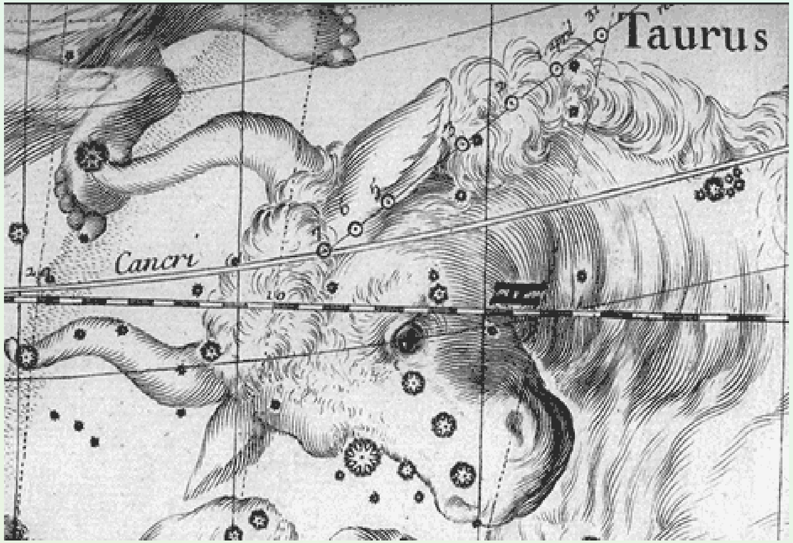
Cluster Analysis

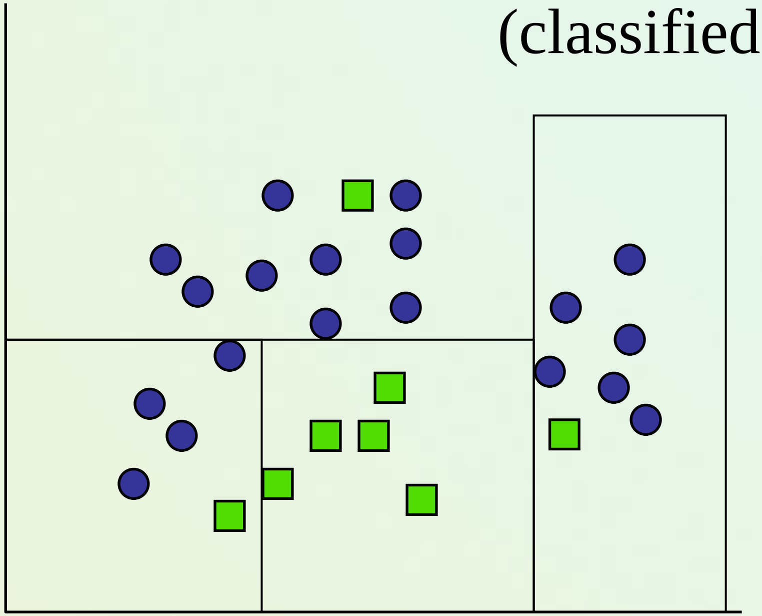
Astronomy - aggregation of stars, galaxies, or super galaxies, …



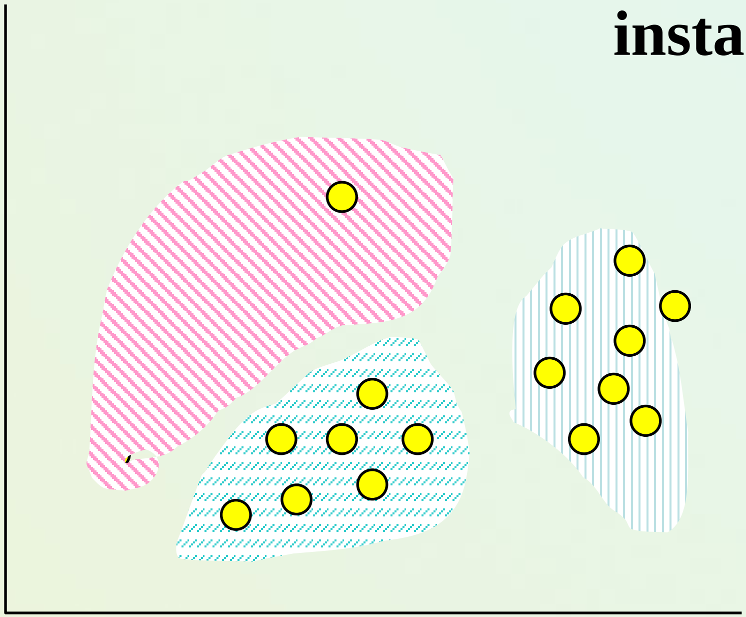
Classification vs. Clustering

Classification: Supervised learning:

Learns a method for predicting the instance class from pre-labeled (classified) instances



Clustering



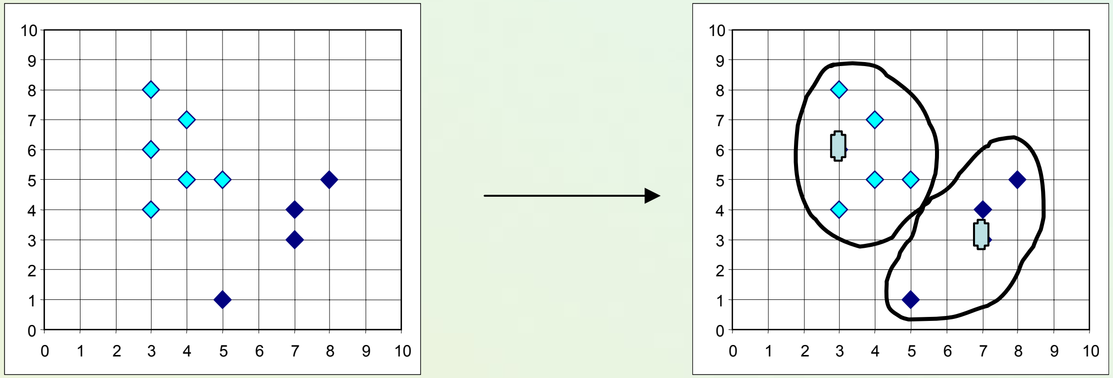
Unsupervised learning:

Finds “natural” grouping of instances given un-labeled data

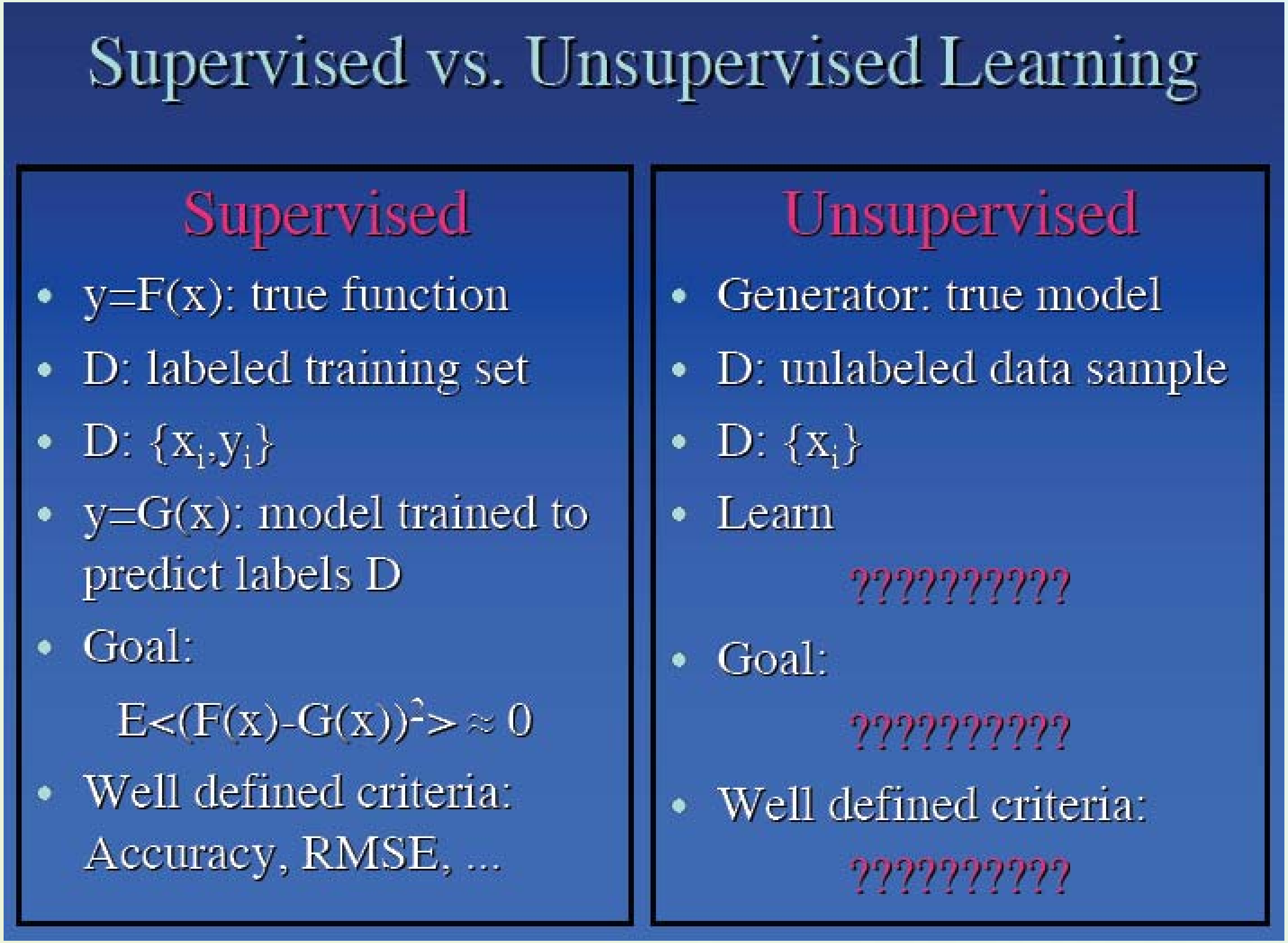
Problem Statement

Given a set of records (instances, examples, objects, observations, …), organize them into clusters (groups, classes)

- Clustering: the process of grouping physical or abstract objects into classes of similar objects



Supervised classification vs. clustering



Supervised vs. Unsupervised Learning

Supervised

- $y=F(x)$: true function
- D: labeled training set
- D: $\{x_i,y_i\}$
- $y=G(x)$: model trained to predict labels D
- Goal:

  $E<(F(x)-G(x))^2> \approx 0$
- Well defined criteria: Accuracy, RMSE, ...

Unsupervised

- Generator: true model
- D: unlabeled data sample
- D: $\{x_i\}$
- Learn

  ??????????
- Goal:

  ??????????
- Well defined criteria:

  ??????????

What is a cluster?

