k clusters that optimizes the chosen partitioning criterion
- Which is the simplest possible clustering algorithm?

# Partitioning Algorithms

- 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

# 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

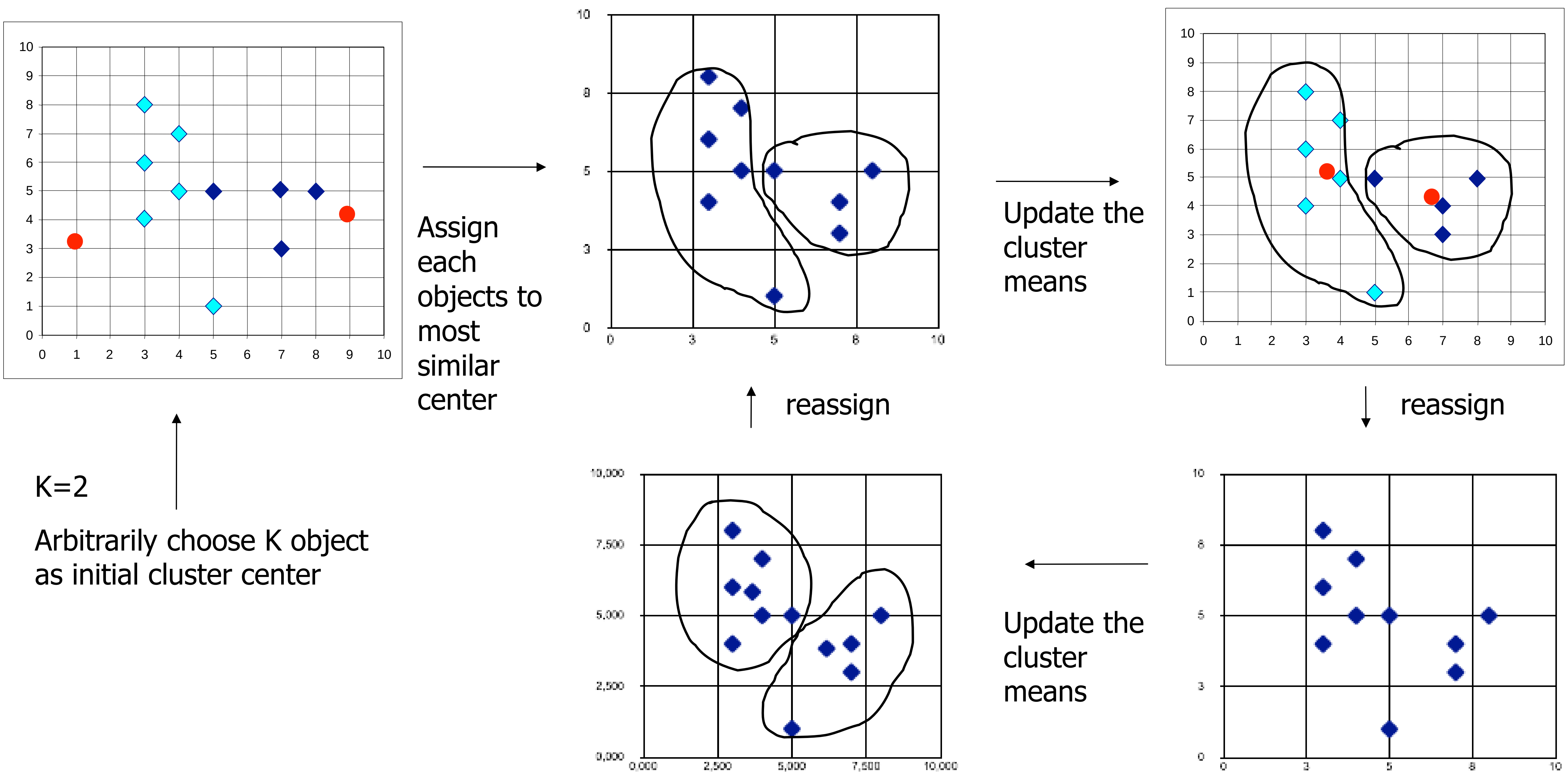
$$D_m = \sqrt{\frac{\sum_{i=1}^{N} \sum_{i=1}^{N} (t_{ip} - t_{iq})^2}{N(N-1)}}$$

# The K-Means Clustering Method

- Given k, the k-means algorithm is implemented in four steps:
  - Partition objects into k nonempty subsets
  - Compute seed points as the centroids of the clusters of the current partition (the centroid is the center, i.e., mean point, of the cluster)
  - Assign each object to the cluster with the nearest seed point
  - Go back to Step 2, stop when no more new assignment

# The K-Means Clustering Method

- Example



# Exercise

Use the k-means algorithm and Euclidean distance to cluster the following 8 examples into 3 clusters (perform 1st iteration): A1=(2,10), A2=(2,5), A3=(8,4), A4=(5,8), A5=(7,5), A6=(6,4), A7=(1,2), A8=(4,9).

