Classification and Clustering Analysis

Jarek Szlichta
http://data.science.uoit.ca/

Classification:

- Rule-Based Classification
- Model Evaluation and Selection
- Techniques to Improve Classification Accuracy: Ensemble Methods
- Summary

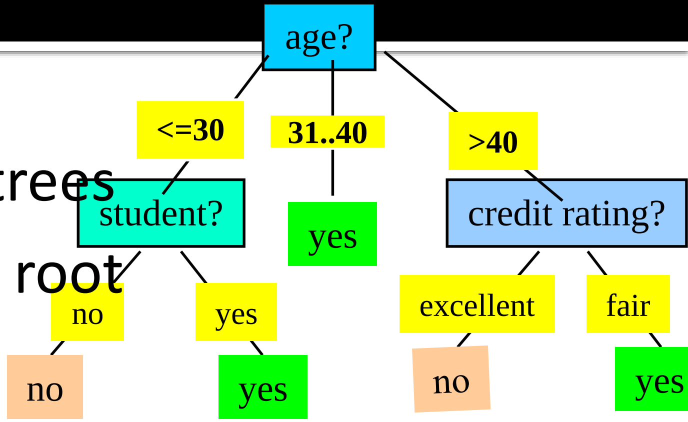
Using IF-THEN Rules for Classification

- Represent the knowledge in the form of IF-THEN rules

  R: IF *age* = youth AND *student* = yes THEN *buys_computer* = yes

  - Rule antecedent/precondition vs. rule consequent
- Assessment of a rule: *coverage* and *accuracy*
  - n_{covers} = # of tuples covered by R
  - $n_{correct}$ = # of tuples correctly classified by R
- If more than one rule are triggered, need **conflict resolution**
  - Size ordering: assign the highest priority to the triggering rules that has the "toughest" requirement (i.e., with the *most attribute tests*)

Rule Extraction from a Decision Tree



- Rules are *easier to understand* than large trees
- One rule is created *for each path* from the root to a leaf
- The leaf holds the class prediction
- Rules are mutually exclusive

- Example: Rule extraction from our *buys_computer* decision-tree

IF *age* = young AND *student* = *no* THEN *buys_computer* = *no*
IF *age* = young AND *student* = *yes* THEN *buys_computer* = *yes*
IF *age* = mid-age THEN *buys_computer* = *yes*
IF *age* = old AND *credit_rating* = *excellent* THEN *buys_computer* = *no*
IF *age* = old AND *credit_rating* = *fair* THEN *buys_computer* = *yes*

Classification: Basic Concepts

- Rule-Based Classification
- Model Evaluation and Selection
- Techniques to Improve Classification Accuracy: Ensemble Methods
- Summary

Model Evaluation and Selection

- Evaluation metrics: How can we measure accuracy? Other metrics to consider?
- Use **test set** of class-labeled tuples instead of training set when assessing accuracy
- Method for estimating a classifier's accuracy:
  - Holdout method, random subsampling

Classifier Evaluation Metrics: Confusion Matrix

Confusion Matrix:

Actual class\Predicted class	C_1	$\neg C_1$
C_1	**True Positives (TP)**	**False Negatives (FN)**
$\neg C_1$	**False Positives (FP)**	**True Negatives (TN)**

Example of Confusion Matrix:

Actual class\Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	**6954**	**46**	7000
buy_computer = no	**412**	**2588**	3000
Total	7366	2634	10000

Accuracy, Error Rate, Sensitivity and Specificity

A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P'	**N'**	**All**

- **Classifier Accuracy,** or recognition rate: percentage of test set tuples that are correctly classified

  Accuracy = (TP + TN)/All
- **Error rate:** *1 – accuracy*, or

  Error rate = (FP + FN)/All

- **Class Imbalance Problem**:
  - One class may be *rare*, e.g. fraud, or HIV-positive
  - **Sensitivity**: True Positive recognition rate
    - **Sensitivity = TP/P**
  - **Specificity**: True Negative recognition rate
    - **Specificity = TN/N**

Precision and Recall, and F-measures

- **Precision**: exactness – what % of tuples that the classifier labeled as positive are actually positive

$$precision = \frac{TP}{TP + FP}$$

- **Recall:** completeness – what % of positive tuples did the classifier label as positive?

$$recall = \frac{TP}{TP + FN}$$

- Perfect score is 1.0
- ***F* measure (*F_1* or *F*-score)**: harmonic mean of precision and recall,

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P'	**N'**	**All**

Classifier Evaluation Metrics: Example

Actual Class\Predicted class	cancer = yes	cancer = no	Total	Recognition(%)
cancer = yes	**90**	**210**	300	30.00 (*sensitivity*
cancer = no	**140**	**9560**	9700	98.56 (*specificity)*
Total	230	9770	10000	96.40 (*accuracy*)

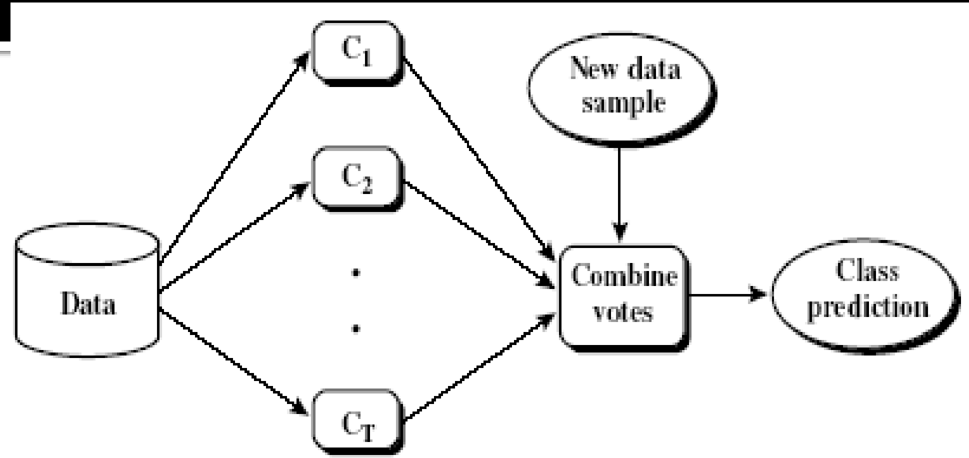
- *Precision* = 90/230 = 39.13% *Recall* = 90/300 = 30.00%