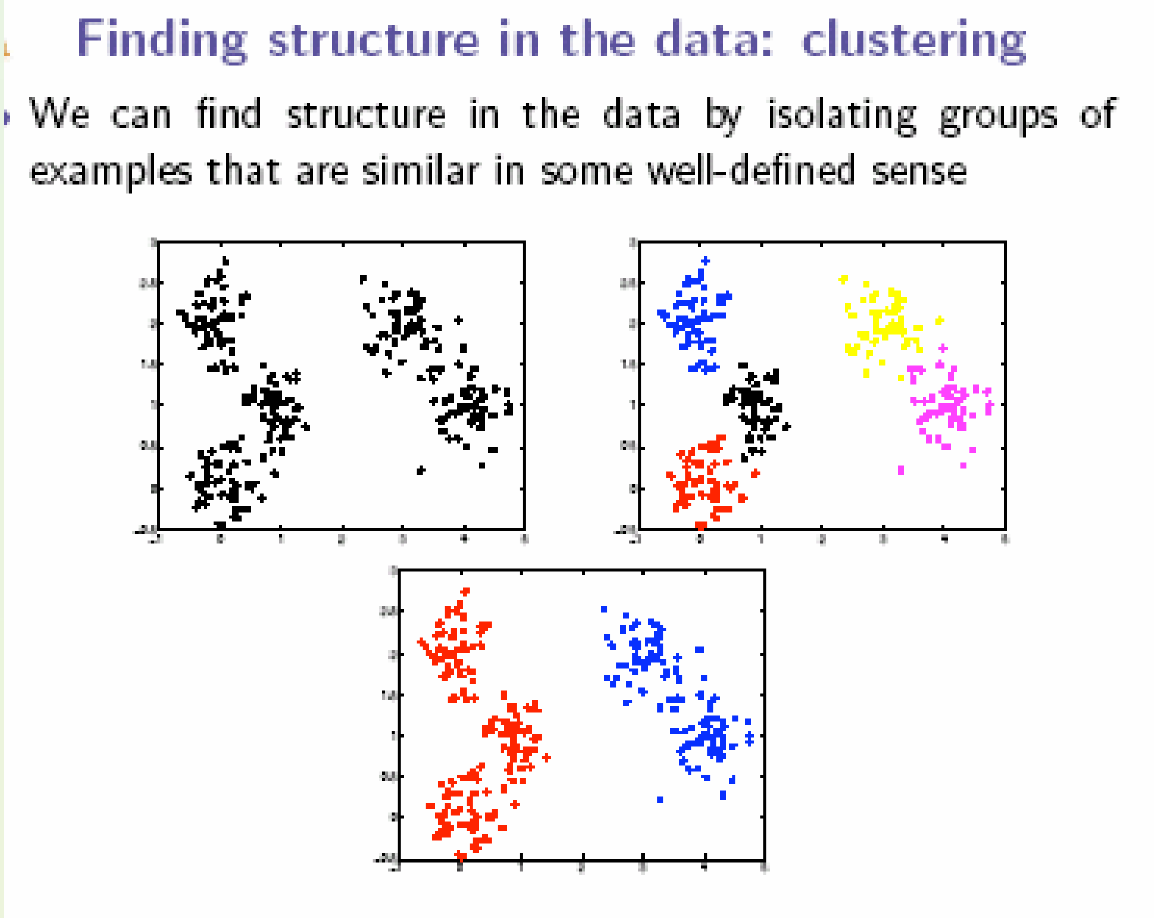
1. A cluster is a subset of objects which are “similar”
2. A subset of objects such that the distance between any two objects in the cluster is less than the distance between any object in the cluster and any object not located inside it.
3. A connected region of a multidimensional space containing a relatively high density of objects.

What Is Clustering ?

- Clustering is a process of partitioning a set of data (or objects) into a set of meaningful sub-classes, called clusters.
    - Help users understand the natural grouping or structure in a data set.
- Clustering: unsupervised classification: no predefined classes.
- Used either as a stand-alone tool to get insight into data distribution or as a preprocessing step for other algorithms.
    - Moreover, data compression, outliers detection, understand human concept formation.

Looking for „comprehensible structures” in data

- Help users to find and try to understand„sth ” in data



Finding structure in the data: clustering

- We can find structure in the data by isolating groups of examples that are similar in some well-defined sense

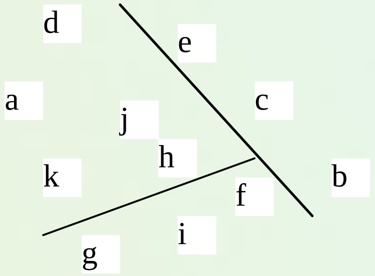
- Still many possible results

What Is Good Clustering?

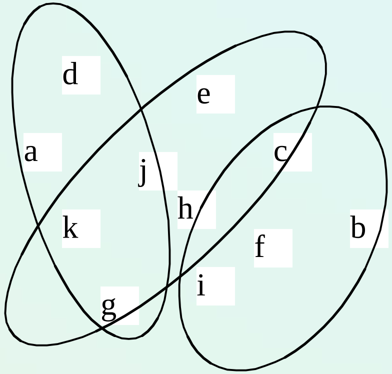
- A good clustering method will produce high quality clusters in which:
  - the intra-class (that is, intra-cluster) similarity is high.
  - the inter-class similarity is low.
- The quality of a clustering result also depends on both the similarity measure used by the method and its implementation.
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns.
- However, objective evaluation is problematic: usually done by human / expert inspection.

Different ways of representing clusters

(a)



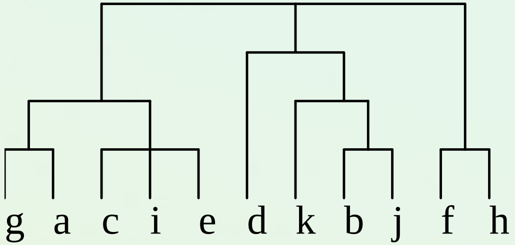
(b)



(c)

	1	2	3
a	0.4	0.1	0.5
b	0.1	0.8	0.1
c	0.3	0.3	0.4
d	0.1	0.1	0.8
e	0.4	0.2	0.4
f	0.1	0.4	0.5
g	0.7	0.2	0.1
h	0.5	0.4	0.1
…			

(d)



Applications of Clustering

Clustering has wide applications in

- Economic Science (especially market research).
- WWW:
  - Document classification
  - Cluster Weblog data to discover groups of similar access patterns
- Pattern Recognition.
- Spatial Data Analysis:
  - create thematic maps in GIS by clustering feature spaces
- Image Processing

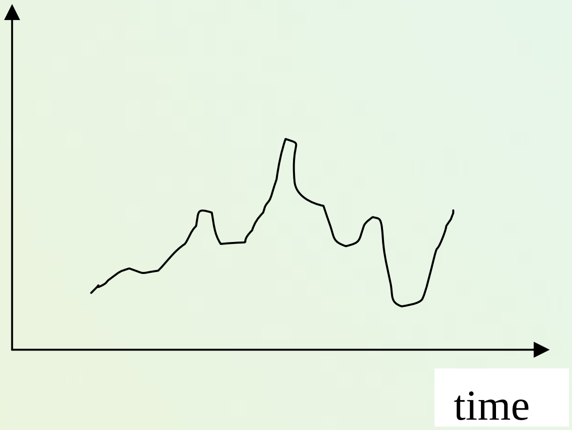
Examples of Clustering Applications

- 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.
- and many others,…

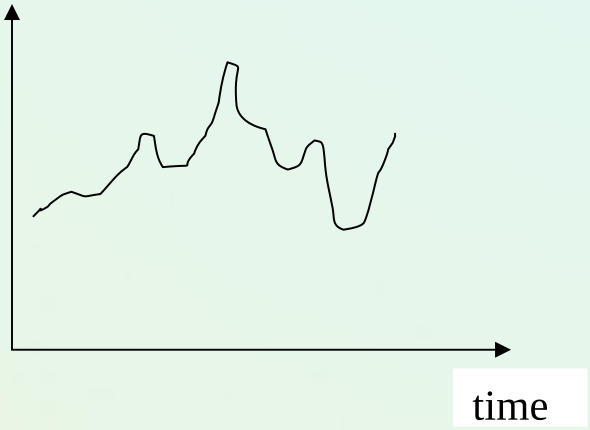
Time-Series Similarities – specific data mining

Given a database of time-series.

Group "similar" time-series



Investor Fund A



Investor Fund B

Clustering Methods

- Many different method and algorithms:
  - For numeric and/or symbolic data
  - Exclusive vs. overlapping
    - Crisp vs. soft computing paradigms
  - Hierarchical vs. flat (non-hierarchical)
  - Access to all data or incremental learning
  - Semi-supervised mode
- Algorithms also vary by:
  - Measures of similarity
  - Linkage methods
  - Computational efficiency

Data Structures

- Data matrix

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

- Dis/similarity matrix

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

Measuring Dissimilarity or Similarity in Clustering

- Dissimilarity/Similarity metric: Similarity is expressed in terms of a distance function, which is typically metric:

  $d(i, j)$
- There are also used in "quality" functions, which estimate the "goodness" of a cluster.
- The definitions of distance functions are usually very different for interval-scaled, boolean, categorical, ordinal and ratio variables.
- Weights should be associated with different variables based on applications and data semantics.

Type of attributes in clustering analysis

- Interval-scaled variables
- Binary variables
- Nominal, ordinal, and ratio variables
- Variables of mixed types

    - Remark: variable vs. attribute

Distance Measures

To discuss whether a set of points is close enough to be considered a cluster, we need a distance measure - D(x, y)

The usual axioms for a distance measure D are:

- $D(x, x) = 0$
- $D(x, y) = D(y, x)$
- $D(x, y) \leq D(x, z) + D(z, y)$ the triangle inequality

Distance Measures (2)

Assume a k-dimensional Euclidean space, the distance between two points, x=[x_1, x_2, ..., x_k] and y=[y_1, y_2, ..., y_k] may be defined using one of the measures:

- Euclidean distance: ("L_2 norm") $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$
- Manhattan distance: ("L_1 norm") $\sum_{i=1}^{k} | x_i - y_i |$
- Max of dimensions: ("L_∞ norm") $\max_{i=1}^{k} | x_i - y_i |$

Distance Measures (3)

- Minkowski distance: $$(\sum_{i=1}^{k}(|x_i - y_i|)^q)^{1/q}$$

When there is no Euclidean space in which to place the points, clustering becomes more difficult: Web page accesses, DNA sequences, customer sequences, categorical attributes, documents, etc.

Standarization / Normalization

- If the values of attributes are in different units then it is likely that some of them will take vary large values, and hence the "distance" between two cases, on this variable, can be a big number.
- Other attributes may be small in values, or not vary much between cases, in which case the difference between the two cases will be small.
- The attributes with high variability / range will dominate the metric.
- Overcome this by standardization or normalization

$$z_i = \frac{x_i - \overline{x}_i}{s_{x_i}}$$

Binary variables

- A contingency table for binary data

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

- Simple matching coefficient (invariant, if the binary variable is <u>*symmetric*</u>): $d(i, j) = \dfrac{b + c}{a + b + c + d}$

- Jaccard coefficient (noninvariant if the binary variable is <u>*asymmetric*</u>): $d(i, j) = \dfrac{b + c}{a + b + c}$

Nominal, ordinal and ratio variables

- nominal variables: > 2 states, e.g., red, yellow, blue, green.
  - *Simple matching*: *u*: # of matches, *p*: total # of variables.

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

  - Also, one can use a large number of binary variables.
- ordinal variables: order is important, e.g., rank.
  - Can be treated like interval-scaled, by replacing x_{if} by their rank $r_{if} \in \{1,..., M_f\}$ and replacing *i*-th object in the *f*-th variable by

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

- ratio variables: a positive measurement on a nonlinear scale, approximately at exponential scale, such as Ae^{Bt} or Ae^{-Bt}
  - One may treat them as continuous ordinal data or perform logarithmic transformation and then treat them as interval-scaled.

Variables of mixed types

- Data sets may contain all types of attrib./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, o.w. $d_{ij}^{(f)} = 1$
  - *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 $z_{if} = \frac{r_{if} - 1}{M_f - 1}$

Main Categories of Clustering Methods

- Partitioning algorithms: Construct various partitions and then evaluate them by some criterion.
- Hierarchy algorithms: Create a hierarchical decomposition of the set of data (or objects) using some criterion.
- Density-based: based on connectivity and density functions
- Grid-based: based on a multiple-level granularity structure
- Model-based: A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other.