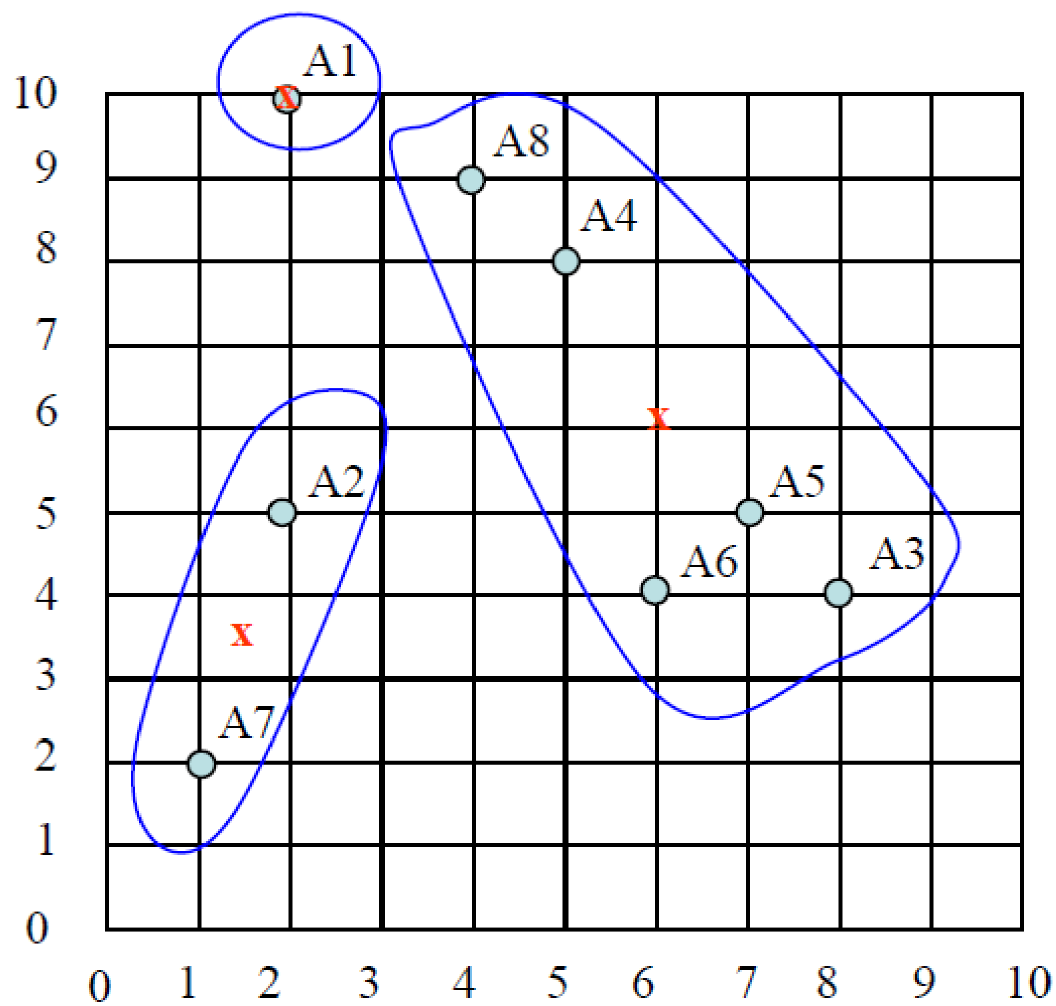
| | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 |
|---|---|---|---|---|---|---|---|---|
| A1 | 0 | $\sqrt{25}$ | $\sqrt{36}$ | $\sqrt{13}$ | $\sqrt{50}$ | $\sqrt{52}$ | $\sqrt{65}$ | $\sqrt{5}$ |
| A2 | | 0 | $\sqrt{37}$ | $\sqrt{18}$ | $\sqrt{25}$ | $\sqrt{17}$ | $\sqrt{10}$ | $\sqrt{20}$ |
| A3 | | | 0 | $\sqrt{25}$ | $\sqrt{2}$ | $\sqrt{2}$ | $\sqrt{53}$ | $\sqrt{41}$ |
| A4 | | | | 0 | $\sqrt{13}$ | $\sqrt{17}$ | $\sqrt{52}$ | $\sqrt{2}$ |
| A5 | | | | | 0 | $\sqrt{2}$ | $\sqrt{45}$ | $\sqrt{25}$ |
| A6 | | | | | | 0 | $\sqrt{29}$ | $\sqrt{29}$ |
| A7 | | | | | | | 0 | $\sqrt{58}$ |
| A8 | | | | | | | | 0 |

Initial seeds (centers of each cluster) are A1, A4 and A7.

http://webdocs.cs.ualberta.ca/~zaiane/courses/cmput695/F07/exercises/Exercises695Clus-solution.pdf

# Results

- After one iteration: 1: {A1}, 2: {A3, A4, A5, A6, A8}, 3: {A2, A7}
- centers of the new clusters: C1= (2, 10), C2= ((8+5+7+6+4)/5, (4+8+5+4+9)/5) = (6, 6), C3= ((2+1)/2, (5+2)/2) = (1.5, 3.5)



# Comments on the K-Means Method

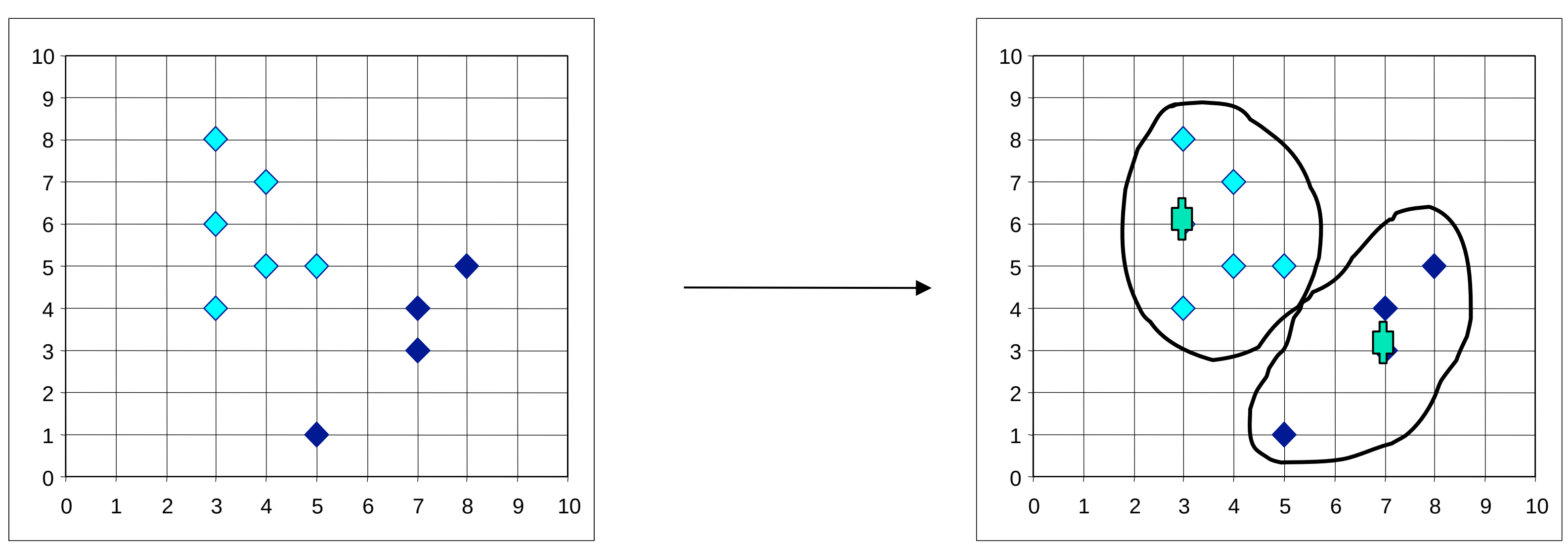
- Strength: Relatively efficient: $O(tkn)$, where $n$ is # objects, $k$ is # clusters, and $t$ is # iterations. Normally, $k, t \ll n$.
    - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$
- Comment: Often terminates at a local optimum. The global optimum may be found using techniques such as: deterministic annealing and genetic algorithms
- Weakness
    - Applicable only when mean is defined, then what about categorical data?
    - 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