A\P	C	¬C	
C	**TP**	**FN**	**P**
¬C	**FP**	**TN**	**N**
	P'	**N'**	**All**

Classification: Basic Concepts

- Rule-Based Classification
- Model Evaluation and Selection
- Techniques to Improve Classification Accuracy: Ensemble Methods
- Summary

Ensemble Methods: Increasing the Accuracy



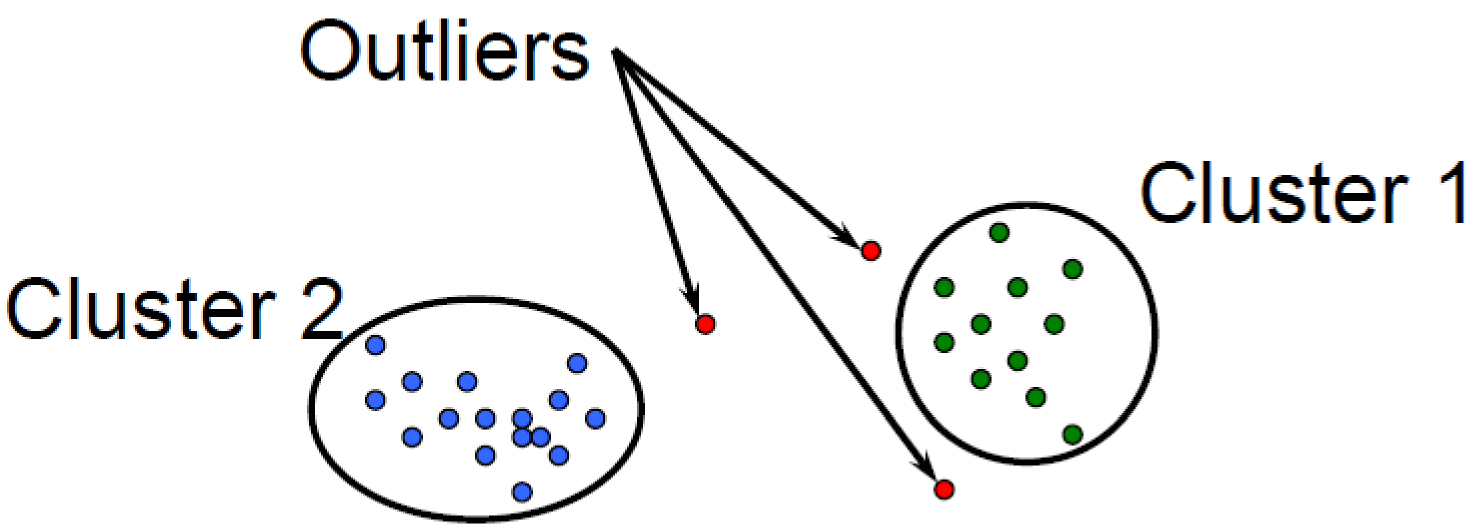
- Ensemble methods
  - Use a combination of models to increase accuracy
  - Combine a series of k learned models, M_1, M_2, …, M_t, with the aim of creating an improved model M*
- Popular ensemble methods
  - Bagging: averaging the prediction over a collection of classifiers
  - Boosting: weighted vote with a collection of classifiers

Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Summary

What is Clustering?

- Group data into clusters
  - the points in one group are similar to each other
  - and are as different as possible from the points in other groups
  - Unsupervised learning: no predefined classes



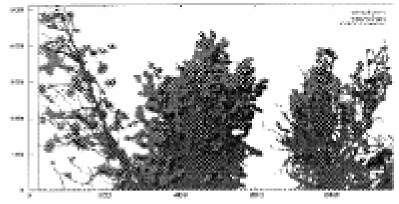
Examples of Clustering Applications

- **Marketing:**
  - Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- **Image processing:**
  - Soil scientists filter trees from background
- **Genomics:**
  - Group genes to predict possible functions of genes with unknown function
- **City-planning:**
  - Identifying groups of houses according to their house type, value, and geographical location
- **WWW:**
  - Cluster web documents
  - Cluster web log data to discover groups of users

Example with Image Processing

- Filtering real images
  - Images of trees taken in near-infrared band (NIR) and visible wavelength (VIS)
  - 512x1024 pixels and each of them contains a pair of brightness values (NIR,VIS)

The images taken in NIR and VIS



The sunlit leaves, branches and shadows

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 result depends on
  - The clustering method
  - The similarity measure used by the method
- The quality of a clustering method is 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: $d(i, j)$
  - The definitions of distance functions are usually rather different for various variables: categorical and numerical etc. (e.g., Euclidean Distance, Manhattan Distance, Hamming Distance)

Requirements and Challenges

- Scalability (Performance)
- Ability to deal with different types of attributes
  - Numerical, binary, categorical, ordinal, linked, and mixture of these
- Interpretability and usability
- Others
  - Discovery of clusters with arbitrary shape
  - Ability to deal with noisy data
  - Insensitivity to input order
  - High dimensionality

Clustering Approaches

- Partitioning approach:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the distance to centroid
  - Typical methods: k-means, k-medoids, CLARA
- Hierarchical approach:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, CAMELEON
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSAN

Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- Summary

Partitioning Algorithms: Basic Concept

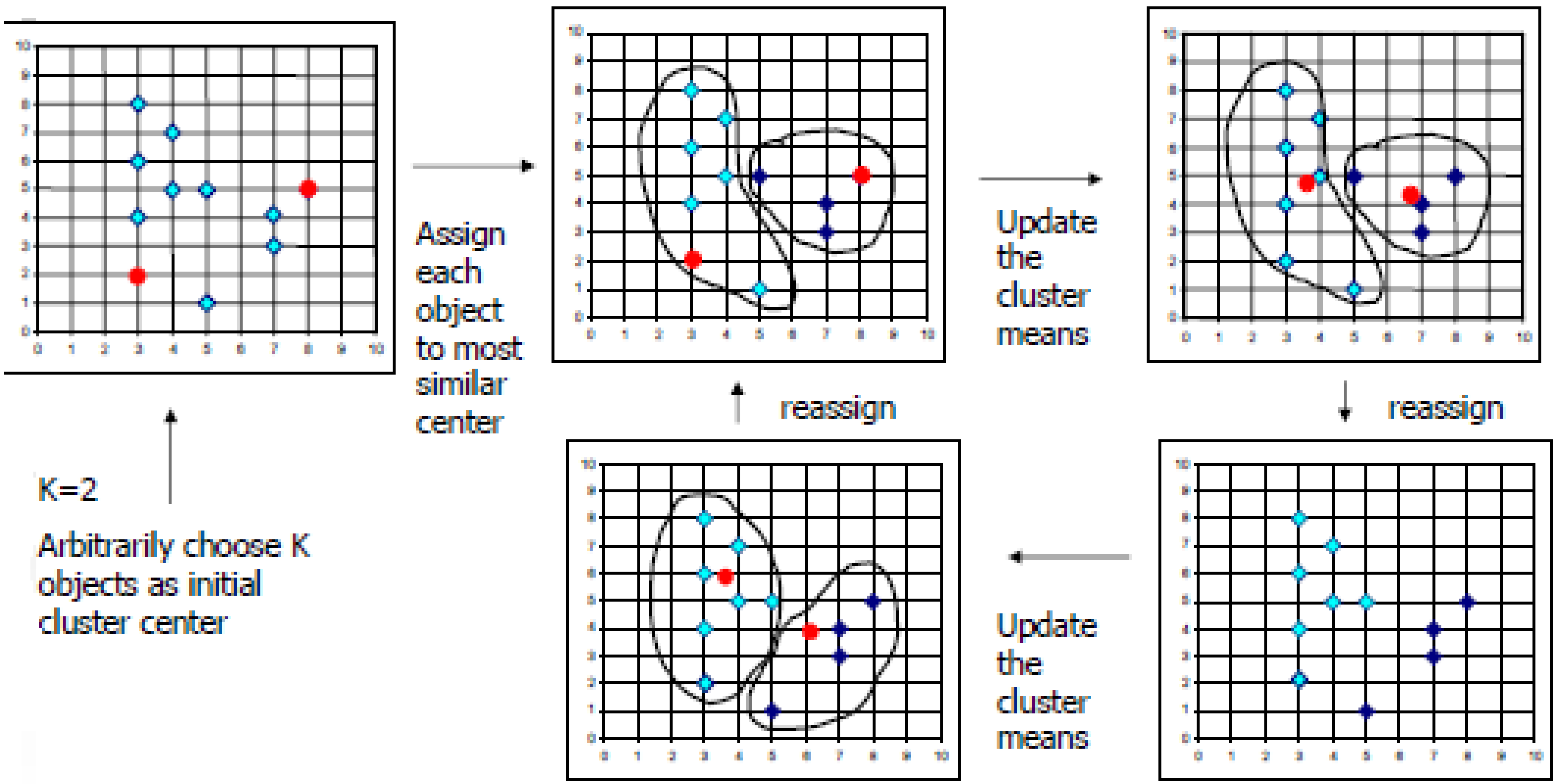
- Partitioning method: Partition *n* objects into *k* clusters
  - Optimize the chosen partitioning criterion
- Global optimal: exhaustively enumerate all partitions (not tractable for large datasets)
- Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means*: Each cluster is represented 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

- Arbitrarily choose k objects as the initial cluster center
- Until no change, do
  - (Re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster
  - Update the cluster means, i.e., calculate the mean value of the objects for each cluster

An Example of *K-Means* Clustering

- Example

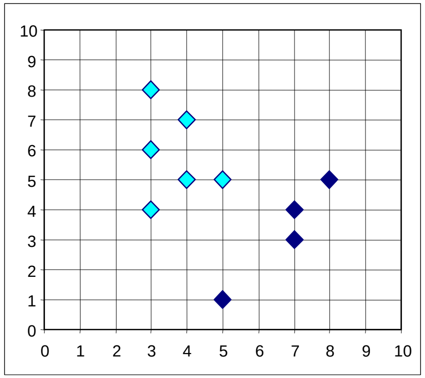


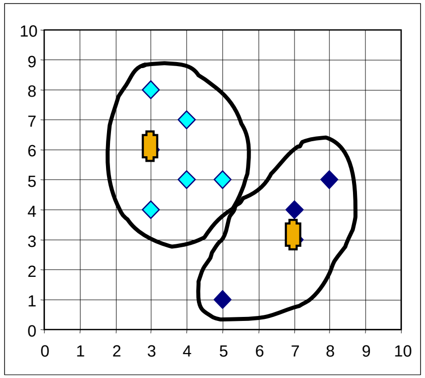
Comments on the *K-Means* Method

- Strength: *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: $O(ks^2 + k(n-k))$
- Comment: Often terminates at a *local optimal*.
- Weakness
  - Applicable only to objects in a continuous n-dimensional space
    - In comparison, k-medoids can be applied to a wide range of data
  - Need to specify *k,* the *number* of clusters, in advance
  - Sensitive to noisy data and *outliers*

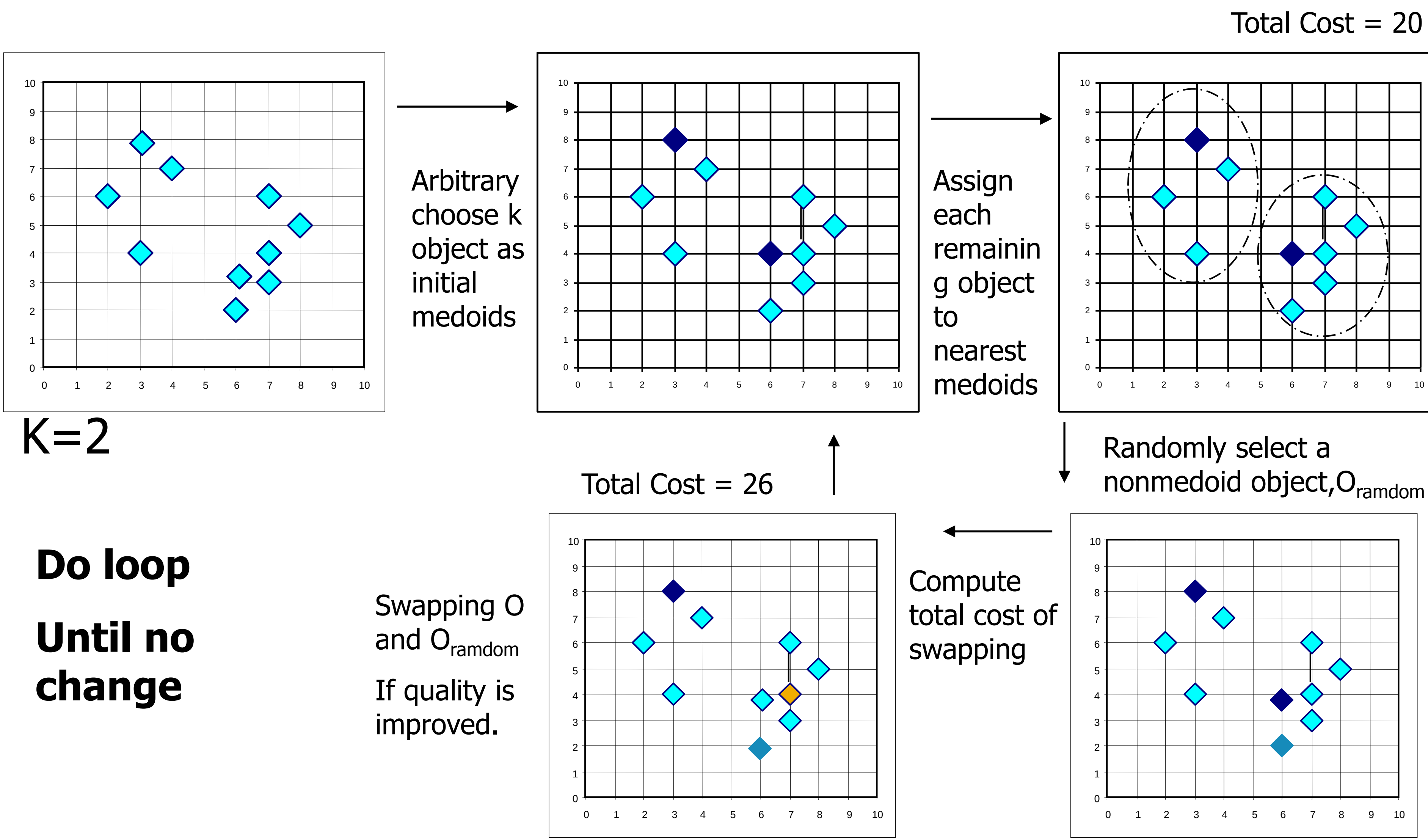
What Is the Problem of the K-Means Method?