Partitioning Algorithms: Basic Concept

- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion.
  - Global optimal: exhaustively enumerate all partitions.
  - Heuristic methods: *k-means* and *k-medoids* algorithms.
  - *k-means* (MacQueen'67): 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.

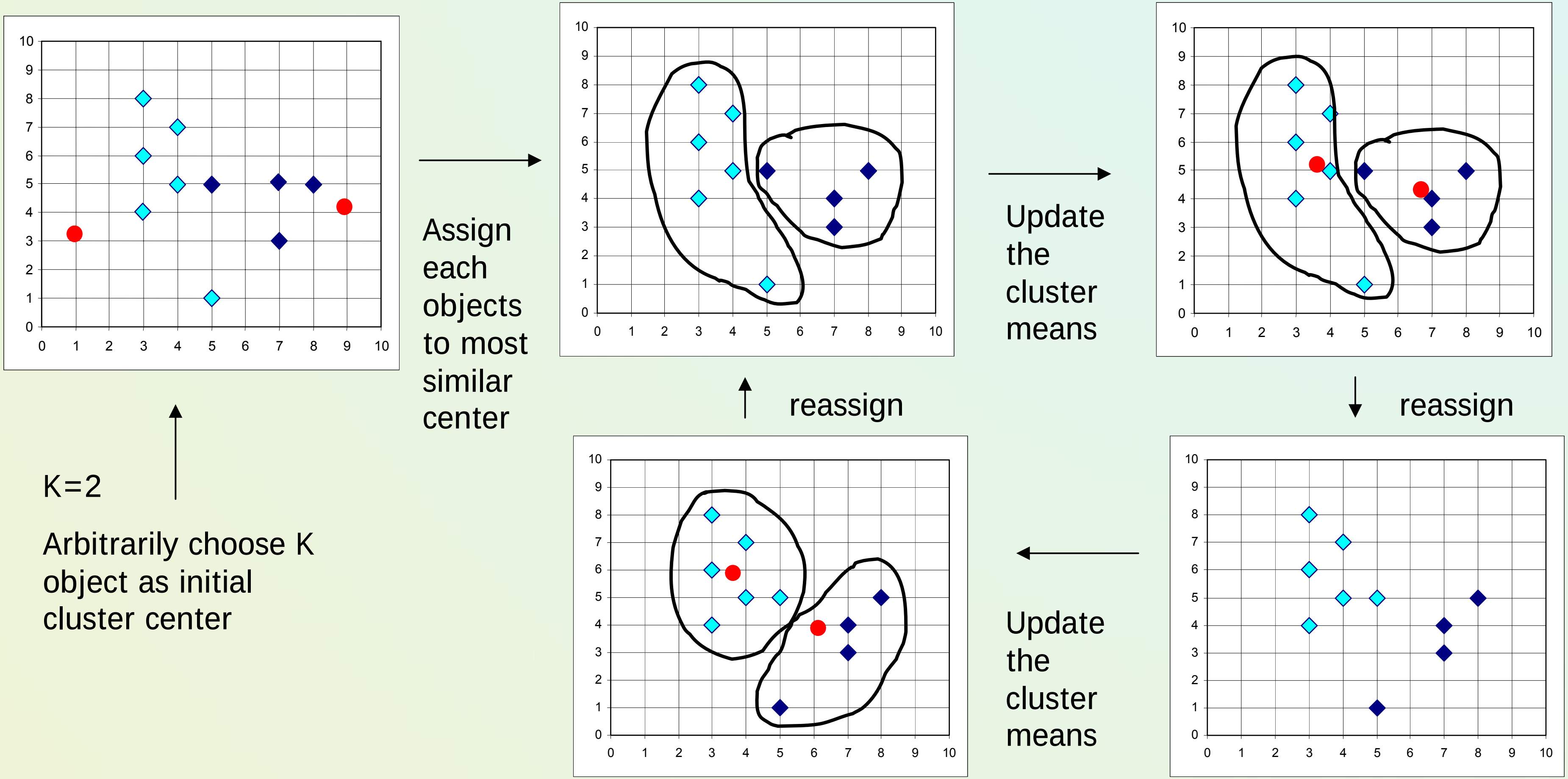
Simple Clustering: K-means

Basic version works with numeric data only

1) Pick a number (K) of cluster centers - *centroids* (at random)
2) Assign every item to its nearest cluster center (e.g. using Euclidean distance)
3) Move each cluster center to the mean of its assigned items
4) Repeat steps 2,3 until convergence (change in cluster assignments less than a threshold)

Illustrating *K-Means*

- Example



K=2

Arbitrarily choose K object as initial cluster center

Assign each objects to most similar center

Update the cluster means

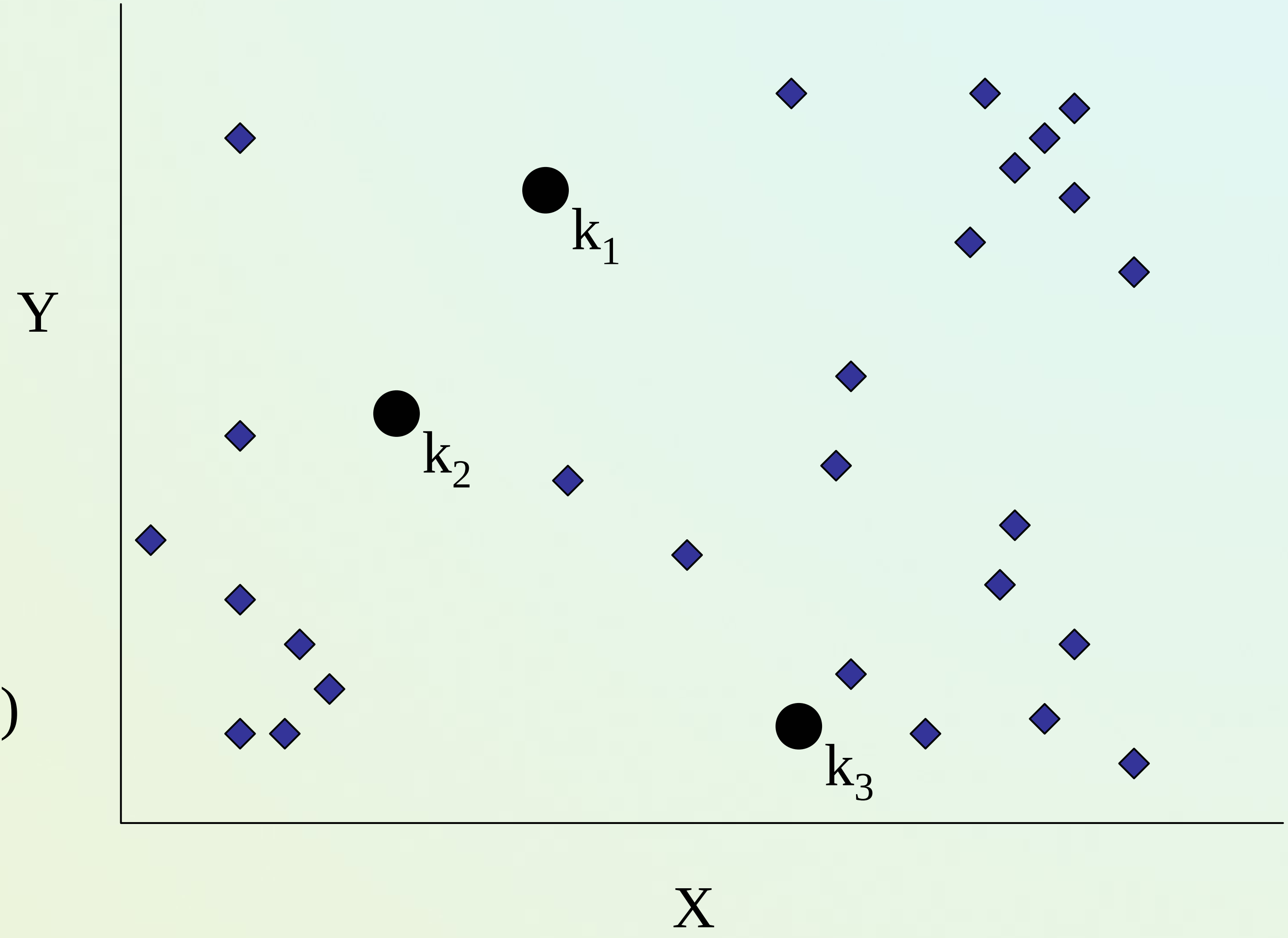
reassign

Update the cluster means

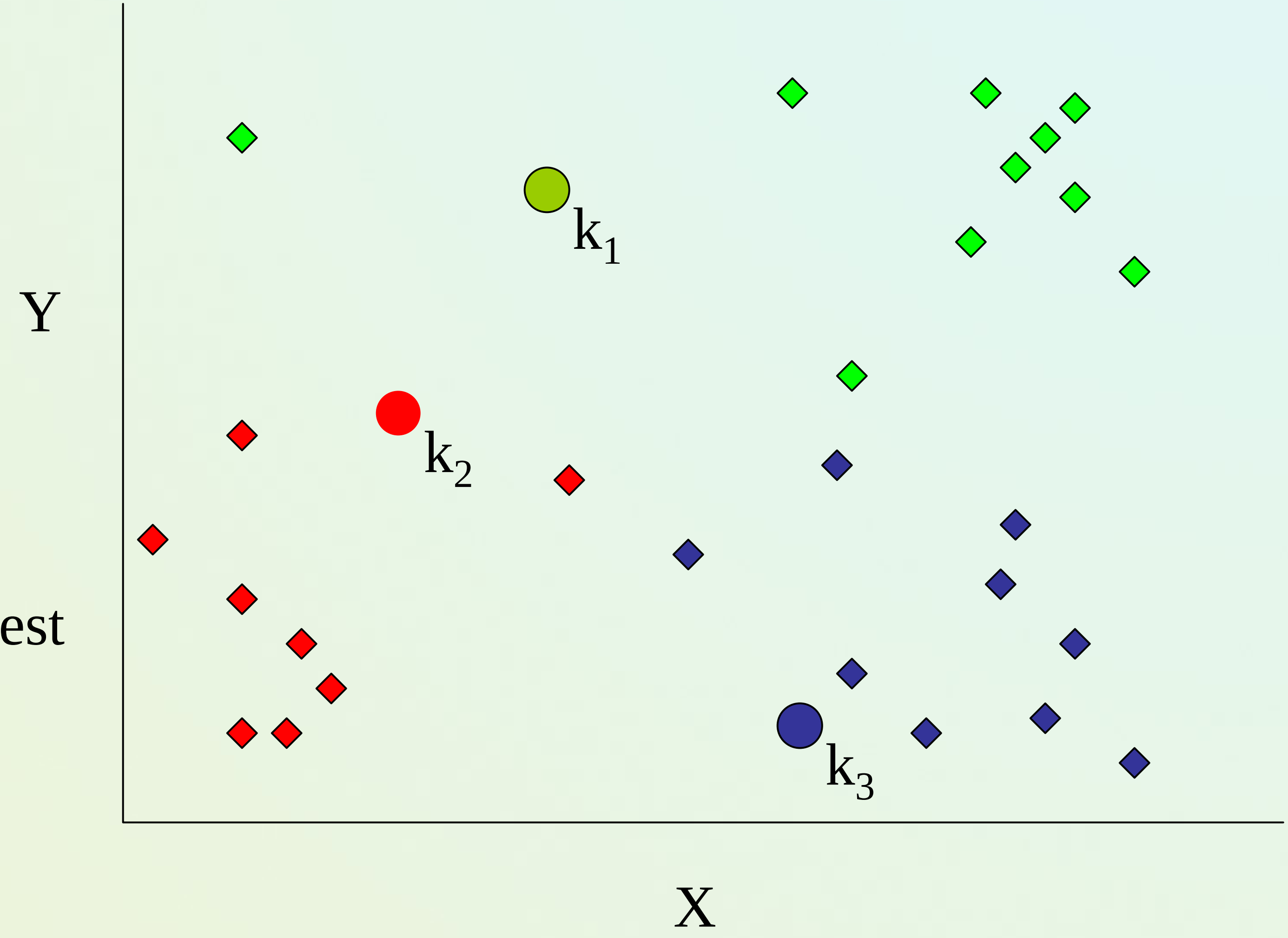
reassign

K-means example, step 1

Pick 3 initial cluster centers (randomly)