# Variations of the K-Means Method

- A few 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 (Huang'98)
  - Replacing means of clusters with <u>modes</u>
  - Using new dissimilarity measures to deal with categorical objects
  - Using a <u>frequency</u>-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.

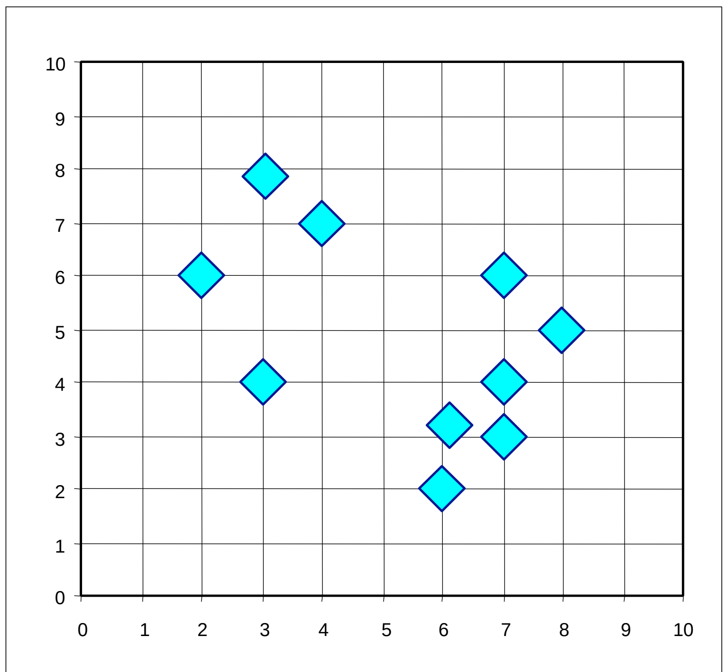


# The K-Medoids Clustering Method

- Find representative objects, called medoids, in clusters
- PAM (Partitioning Around Medoids, 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 distance of the resulting clustering
  - PAM works effectively for small data sets, but does not scale well for large data sets
- CLARA (Kaufmann & Rousseeuw, 1990)
- CLARANS (Ng & Han, 1994): Randomized sampling
- Focusing + spatial data structure (Ester et al., 1995)

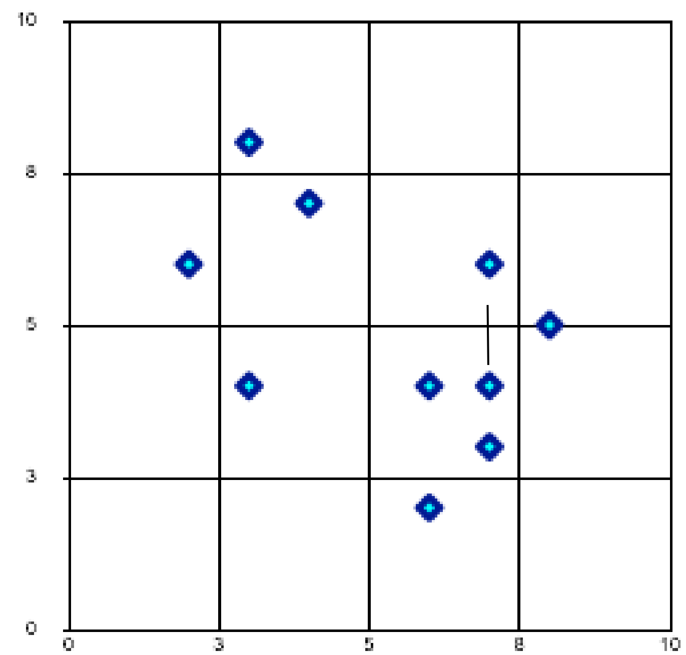
# A Typical K-Medoids Algorithm (PAM)