- 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





PAM: A Typical K-Medoids Algorithm



K=2

Do loop

Until no change

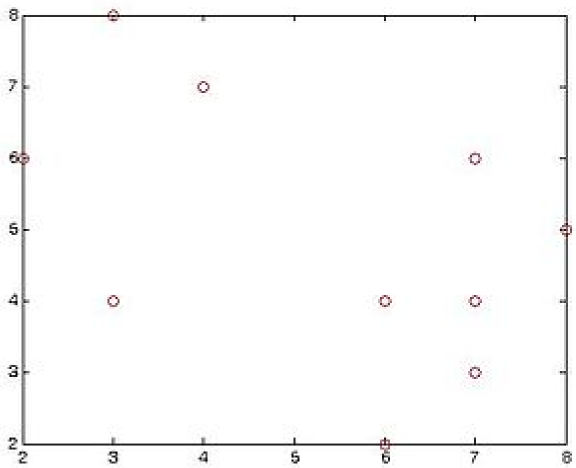
The K-Medoid Clustering Method

- *K-Medoids* Clustering: Find *representative* objects (medoids) in clusters
  - *PAM* (Partitioning Around Medoids)
    - 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
- PAM is more robust than k-means in the presence of noise and outliers
  - Medoids are less influenced by outliers

Demonstration of PAM

- Cluster the following data set of ten objects into two clusters i.e. k = 2.

X_1	2	6
X_2	3	4
X_3	3	8
X_4	4	7
X_5	6	2
X_6	6	4
X_7	7	3
X_8	7	4
X_9	8	5
X_{10}	7	6



Step 1

- Initialize *k* centers
- Let us assume x_2 and x_8 are selected as medoids, so the centers are $c_1 = (3,4)$ and $c_2 = (7,4)$
- Calculate distances to each center so as to associate each data object to its nearest medoid.
  - Cost is calculated using Manhattan distance

Step 1

- Costs to the nearest medoid are shown bold in the table

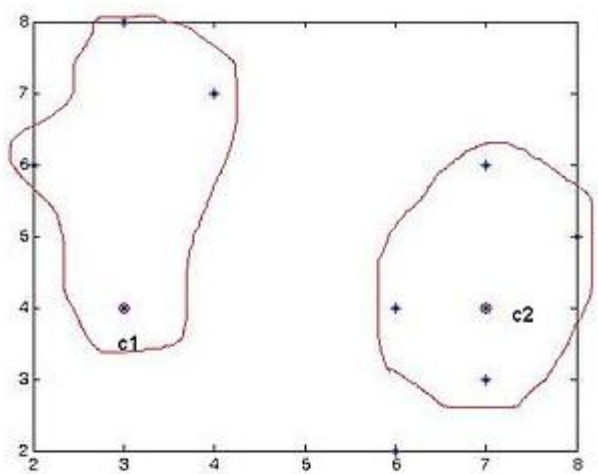
Cost (distance) to c_1					
i	c_1		Data objects (X_i)		Cost (distance)
1	3	4	2	6	**3**
3	3	4	3	8	**4**
4	3	4	4	7	**4**
5	3	4	6	2	5
6	3	4	6	4	3
7	3	4	7	3	5
9	3	4	8	5	6
10	3	4	7	6	6

Cost (distance) to c_2					
i	c_2		Data objects (X_i)		Cost (distance)
1	7	4	2	6	7
3	7	4	3	8	8
4	7	4	4	7	6
5	7	4	6	2	**3**
6	7	4	6	4	**1**
7	7	4	7	3	**1**
9	7	4	8	5	**2**
10	7	4	7	6	**2**

Step 1

- Then the clusters become:
  - $Cluster_1$ = {(3,4)(2,6)(3,8)(4,7)}
  - $Cluster_2$ = {(7,4)(6,2)(6,4)(7,3)(8,5)(7,6)}

$$\begin{aligned}\text{total cost} &= \{\text{cost}((3,4),(2,6)) + \text{cost}((3,4),(3,8)) + \text{cost}((3,4),(4,7))\} \\ &\quad + \{\text{cost}((7,4),(6,2)) + \text{cost}((7,4),(6,4)) + \text{cost}((7,4),(7,3)) \\ &\quad + \text{cost}((7,4),(8,5)) + \text{cost}((7,4),(7,6))\} \\ &= (3+4+4) + (3+1+1+2+2) \\ &= 20\end{aligned}$$



Step 2

- Select randomly one of the nonmedoids O′
- Let us assume O′ = (7,3), i.e. x_7
- So now the medoids are c_1(3,4) and O′(7,3)
  - calculate the total cost involved