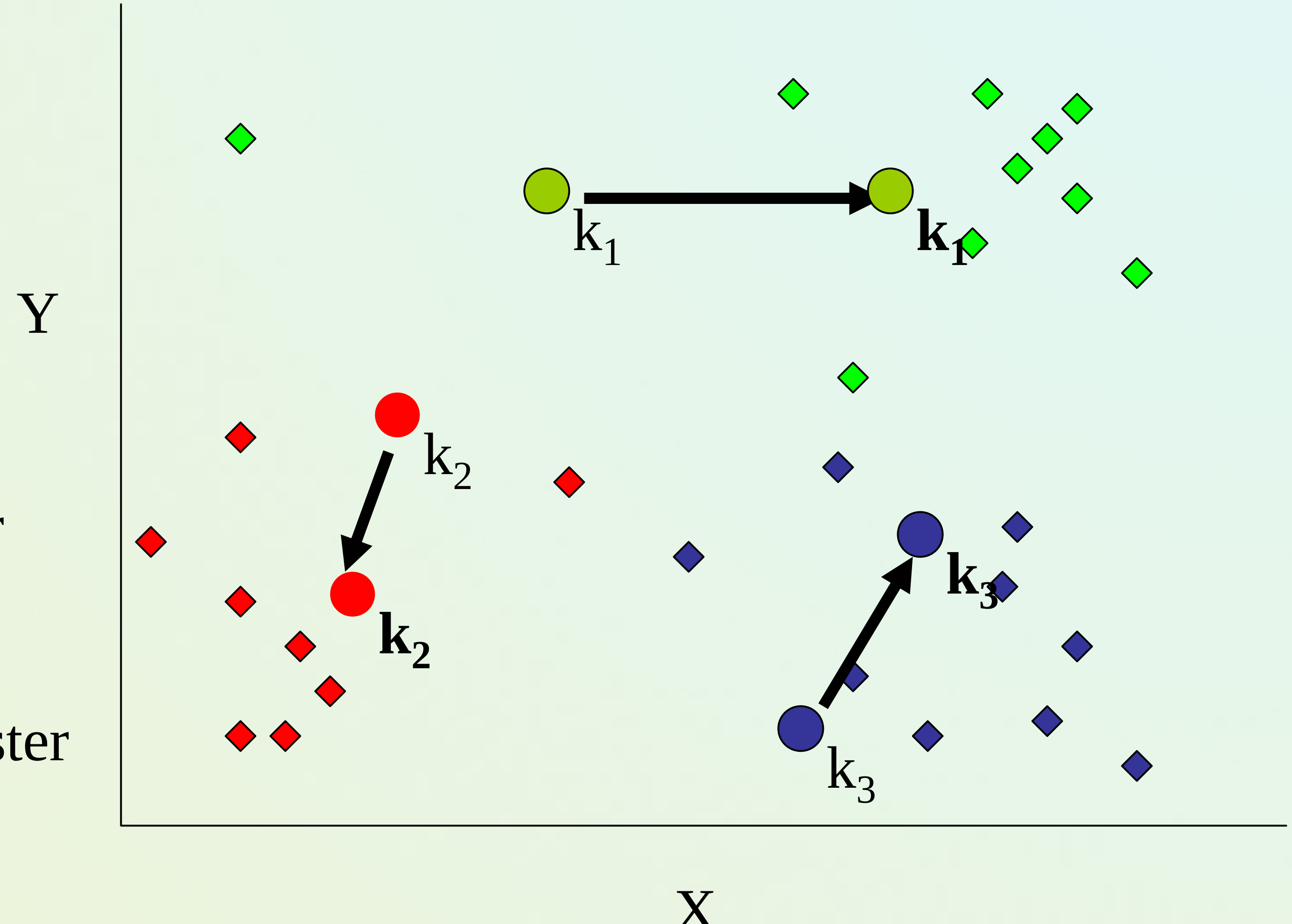


K-means example, step 2



Assign each point to the closest cluster center

K-means example, step 3

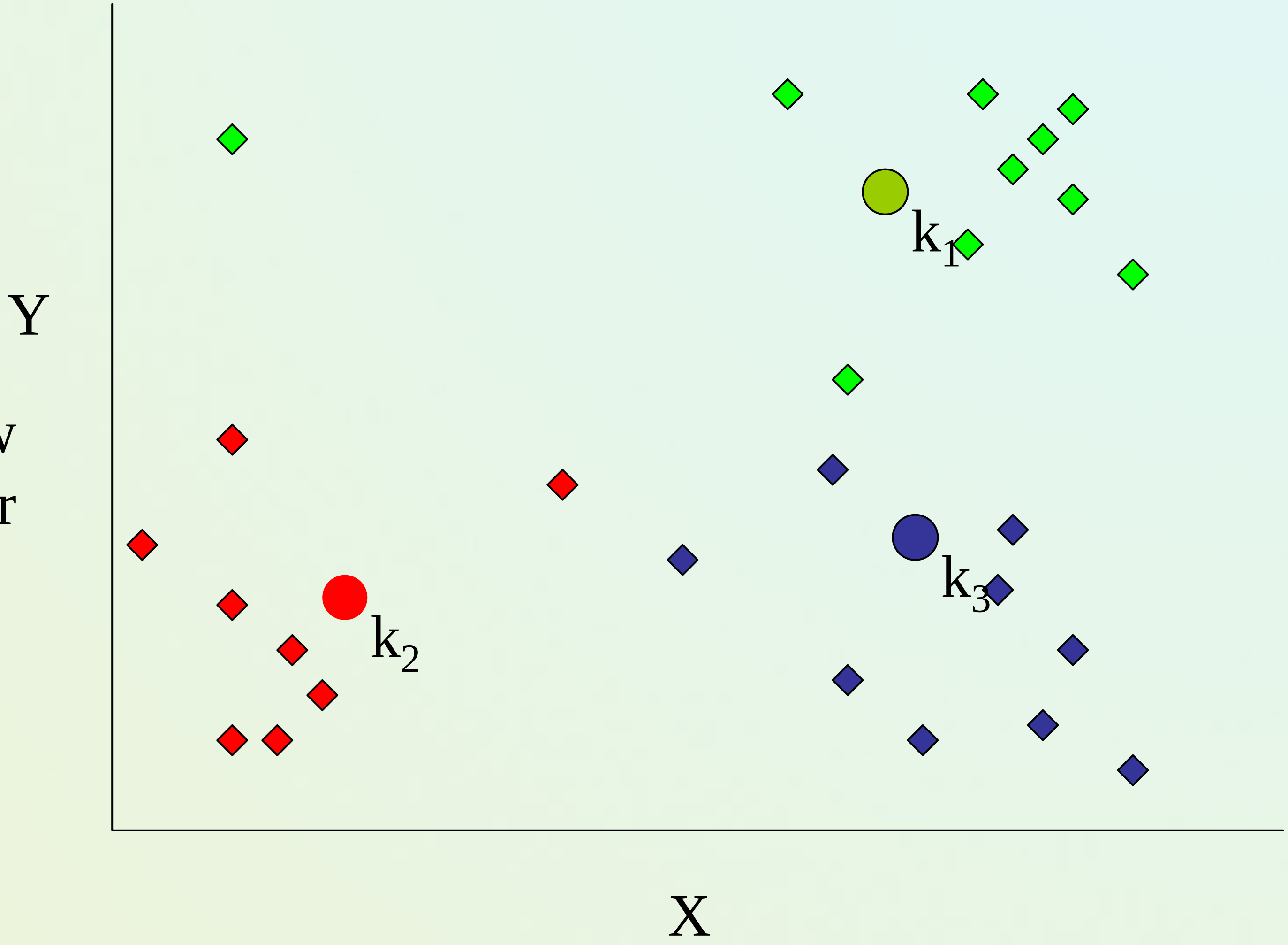


Move
each cluster
center
to the mean
of each cluster

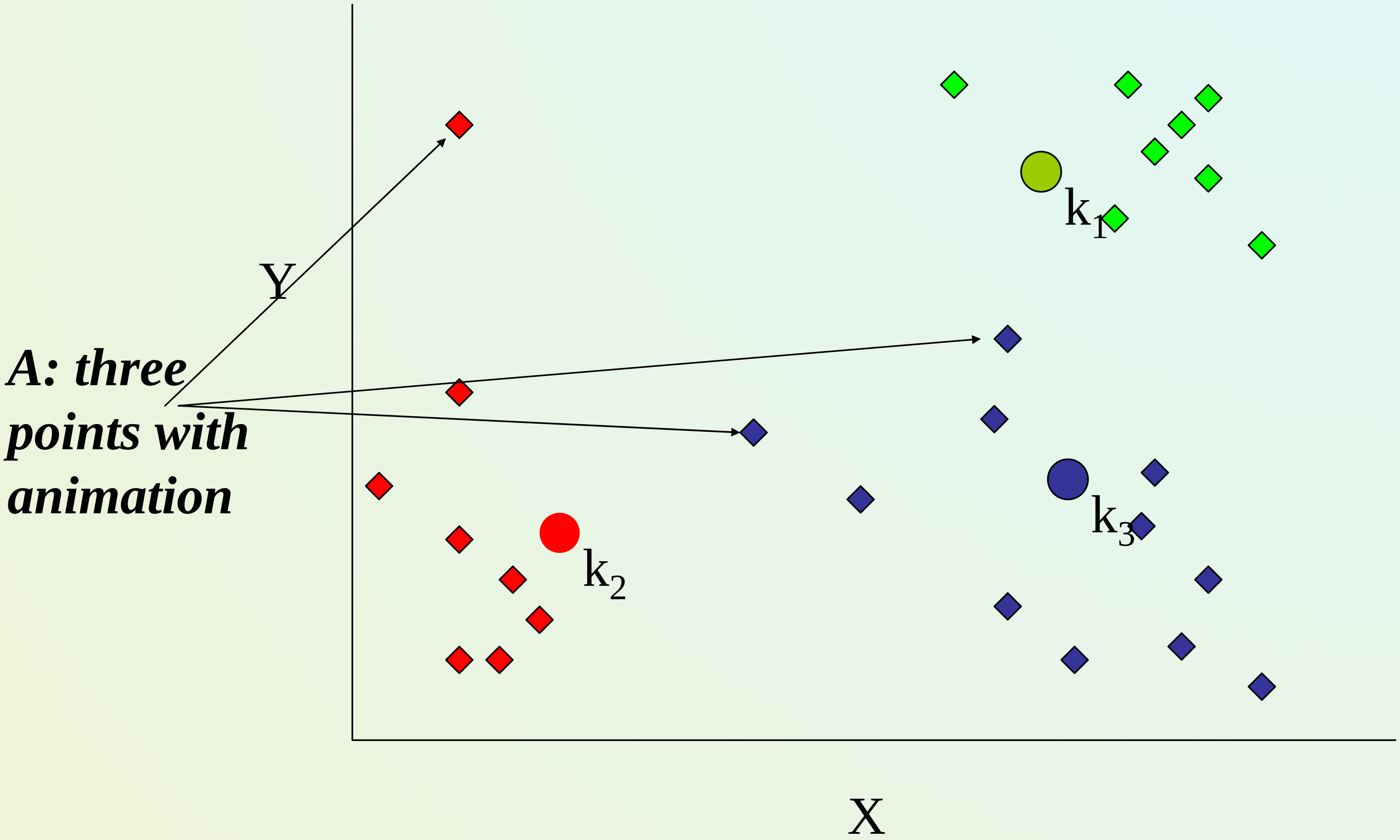
K-means example, step 4

Reassign points closest to a different new cluster center

Q: Which points are reassigned?