Total Cost = 20

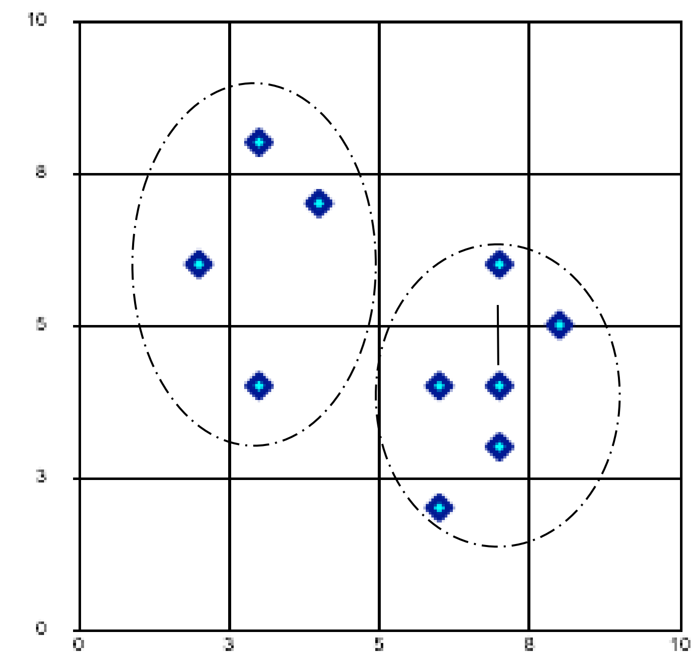


K=2

Arbitrary choose k object as initial medoids



Assign each remaining object to nearest medoids



Randomly select a nonmedoid object,$O_{ramdom}$

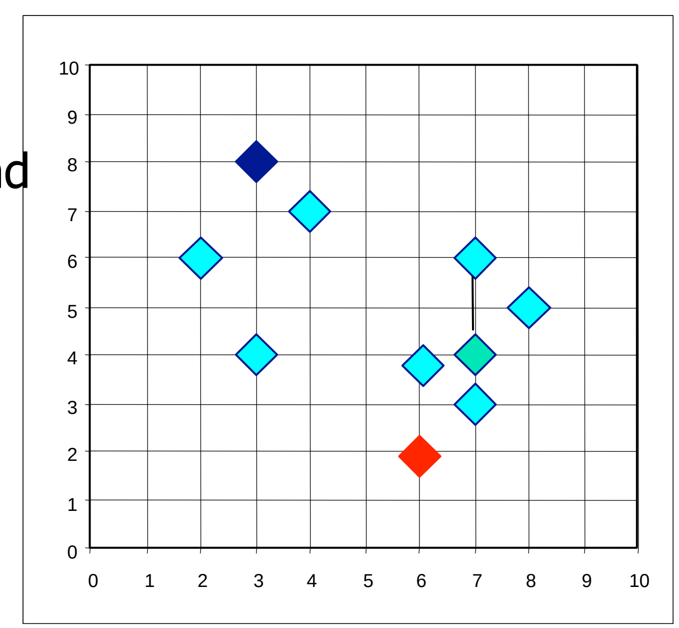
Total Cost = 26

**Do loop**

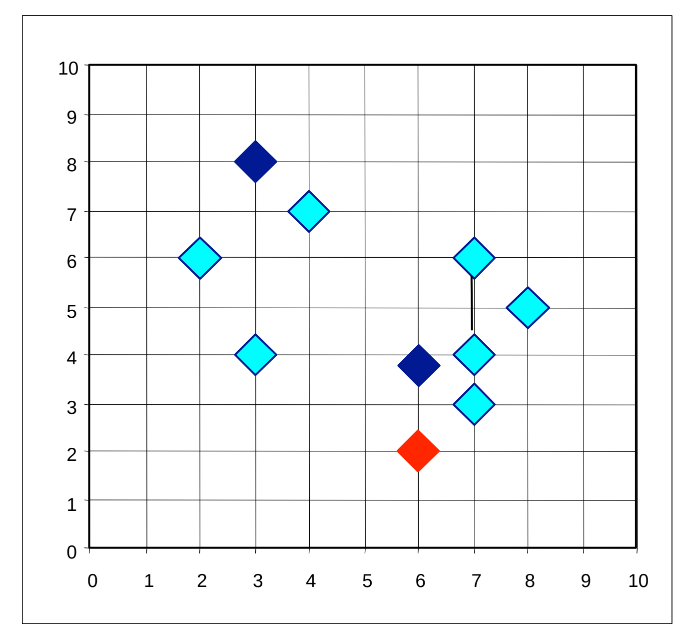
**Until no change**

Swapping O and $O_{ramdom}$

If quality is improved.



Compute total cost of swapping



# PAM (Partitioning Around Medoids) (1987)

1. Initialize: randomly select *k* of the *n* data points as the medoids
2. Associate each data point to the closest medoid
3. For each medoid *m*
   For each non-medoid data point *o*
   Swap *m* and *o* and compute the total cost of the configuration
4. Select the configuration with the lowest cost.
4. Repeat steps 2 to 4 until there is no change in the medoid.

Sergios Theodoridis & Konstantinos Koutroumbas (2006). *Pattern Recognition 3rd ed*. p. 635.

# What Is the Problem with PAM?

- 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

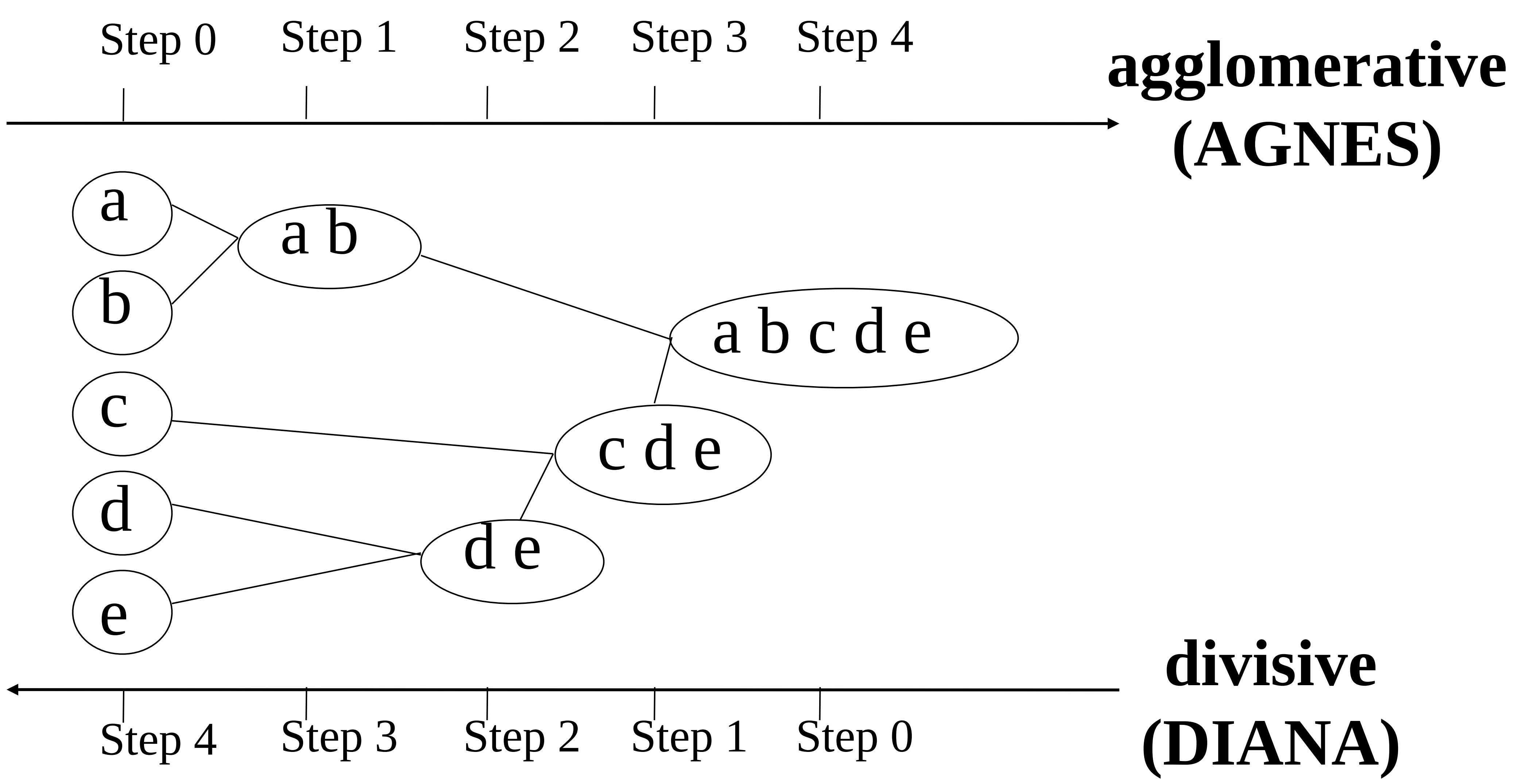
    where n is # of data,k is # of clusters