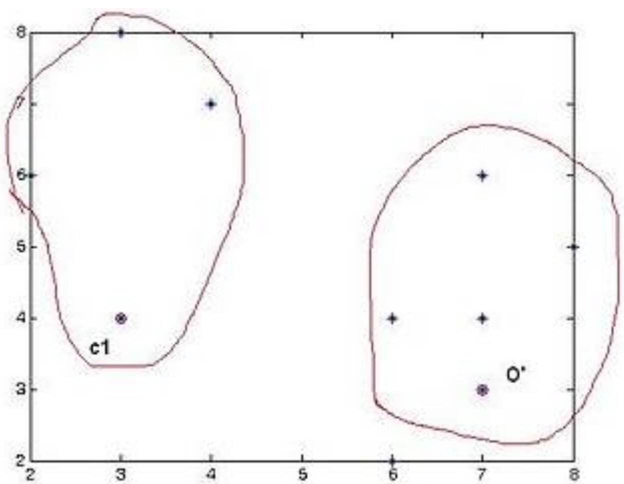
Step 2

i	c_1		Data objects ($\mathbf{X}_i$)		Cost (distance)
1	3	4	2	6	**3**
3	3	4	3	8	**4**
4	3	4	4	7	**4**
5	3	4	6	2	5
6	3	4	6	4	3
8	3	4	7	4	4
9	3	4	8	5	6
10	3	4	7	6	6

i	O'		Data objects ($\mathbf{X}_i$)		Cost (distance)
1	7	3	2	6	8
3	7	3	3	8	9
4	7	3	4	7	7
5	7	3	6	2	**2**
6	7	3	6	4	**2**
8	7	3	7	4	**1**
9	7	3	8	5	**3**
10	7	3	7	6	**3**

Step 2

- Total cost is 22 (3 + 4 + 4 + 2 + 2 + 1 + 3 + 3)
- So moving to O’ would be a bad idea, so the previous choice was better
- So we try other nonmedoids and found that our first choice was the best. So the configuration does not change and algorithm terminates here (i.e. there is no change in the medoids).
  - In practice it may happen some data points may shift from one cluster to another cluster depending upon their closeness to medoid!



Chapter 10. Cluster Analysis: Basic Concepts and Methods

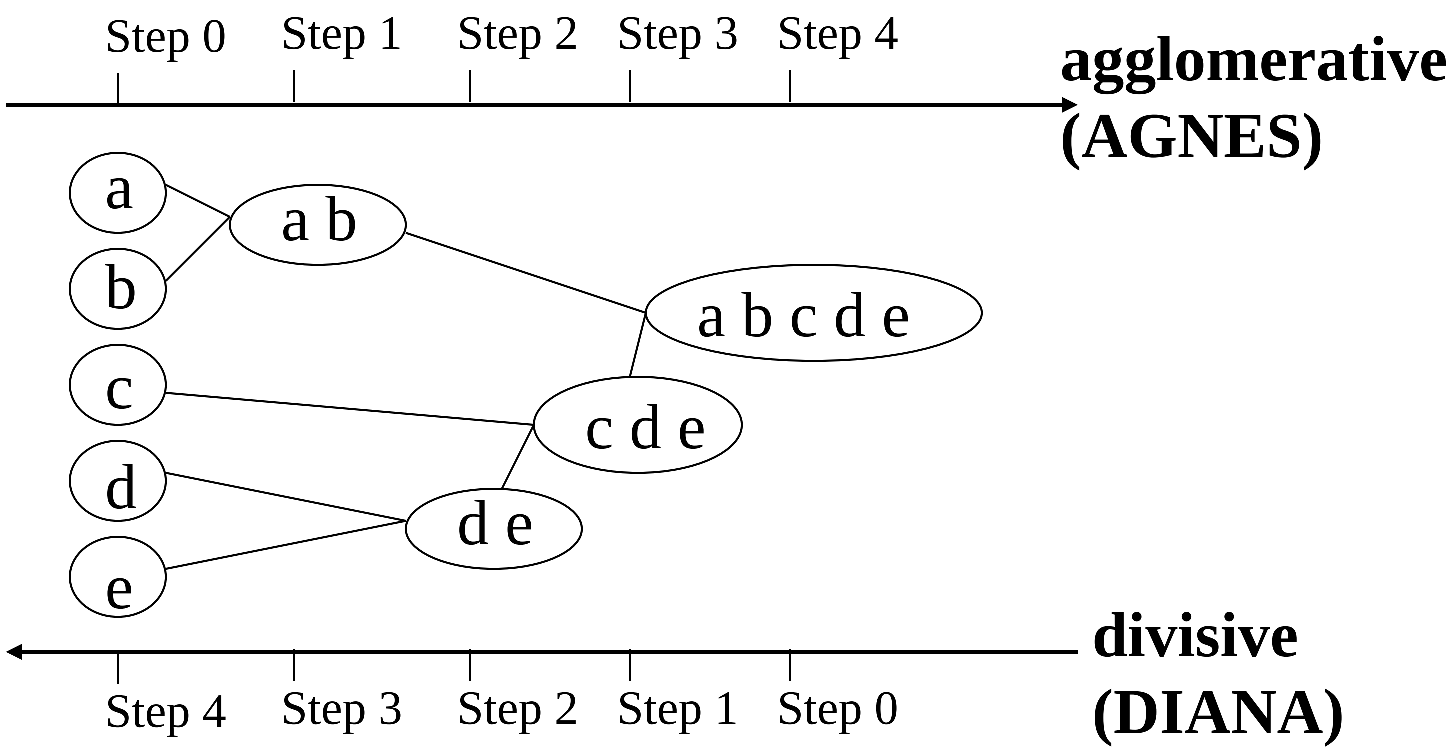
- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Evaluation of Clustering
- Summary

Hierarchical Clustering

- Iteratively merge or split clusters to form a tree of clusters
  - Two types
    - Agglomerative (bottom-up): merge clusters iteratively
      - Start by placing each object in its own cluster
      - Merge these small clusters into larger and larger clusters
      - until all objects are in a single cluster
    - Divisive (top-down): split a cluster iteratively
      - Start with all objects in one cluster and subdivide them into smaller pieces