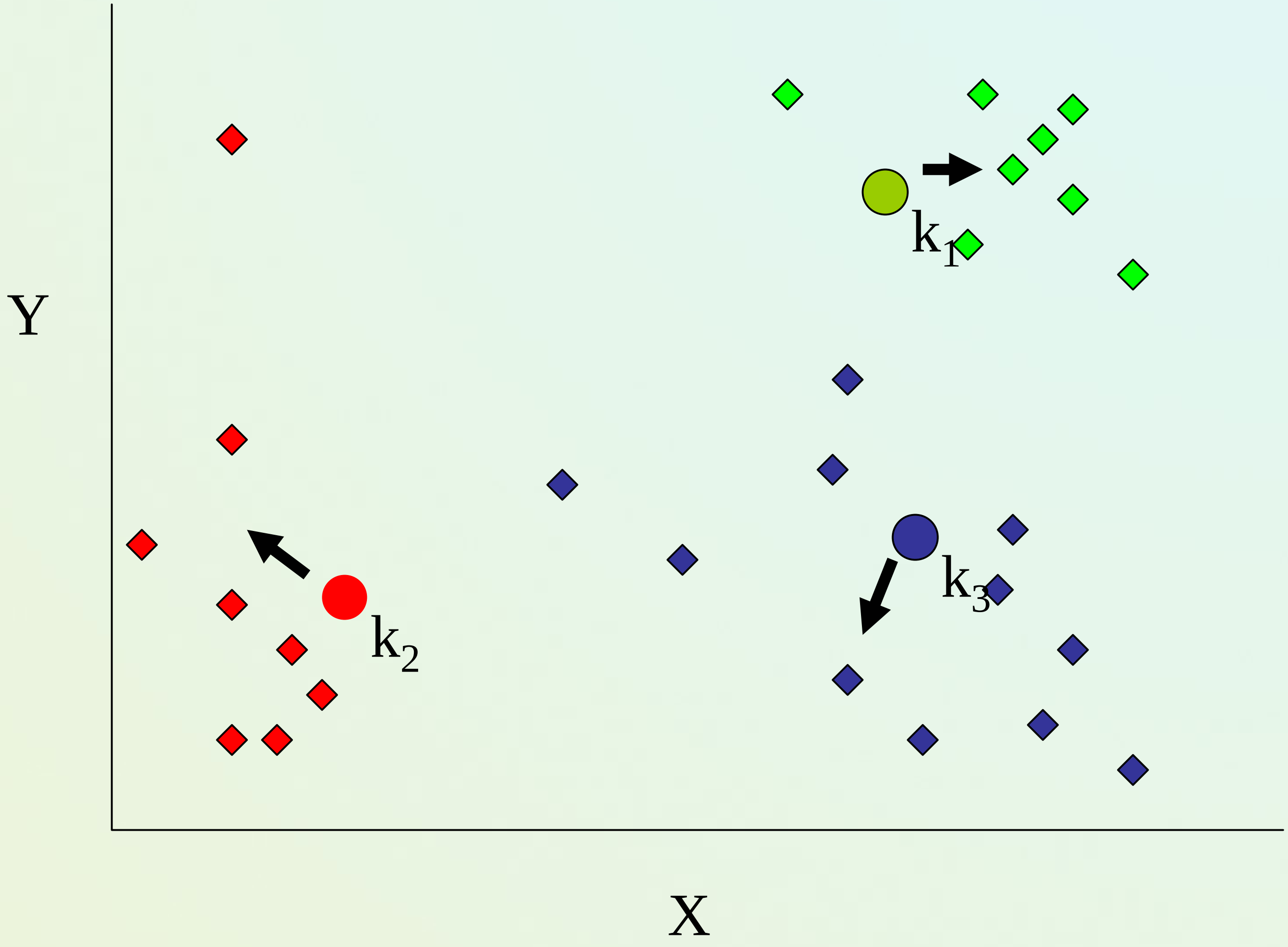
K-means example, step 4 …



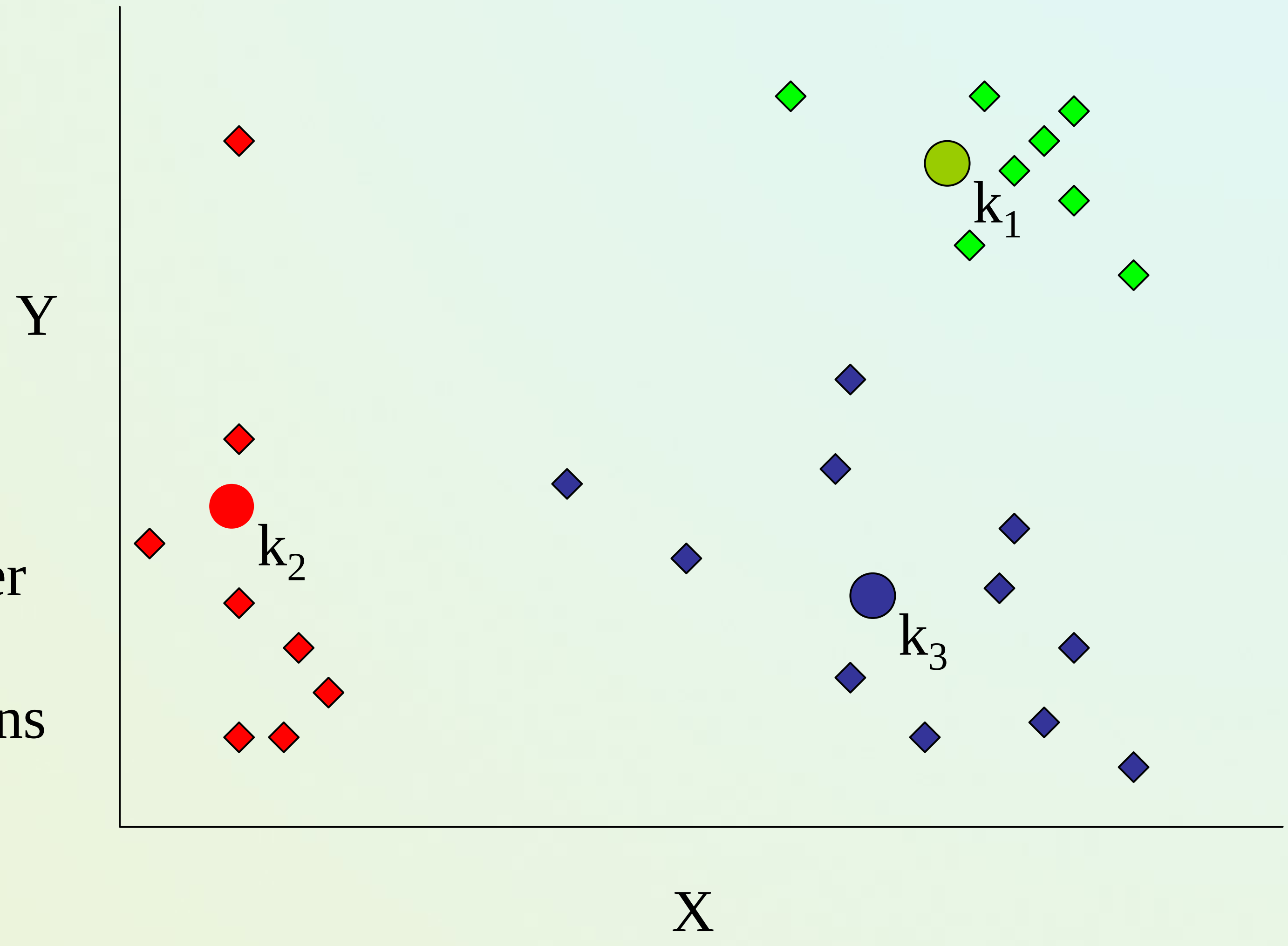
A: three points with animation

K-means example, step 4b

re-compute cluster means



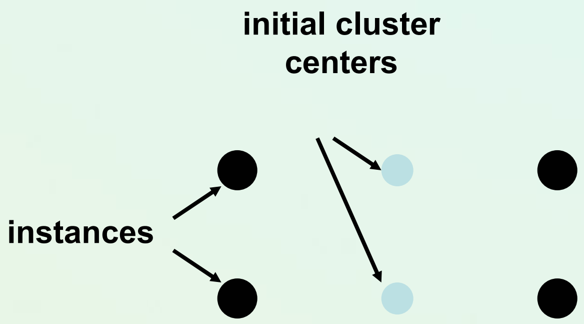
K-means example, step 5



move cluster centers to cluster means

Discussion

- Result can vary significantly depending on initial choice of seeds
- Can get trapped in local minimum
  - Example:



- To increase chance of finding global optimum: restart with different random seeds

K-means clustering summary

Advantages

- Simple, understandable
- items automatically assigned to clusters

Disadvantages

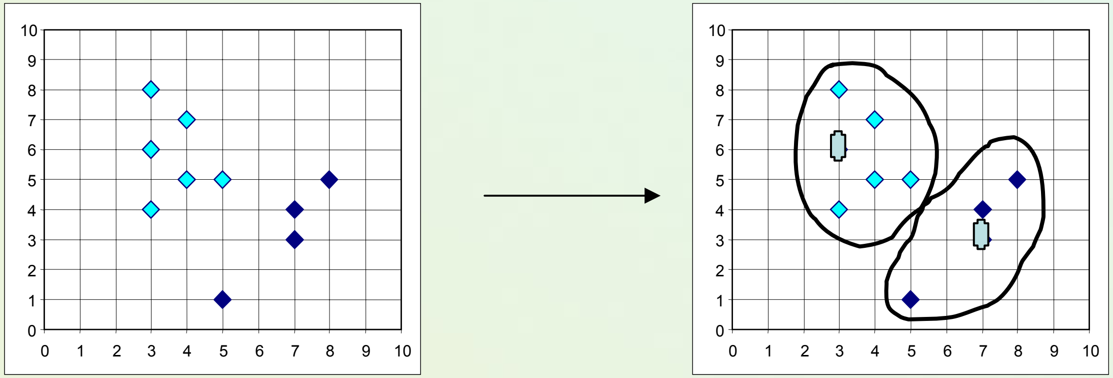
- Must pick number of clusters before hand
- Often terminates at a *local optimum.*
- All items forced into a cluster
- Too sensitive to outliers

Time Complexity

- Assume computing distance between two instances is O(m) where m is the dimensionality of the vectors.
- Reassigning clusters: O(kn) distance computations, or O(knm).
- Computing centroids: Each instance vector gets added once to some centroid: O(nm).
- Assume these two steps are each done once for I iterations: O($Iknm$).
- Linear in all relevant factors, assuming a fixed number of iterations, more efficient than O(n^2) HAC.

What is the problem of 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.
- There are other limitations – still a need for reducing costs of calculating distances to centroids.
- **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.