➔Sampling based method,

CLARA(Clustering LARge Applications)

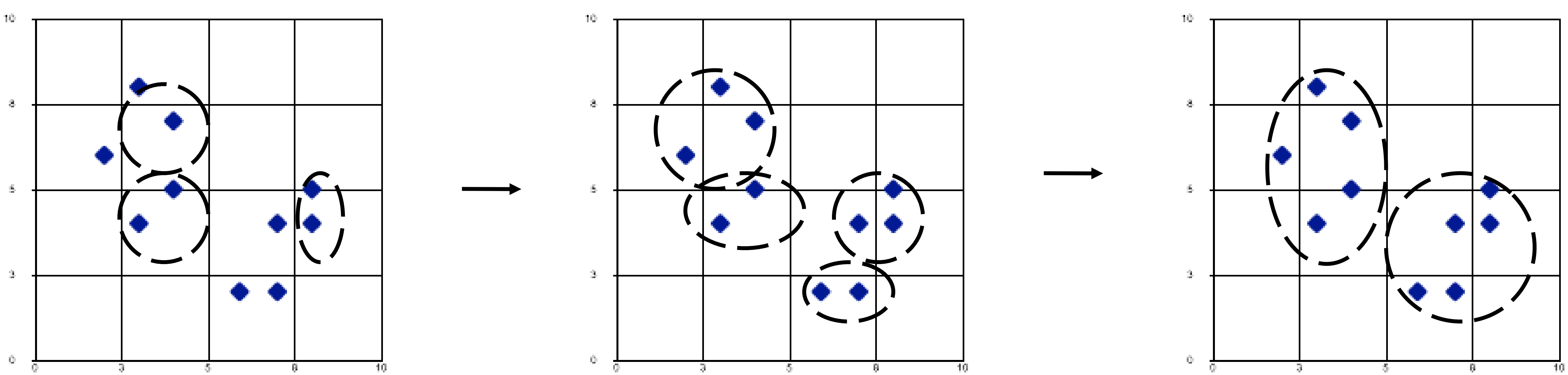
# Hierarchical Clustering

- 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 analysis 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



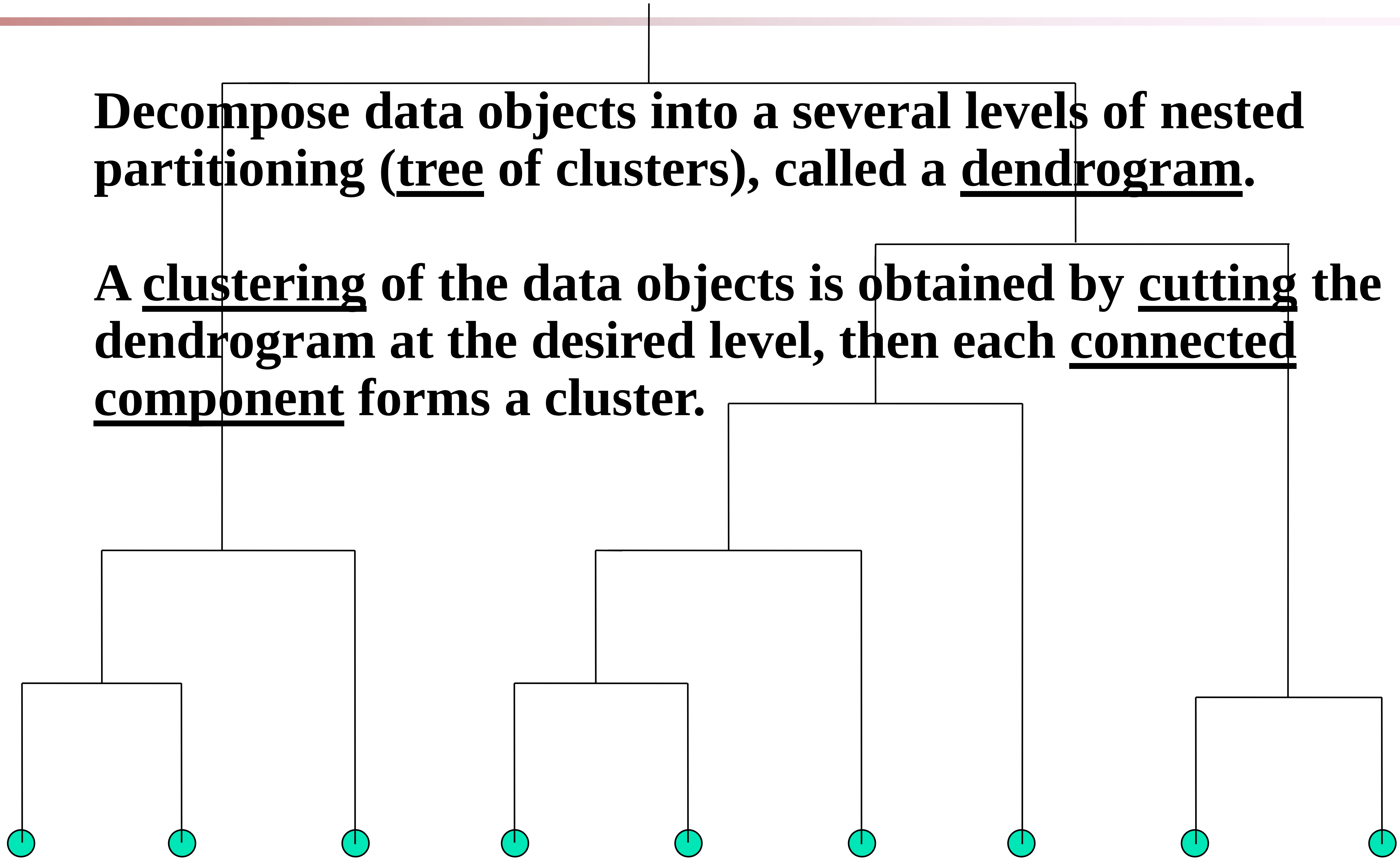
# Typical Alternatives to Calculate the Distance between Clusters

