

The background of the slide is a collage of abstract data visualizations. It features several network graphs with nodes and edges in various colors (red, green, blue, orange). There are also scatter plots with points of different colors (orange, brown, blue) and some heatmaps or density plots. The overall aesthetic is technical and data-driven.

# **Lecture 4. Hierarchical Clustering Methods**

# Lecture 2. Hierarchical Clustering Methods

---

- ❑ Basic Concepts of Hierarchical Algorithms
- ❑ Agglomerative Clustering Algorithms
- ❑ Divisive Clustering Algorithms
- ❑ Extensions to Hierarchical Clustering
- ❑ BIRCH: A Micro-Clustering-Based Approach
- ❑ CURE: Exploring Well-Scattered Representative Points
- ❑ CHAMELEON: Graph Partitioning on the KNN Graph of the Data
- ❑ Probabilistic Hierarchical Clustering
- ❑ Summary



The background of the slide is a complex, abstract composition. It features a dark, muted purple and brown color palette. Overlaid on this are several geometric and data-related elements: a network of thin, light-colored lines forming a mesh or web-like structure; numerous small, semi-transparent circles in shades of green, blue, and orange scattered across the field; and a grid of small white plus signs. In the upper left, there's a faint, horizontal band of more complex, possibly fractal-like patterns. A large, white, angular shape, resembling a stylized letter 'A' or a large triangle, is positioned behind the main text, creating a sense of depth and framing.

# **Session 1: Basic Concepts of Hierarchical Algorithms**

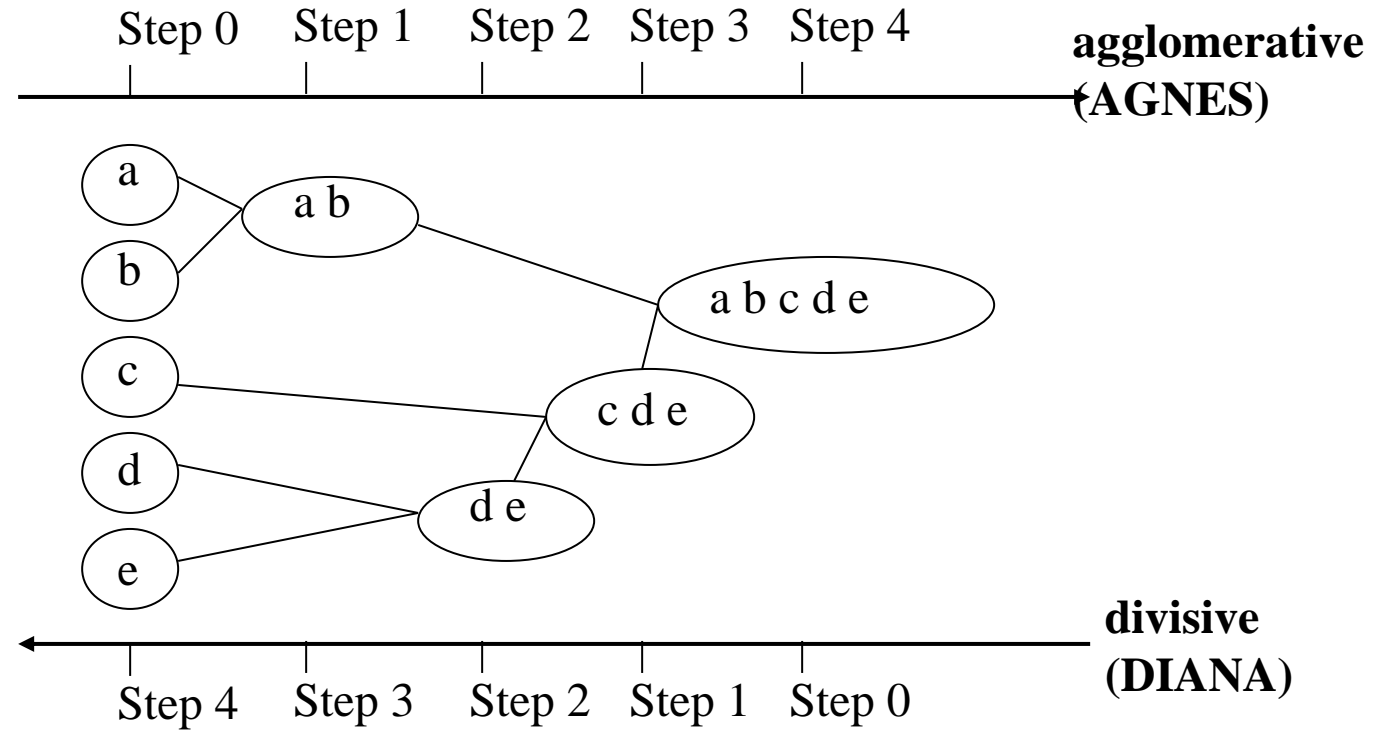
# Hierarchical Clustering: Basic Concepts