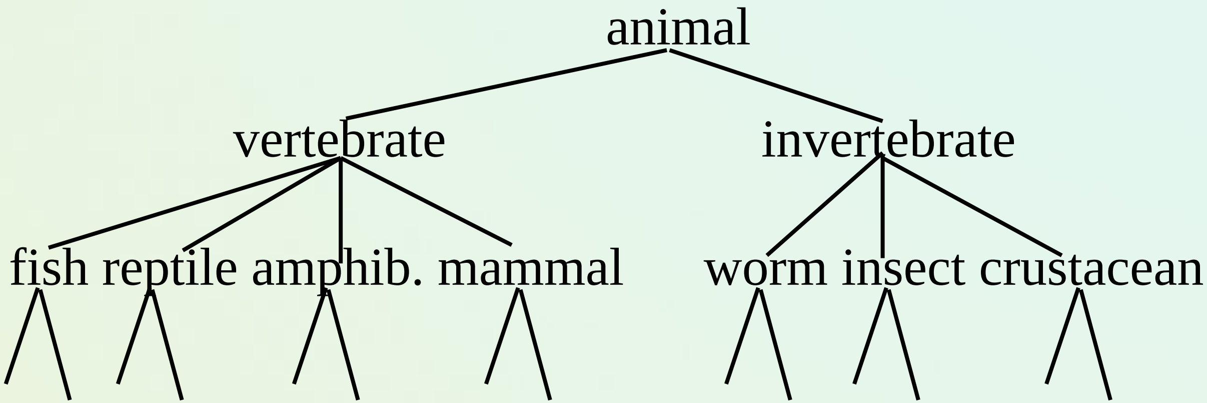


The *K-Medoids* Clustering Method

- Find *representative* objects, called medoids, in clusters
  - To achieve this goal, only the definition of distance from any two objects is needed.
- *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).

Hierarchical Clustering

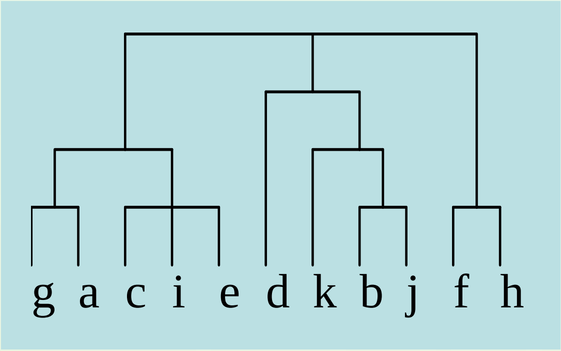
- Build a tree-based hierarchical taxonomy (*dendrogram*) from a set of unlabeled examples.



- Recursive application of a standard clustering algorithm can produce a hierarchical clustering.

*Hierarchical clustering

- Bottom up (aglomerative)
  - Start with single-instance clusters
  - At each step, join the two closest clusters
  - Design decision: distance between clusters
    - e.g. two closest instances in clusters vs. distance between means
- Top down (divisive approach / deglomerative)
  - Start with one universal cluster
  - Find two clusters
  - Proceed recursively on each subset
  - Can be very fast
- Both methods produce a *dendrogram*



HAC Algorithm (aglomerative)

```
Start with all instances in their own cluster.
Until there is only one cluster:
    Among the current clusters, determine the two
        clusters, $c_i$ and $c_j$, that are most similar.
    Replace $c_i$ and $c_j$ with a single cluster $c_i \cup c_j$
```

Distance between Clusters

Single linkage minimum distance: $$d_{\min}(C_{i,}C_j) = \min_{p \in C_i, p^{'} \in C_j} \left\| p - p^{'} \right\|$$

Complete linkage maximum distance: $$d_{\max}(C_{i,}C_j) = \max_{p \in C_i, p^{'} \in C_j} \left\| p - p^{'} \right\|$$

mean distance: $$d_{mean}(C_{i,}C_j) = \left\| m_i - m_j \right\|$$

average distance: $$d_{ave}(C_{i,}C_j) = 1/(n_i n_j) \sum_{p \in C_i} \sum_{p^{'} \in C_j} \left\| p - p^{'} \right\|$$

m_i is the mean for cluster C_i n_i is the number of points in C_i

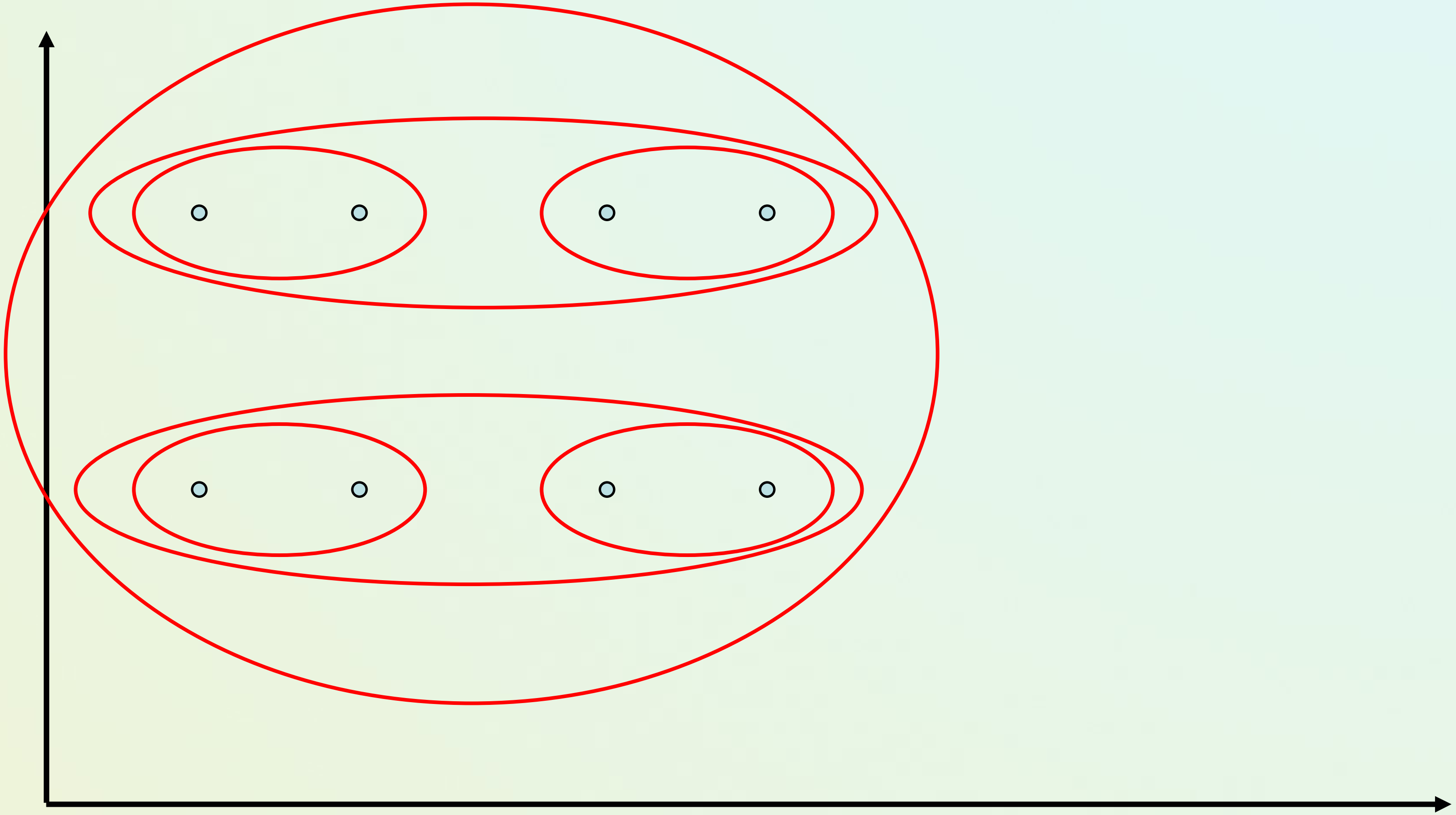
Single Link Agglomerative Clustering

- Use minium similarity of pairs:

$$sim(c_i, c_j) = \min_{x \in c_i, y \in c_j} sim(x, y)$$

- Can result in “straggly” (long and thin) clusters due to *chaining effect*.
    - Appropriate in some domains, such as clustering islands.

Single Link Example



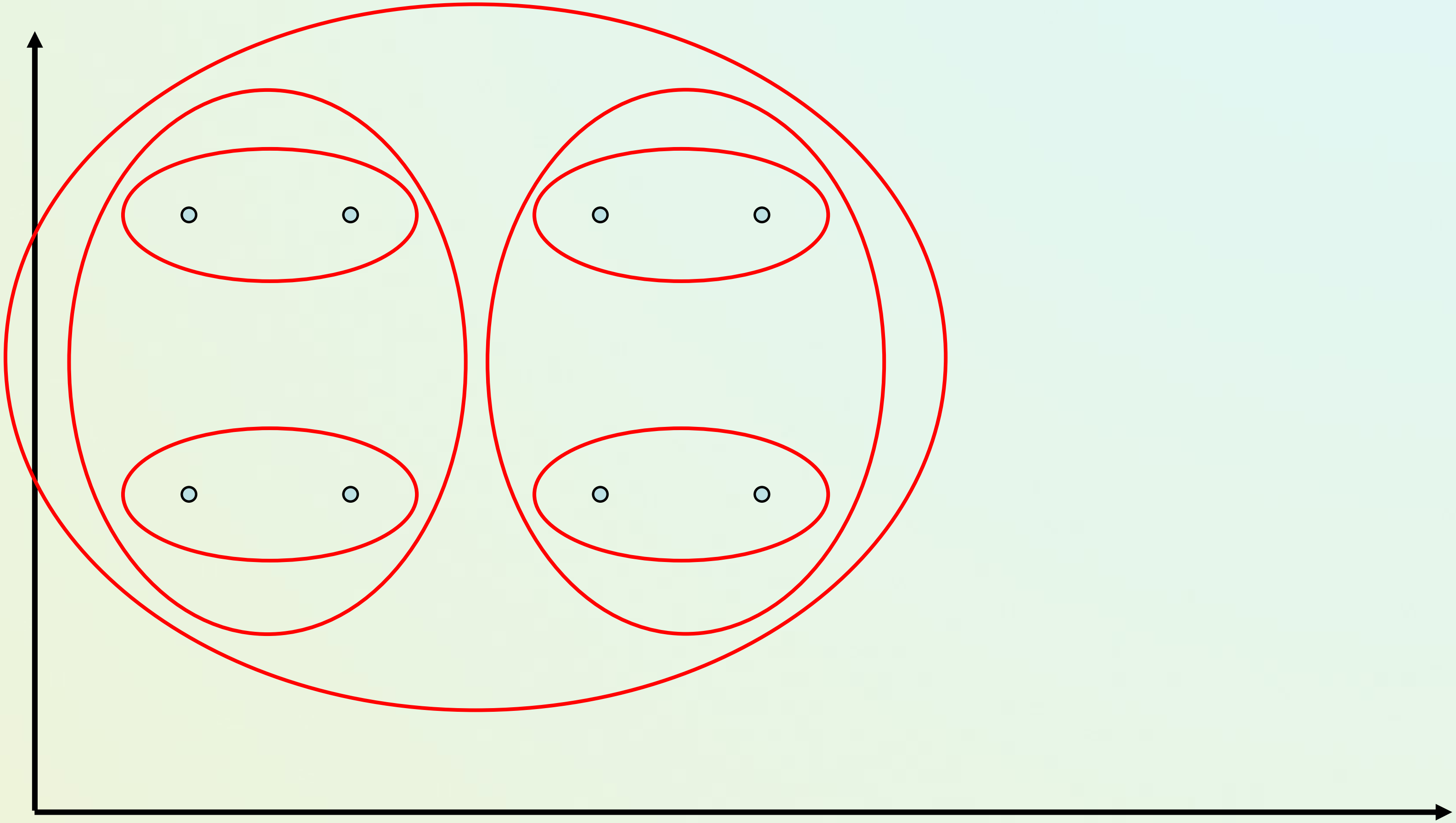
Complete Link Agglomerative Clustering

- Use maximum similarity of pairs:

$$sim(c_i,c_j) = \max_{x \in c_i, y \in c_j} sim(x,y)$$

- Makes more “tight,” spherical clusters that are typically preferable.