- Single link: smallest distance between an element in one cluster and an element in the other, i.e., $dis(K_i, K_j) = min(t_{ip}, t_{jq})$
- Complete link: largest distance between an element in one cluster and an element in the other, i.e., $dis(K_i, K_j) = max(t_{ip}, t_{jq})$
- Average: avg distance between an element in one cluster and an element in the other, i.e., $dis(K_i, K_j) = avg(t_{ip}, t_{jq})$
- Centroid: distance between the centroids of two clusters, i.e., $dis(K_i, K_j) = dis(C_i, C_j)$
- 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

# *Dendrogram:* Shows How the Clusters are Merged

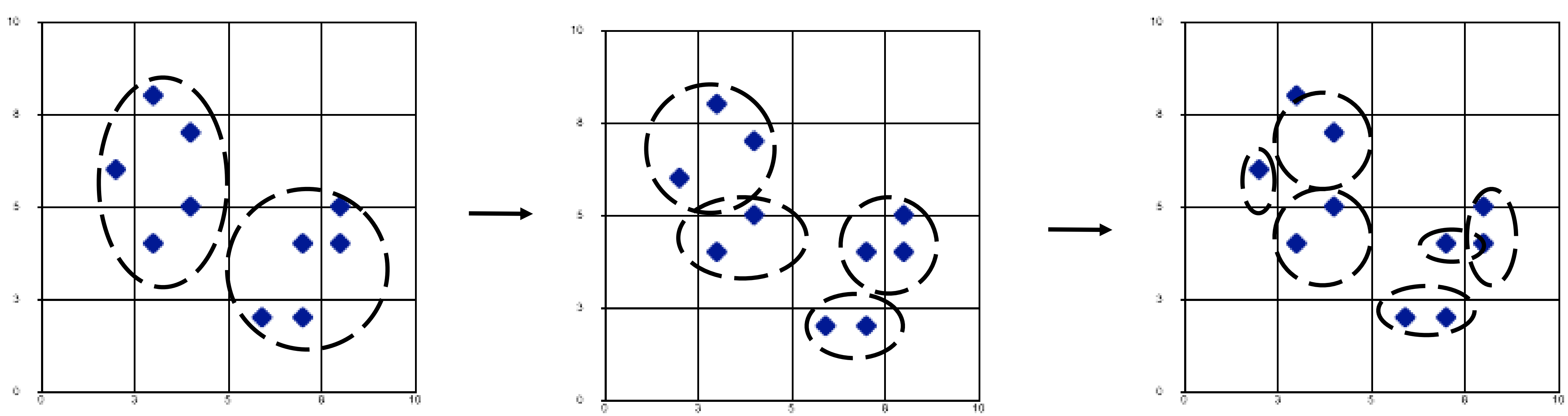
Decompose data objects into a several levels of nested partitioning (<u>tree</u> of clusters), called a <u>dendrogram</u>.

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



# DIANA (Divisive Analysis)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in several statistical analysis packages
- Inverse order of AGNES
- Eventually each node forms a cluster on its own



# Recent Hierarchical Clustering Methods

- Major weakness of agglomerative clustering methods
  - do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects
  - can never undo what was done previously
- Integration of hierarchical with distance-based clustering
  - BIRCH (1996): uses *clustering feature tree* (CF-tree) and incrementally adjusts the quality of sub-clusters
  - ROCK (1999): clustering categorical data by neighbor and link analysis (the number of common neighbors between two objects)
  - CHAMELEON (1999): hierarchical clustering using dynamic modeling