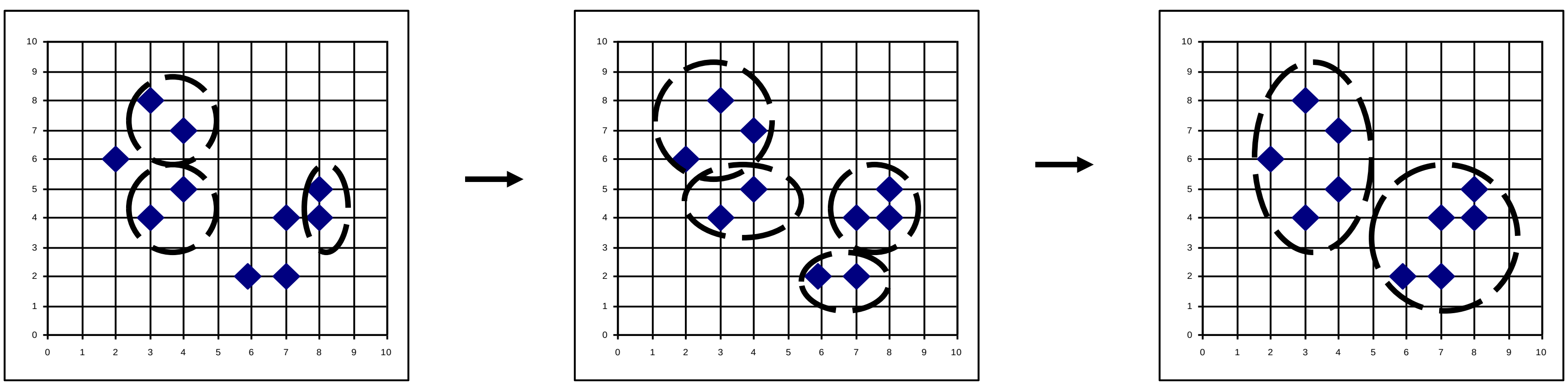
Hierarchical Clustering

- 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)

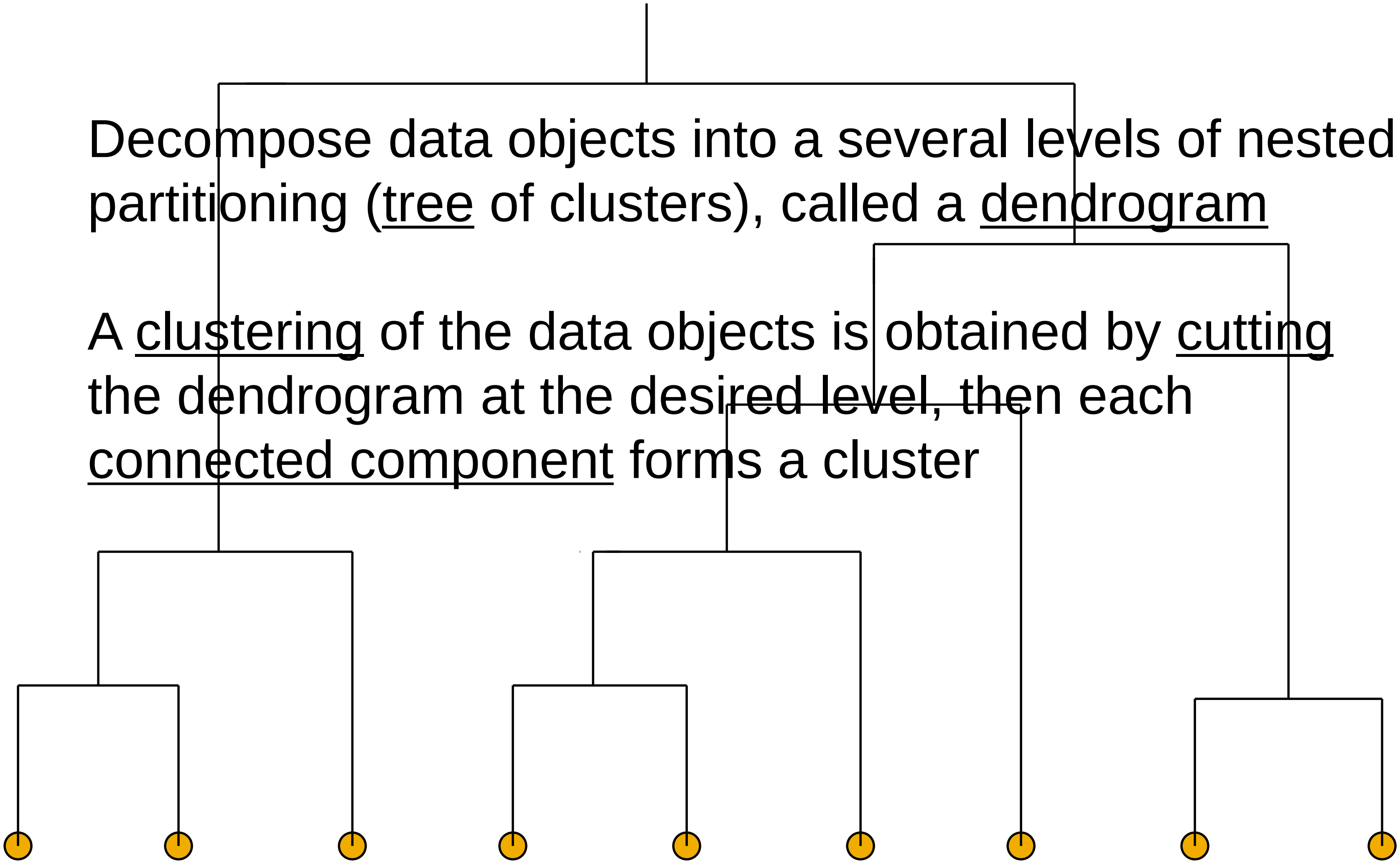
- Implemented in statistical packages, e.g., Splus
- Use the **single-link** method (see slide 42)
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