Complete Link Example



Single vs. Complete Linkage

- A.Jain et al.: Data Clustering. A Review.

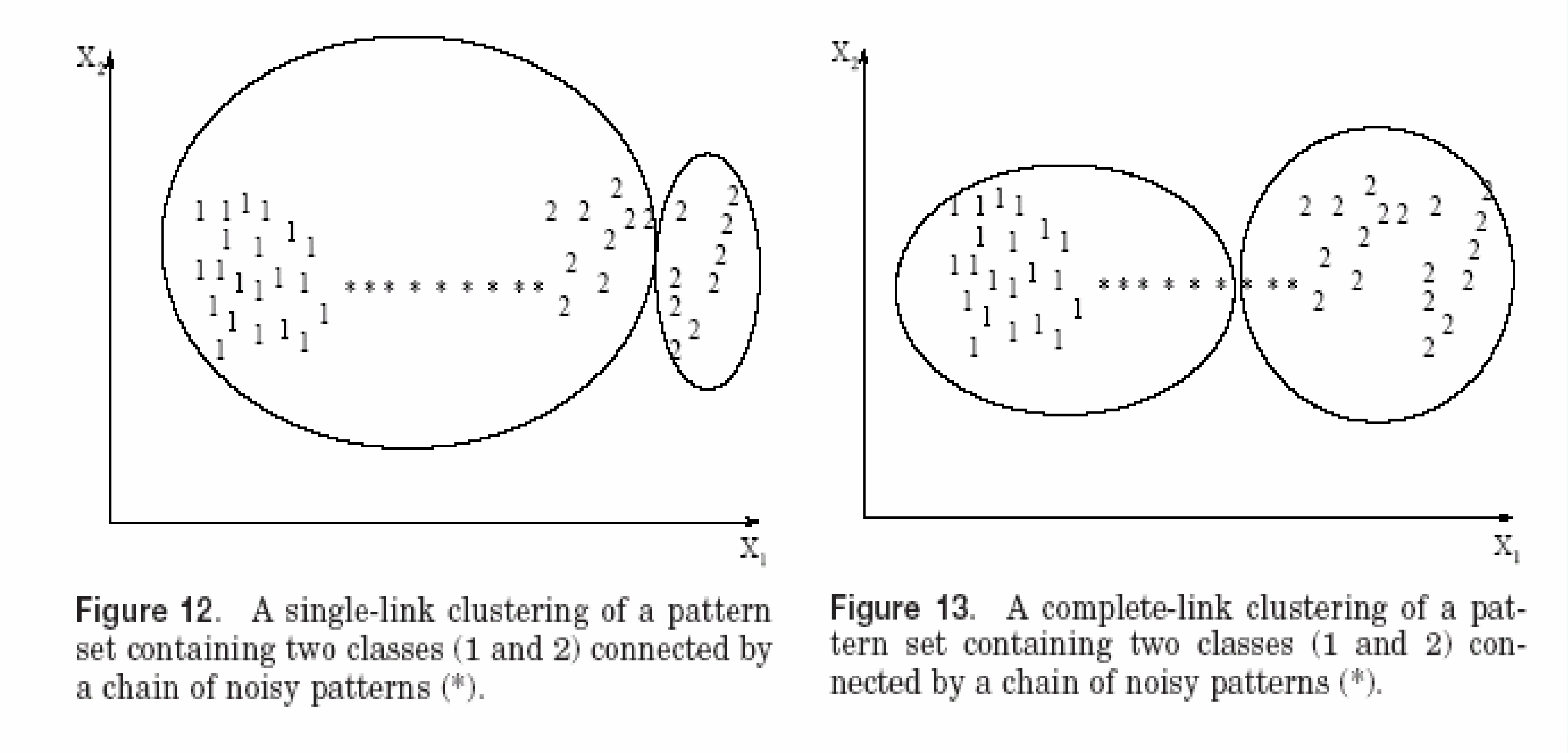


Figure 12. A single-link clustering of a pattern set containing two classes (1 and 2) connected by a chain of noisy patterns (*).

Figure 13. A complete-link clustering of a pattern set containing two classes (1 and 2) connected by a chain of noisy patterns (*).

Changing Linkage Methods

Diagram dla 22 przyp.
Pojedyncze wiązanie
Odległości euklidesowe

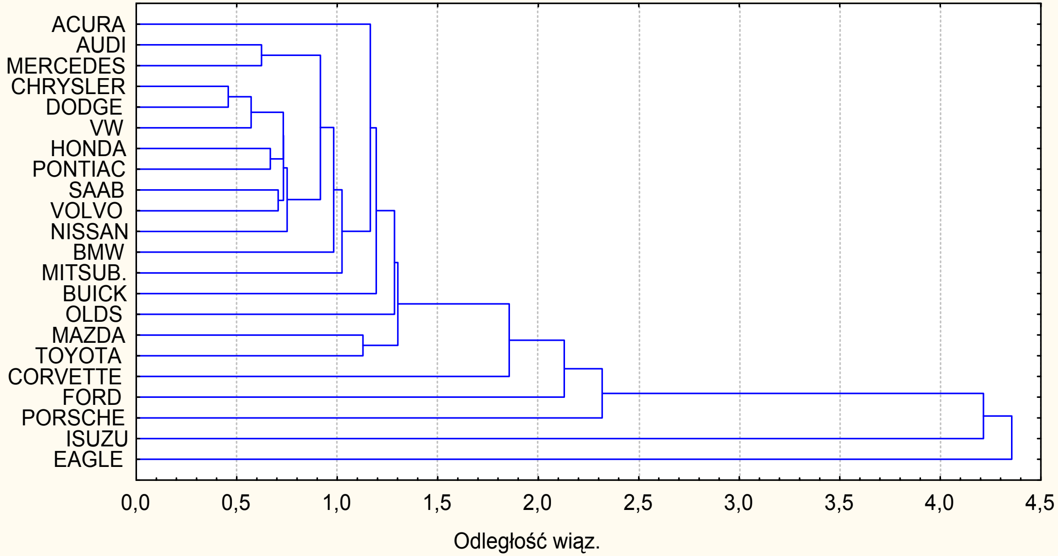
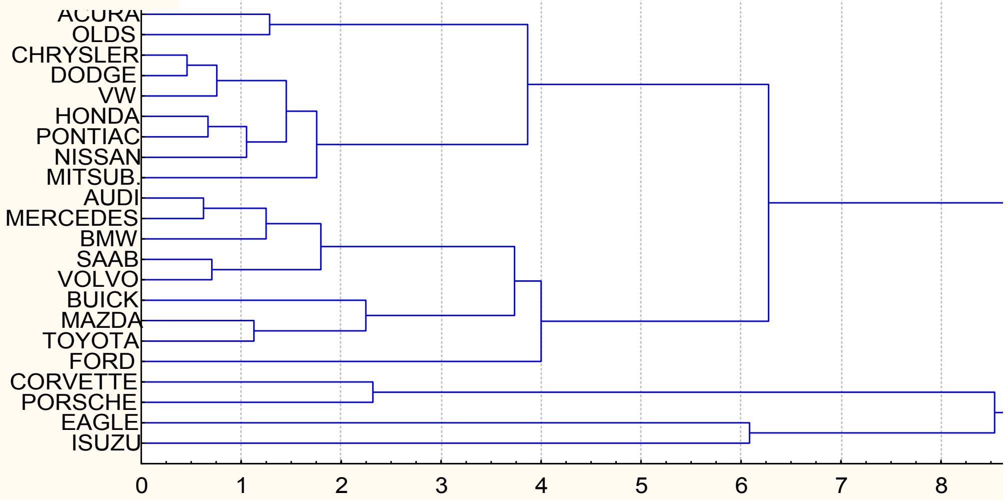
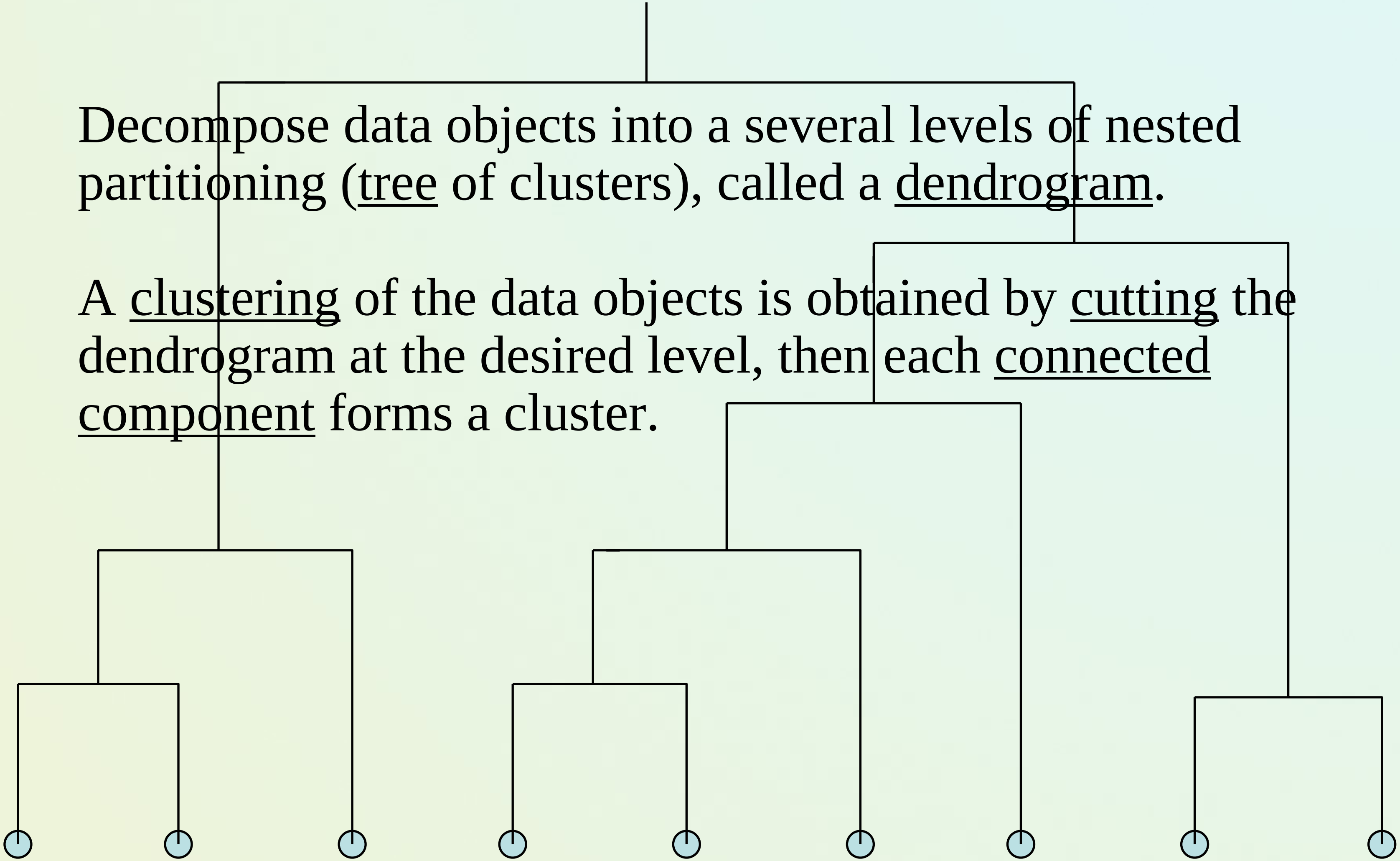


Diagram dla 22 przyp.
Metoda Warda
Odległości euklidesowe



Dendrogram: Shows How the Clusters are Merged



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

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

Computational Complexity

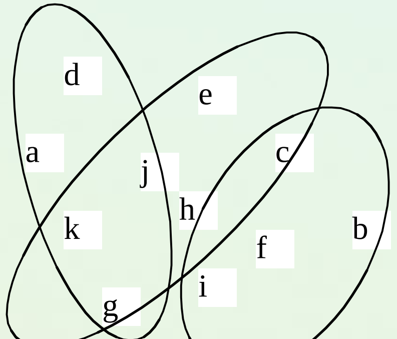
- In the first iteration, all HAC methods need to compute similarity of all pairs of n individual instances which is O(n^2).
- In each of the subsequent n–2 merging iterations, it must compute the distance between the most recently created cluster and all other existing clusters.
- In order to maintain an overall O(n^2) performance, computing similarity to each other cluster must be done in constant time.