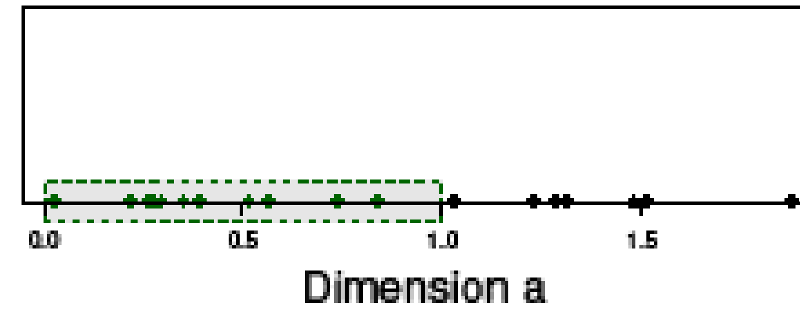
# Clustering High-Dimensional Data

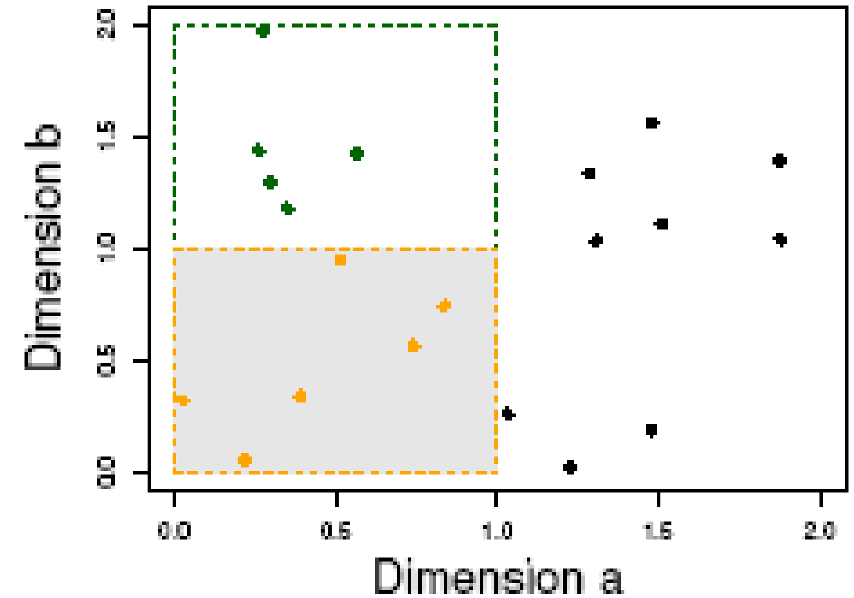
- Clustering high-dimensional data
  - Many applications: text documents, DNA micro-array data
  - Major challenges:
    - Many irrelevant dimensions may mask clusters
    - Distance measure becomes meaningless—due to equi-distance
    - Clusters may exist only in some subspaces
- Methods
  - Feature transformation: only effective if most dimensions are relevant
    - PCA & SVD useful only when features are highly correlated/redundant
  - Feature selection: wrapper or filter approaches
    - useful to find a subspace where the data have nice clusters
  - Subspace-clustering: find clusters in all the possible subspaces
    - CLIQUE, ProClus, and frequent pattern-based clustering

# The Curse of Dimensionality

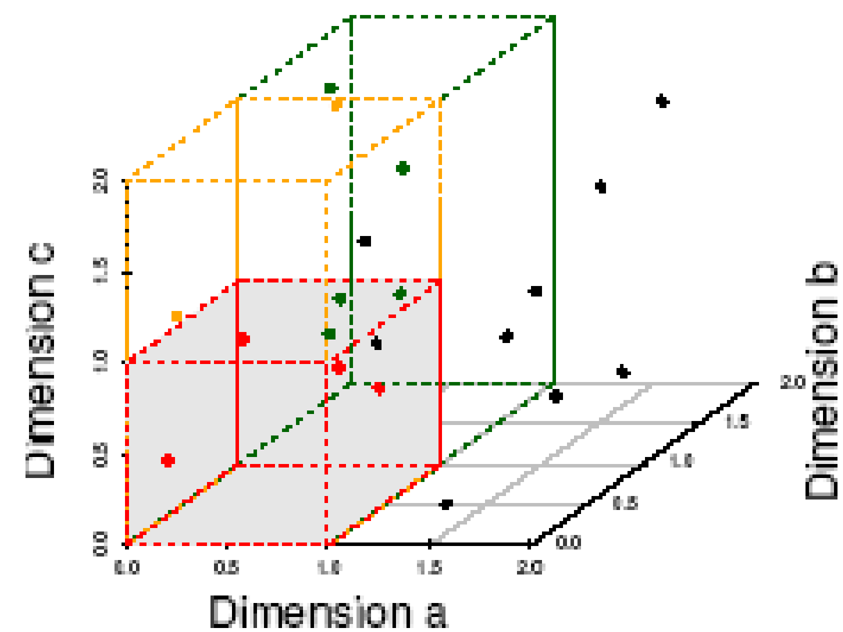
(graphs adapted from Parsons et al. KDD Explorations 2004)

- Data in only one dimension is relatively packed
- Adding a dimension “stretches” the points across that dimension, making them further apart
- Adding more dimensions will make the points further apart—high dimensional data is extremely sparse
- Distance measure becomes meaningless—due to equi-distance