Dendrogram: Shows How Clusters are Merged

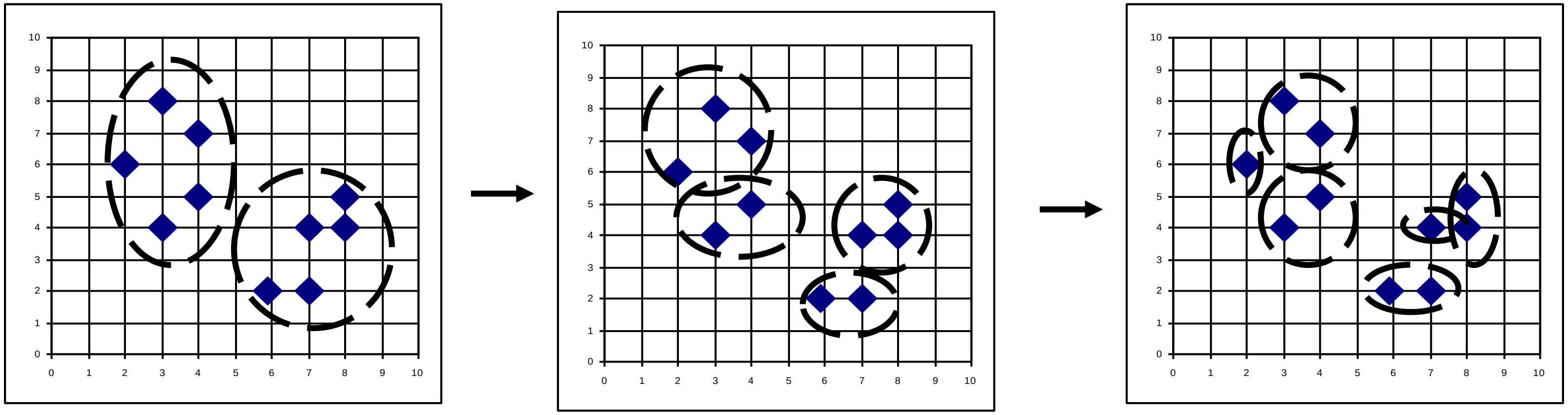
Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram

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

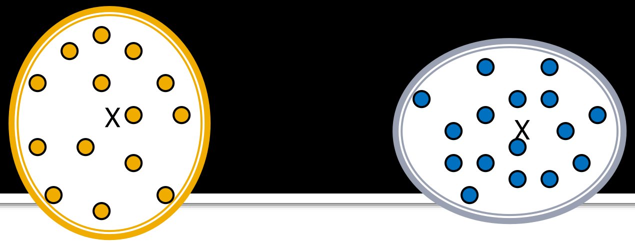


DIANA (Divisive Analysis)

- Implemented in statistical analysis packages, e.g., Splus
- Inverse order of AGNES
- Eventually each node forms a cluster on its own



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., $dist(K_i, K_j) = avg(t_{ip}, t_{jq})$
- Centroid: distance between the centroids of two clusters, i.e., $dist(K_i, K_j) = dist(C_i, C_j)$
- Medoid: distance between the medoids of two clusters, i.e., $dist(K_i, K_j) = dist(M_i, M_j)$
  - Medoid: a chosen, centrally located object in the cluster

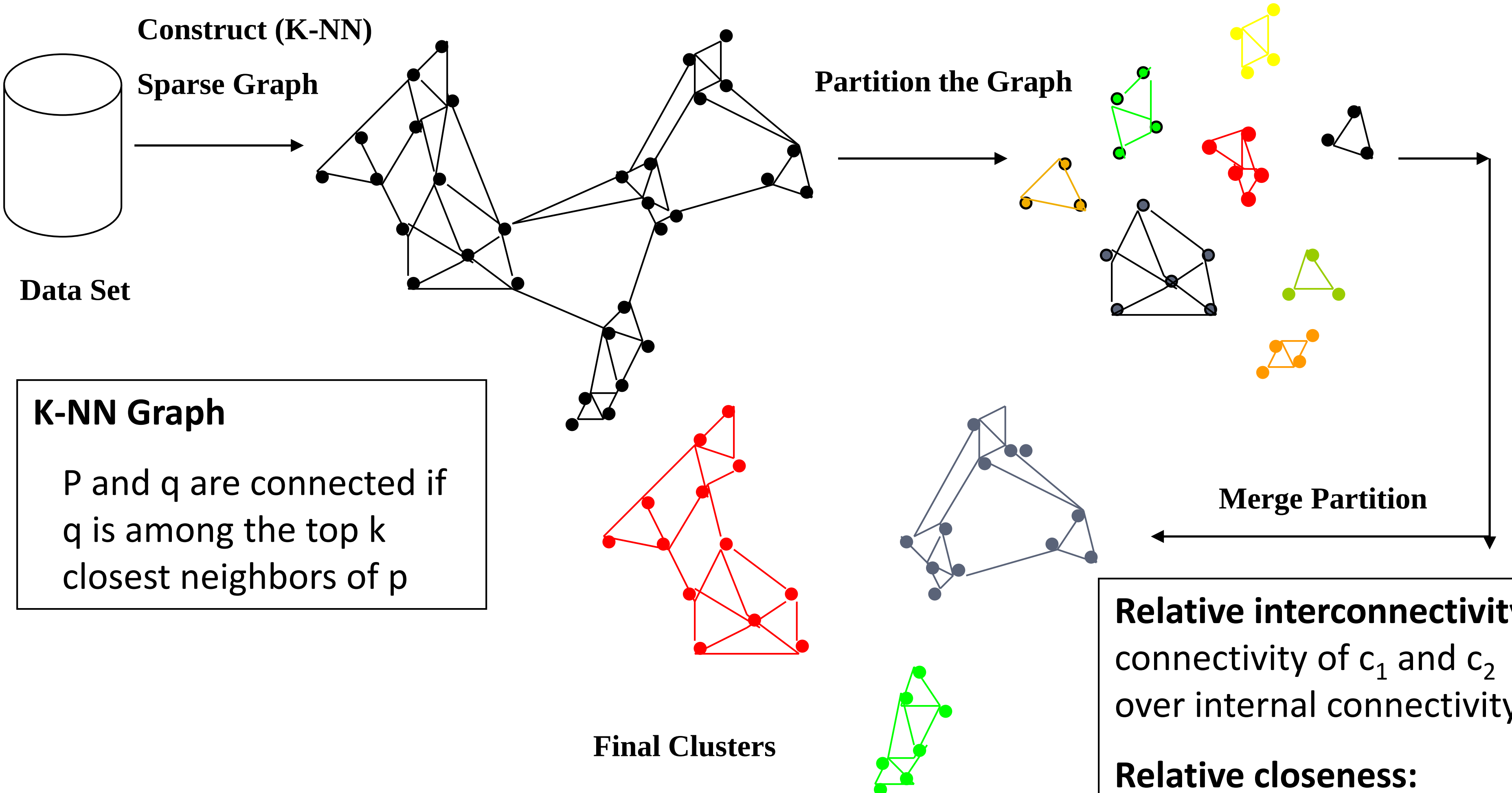
Strength and Limitations of Hierarchical Clustering

- Conceptually simple
- Theoretical properties are well understood
- Major weakness of agglomerative clustering methods
  - Can never undo what was done previously
  - Do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects
- Integration of hierarchical & distance-based clustering
  - CHAMELEON: hierarchical clustering using dynamic modeling