More on 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 clustering with distance-based method:
  - BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters.
  - CURE (1998): selects well-scattered points from the cluster and then shrinks them towards the center of the cluster by a specified fraction.

Soft Clustering

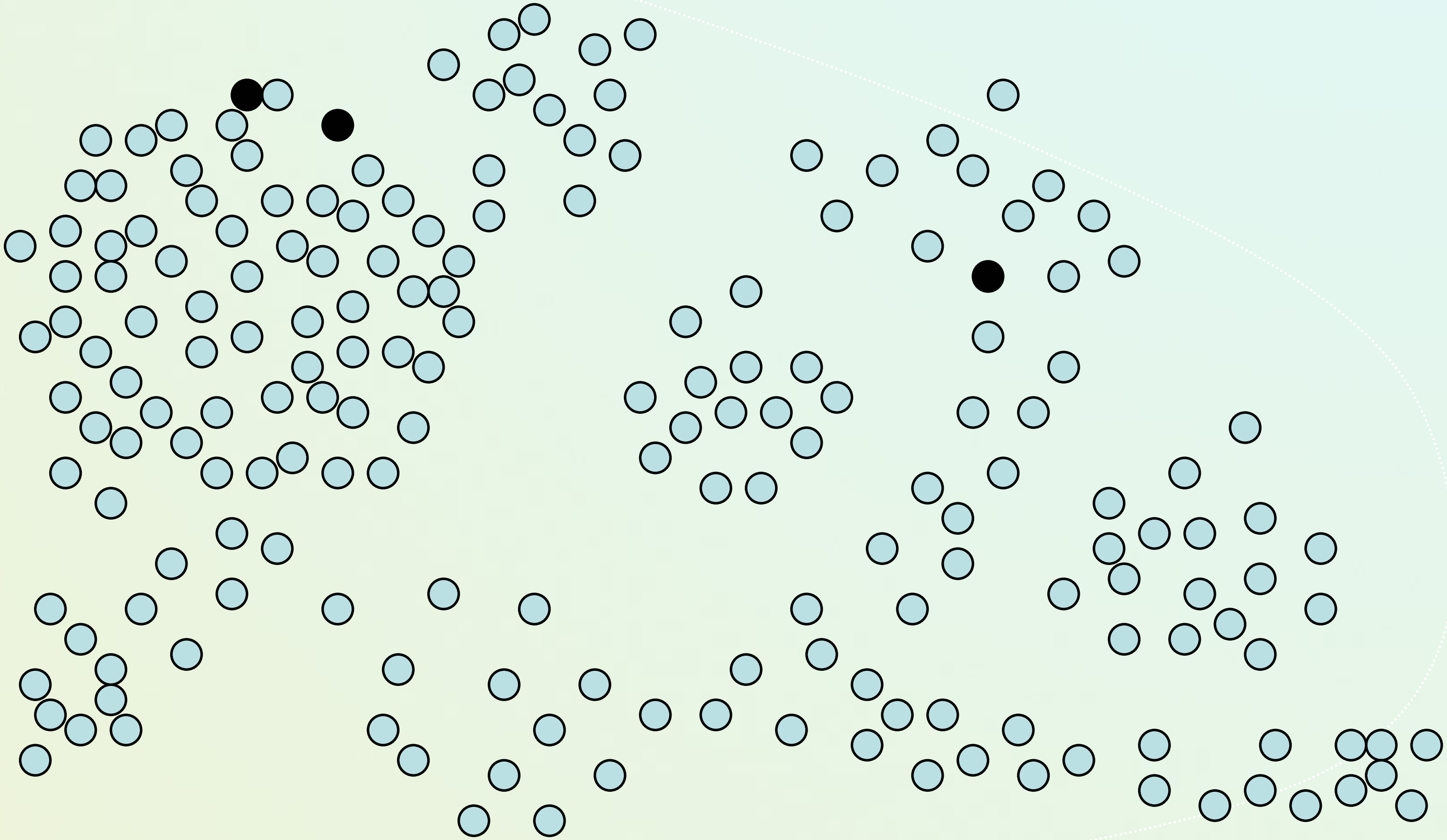
- Clustering typically assumes that each instance is given a “hard” assignment to exactly one cluster.
- Does not allow uncertainty in class membership or for an instance to belong to more than one cluster.
- *Soft clustering* gives probabilities that an instance belongs to each of a set of clusters.
- Each instance is assigned a probability distribution across a set of discovered categories (probabilities of all categories must sum to 1).



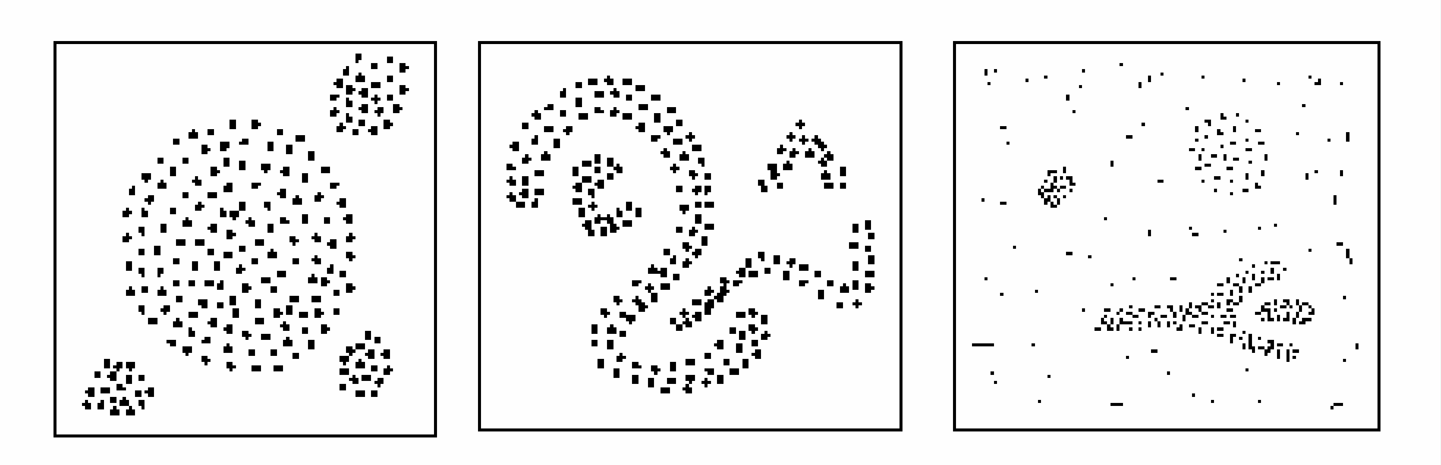
Expectation Maximization (EM Algorithm)

- Probabilistic method for soft clustering.
- Direct method that assumes *k* clusters:$\{c_1, c_2, \ldots c_k\}$
- Soft version of *k*-means.
- Assumes a probabilistic model of categories that allows computing $P(c_i \mid E)$ for each category, c_i, for a given example, *E*.
- For text, typically assume a naïve-Bayes category model.
  - Parameters $\theta = \{P(c_i), P(w_j \mid c_i): i \in \{1, \ldots k\}, j \in \{1, \ldots, |V|\}\}$

Handling Complex Shaped Clusters

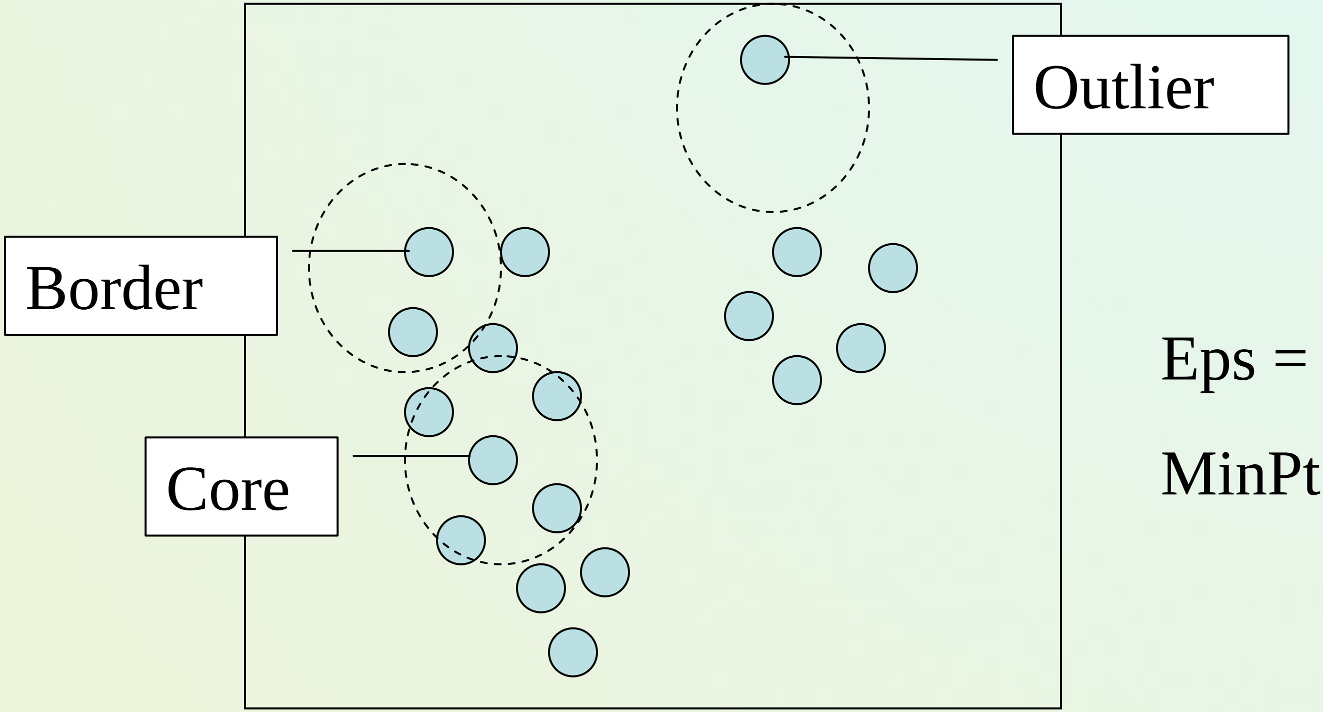


Density-Based Clustering



- Clustering based on density (local cluster criterion), such as density-connected points
- Each cluster has a considerable higher density of points than outside of the cluster

DBSCAN: General Ideas



Eps = 1cm

MinPts = 5

Clustering Evaluation

- Manual inspection
- Benchmarking on existing labels
  - Comparing clusters with ground-truth categories
- Cluster quality measures
  - distance measures
  - high similarity within a cluster, low across clusters