(b) 6 Objects in One Unit Bin

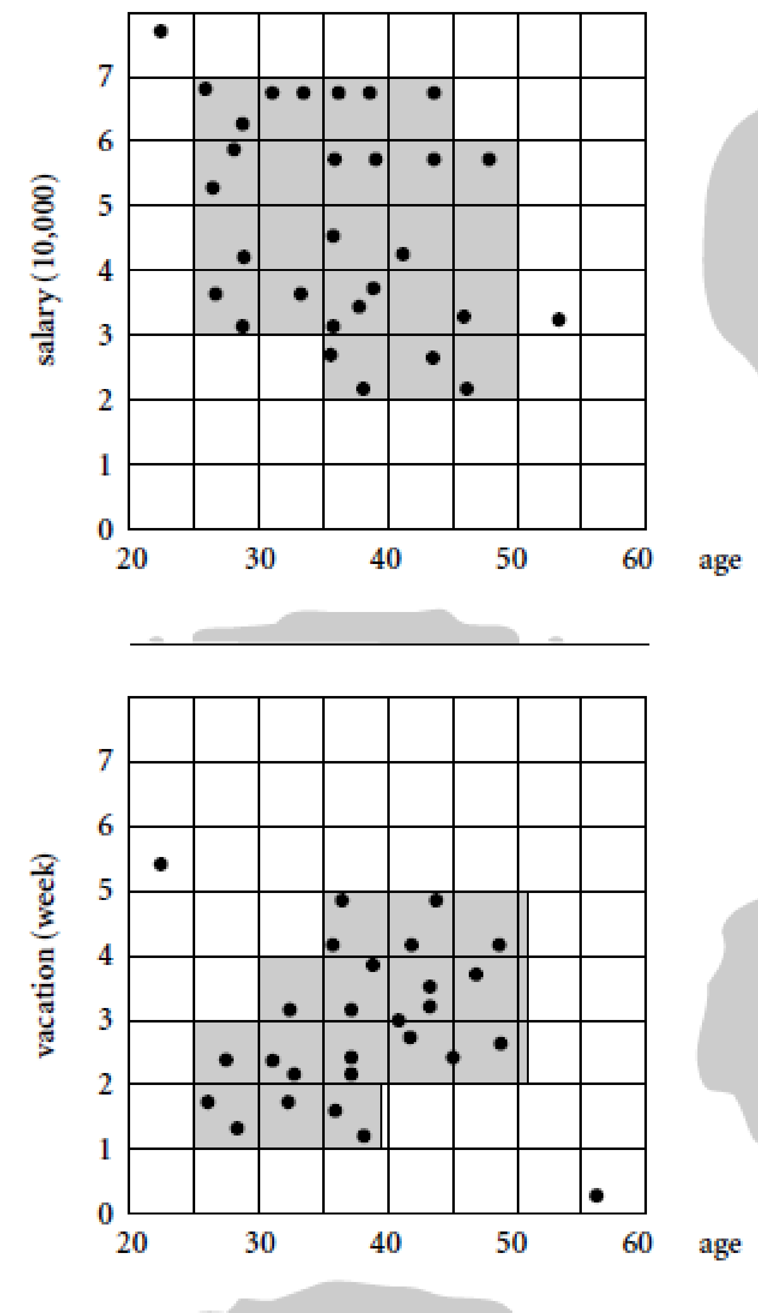



(c) 4 Objects in One Unit Bin

# Frequent Pattern-Based Approach

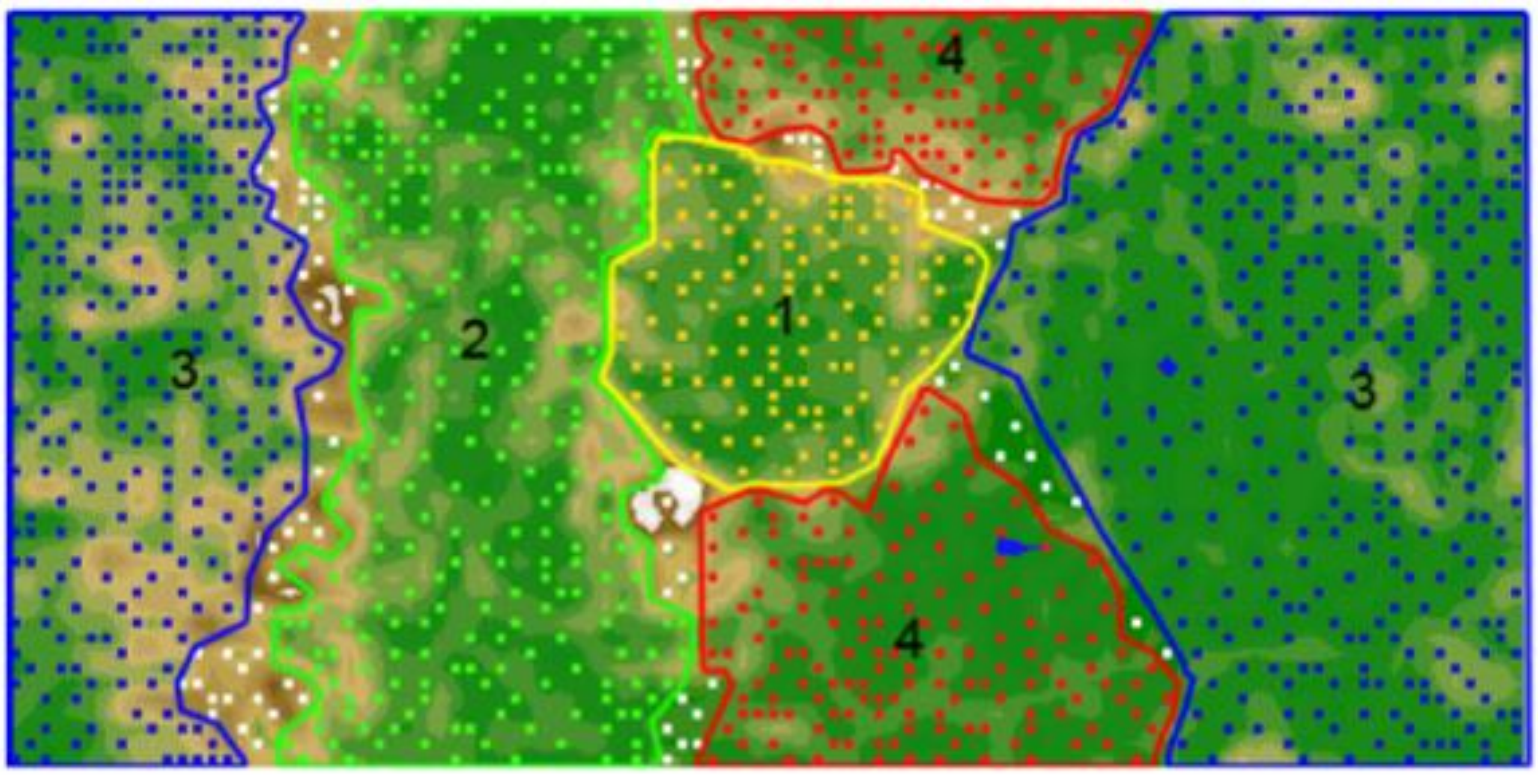
- Clustering high-dimensional space (e.g., clustering text documents, microarray data)
  - Projected subspace-clustering: which dimensions to be projected on?
    - CLIQUE, ProClus
  - Feature extraction: costly and may not be effective?
  - Using frequent patterns as "features"
    - "Frequent" are inherent features
    - Mining freq. patterns may not be so expensive
- Typical methods
  - Frequent-term-based document clustering
  - Clustering by pattern similarity in micro-array data (pClustering)

CLIQUE



# Cluster analysis example

Player modelling using self-organisation in *Tomb Raider: Underworld,* Drachen, Canossa & Yannakakis, CIG 2009




http://www.youtube.com/watch?v=HJS-SxgXAI4!

# Cluster analysis example

*Cluster number 1 corresponds to players that*

- *die very few times;*
- *their death is caused mainly by the environment*
- *and they complete TRU very fast.*
- *These players' HOD requests vary from low to average*

*and they are labeled as* ***Veterans***

*as they are the most well performing group of players despite the high number of environment-related deaths.*

# Cluster analysis example

*Likewise, cluster number 2 corresponds to players that*

- *die quite often mainly due to falling;*
- *it takes them quite a long time to complete the game;*
- *And they do not appear to ask for puzzle hints or answers.*

*Players of this cluster are labeled as* ***Solvers****, because they are adept at solving the puzzles of TRU.*

*Their long completion times, low number of deaths by enemies or environment effects indicate a slow-moving, careful style of play with the number one cause of death being falling (jumping).*

# 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
- There are still lots of research issues on cluster analysis

# Lab

- Implement k-means