CHAMELEON: Hierarchical Clustering Using Dynamic Modeling (1999)

- CHAMELEON:
- Measures the similarity based on a dynamic model
  - Two clusters are merged only if the *interconnectivity* and *closeness (proximity)* between two clusters are high *relative to* the internal interconnectivity of the clusters and closeness of items within the clusters
- Graph-based, and a two-phase algorithm
  1. Use a graph-partitioning algorithm: cluster objects into a large number of relatively small sub-clusters
  2. Use an agglomerative hierarchical clustering algorithm: find the genuine clusters by repeatedly combining these sub-clusters

Overall Framework of CHAMELEON



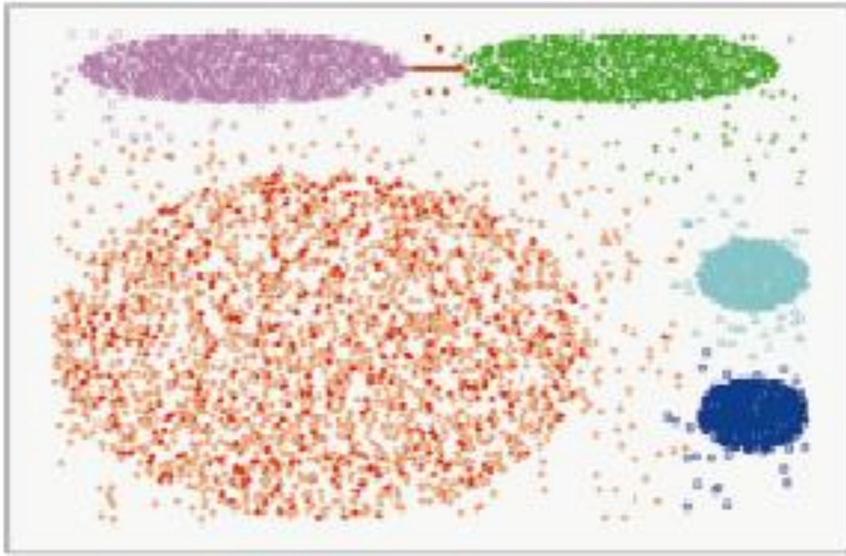
K-NN Graph

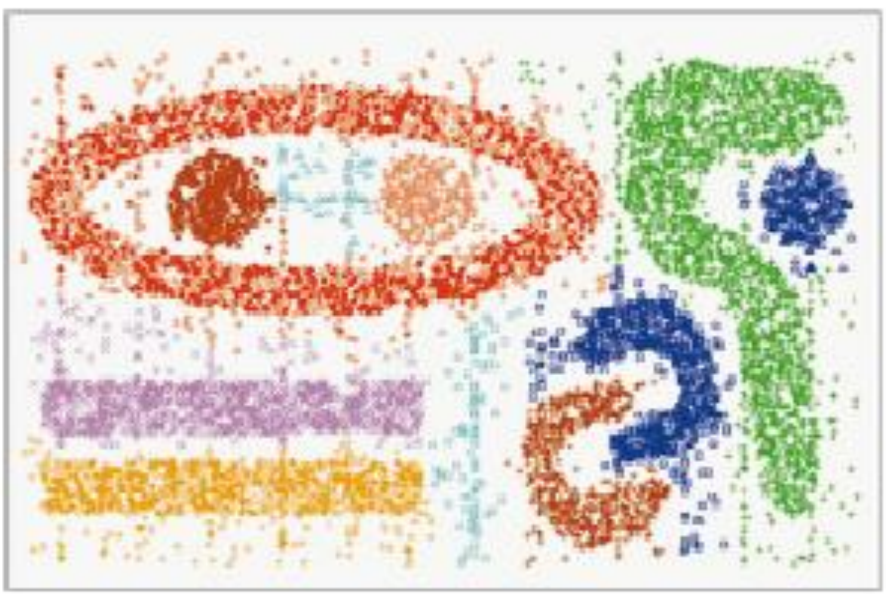
P and q are connected if q is among the top k closest neighbors of p

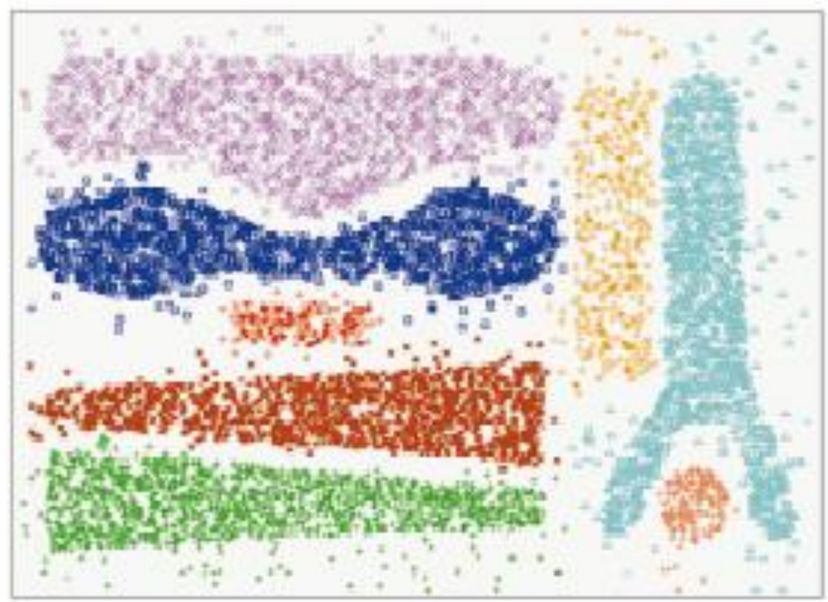
Relative interconnectivity: connectivity of c_1 and c_2 over internal connectivity

Relative closeness: closeness of c_1 and c_2 over internal closeness

CHAMELEON (Clustering Complex Objects)







Summary

- Cluster analysis groups objects based on their similarity and has wide applications
  - We have looked at different clustering algorithms
  - We examined their strengths and weaknesses

Reading List

- Recommended
  - Review Slides!
  - Book: Jiawei Han, Micheline Kamber and Jian Pei, Data Mining - Concepts and Techniques, Morgan Kaufmann, Third Edition, 2011 (or 2nd edition)
    - http://ccs1.hnue.edu.vn/hungtd/DM2012/DataMining_BOOK.pdf
    - Chapter: 7