## □ Hierarchical clustering

- Generate a clustering hierarchy (drawn as a **dendrogram**)
- Not required to specify **K**, the number of clusters
- More deterministic
- No iterative refinement

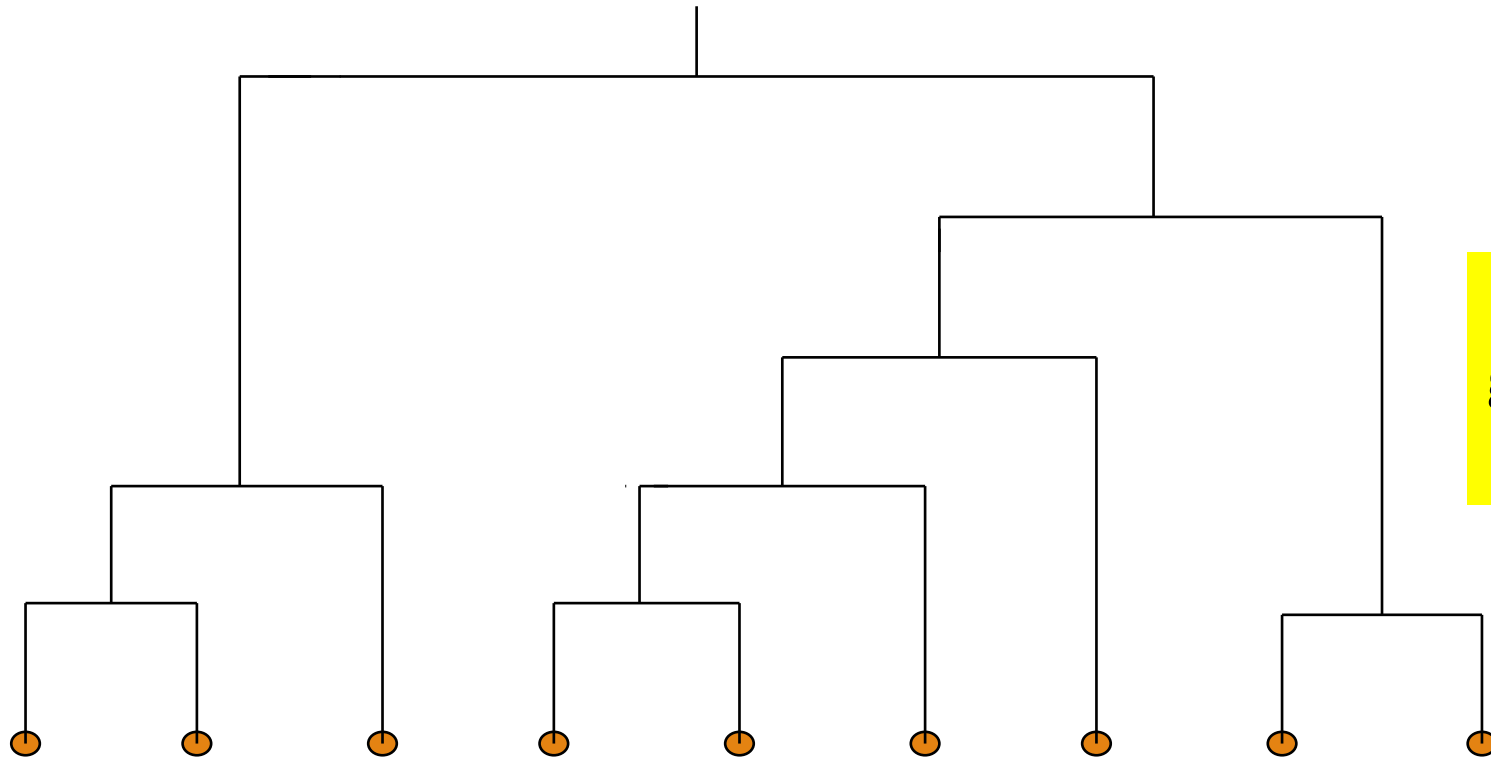
## □ Two categories of algorithms:

- **Agglomerative**: Start with singleton clusters, continuously merge two clusters at a time to build a **bottom-up** hierarchy of clusters
- **Divisive**: Start with a huge macro-cluster, split it continuously into two groups, generating a **top-down** hierarchy of clusters



# Dendrogram: Shows How Clusters are Merged

- ❑ Dendrogram: Decompose a set of data objects into a tree of clusters by multi-level nested partitioning
- ❑ A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster



Hierarchical clustering  
generates a dendrogram  
(a hierarchy of clusters)



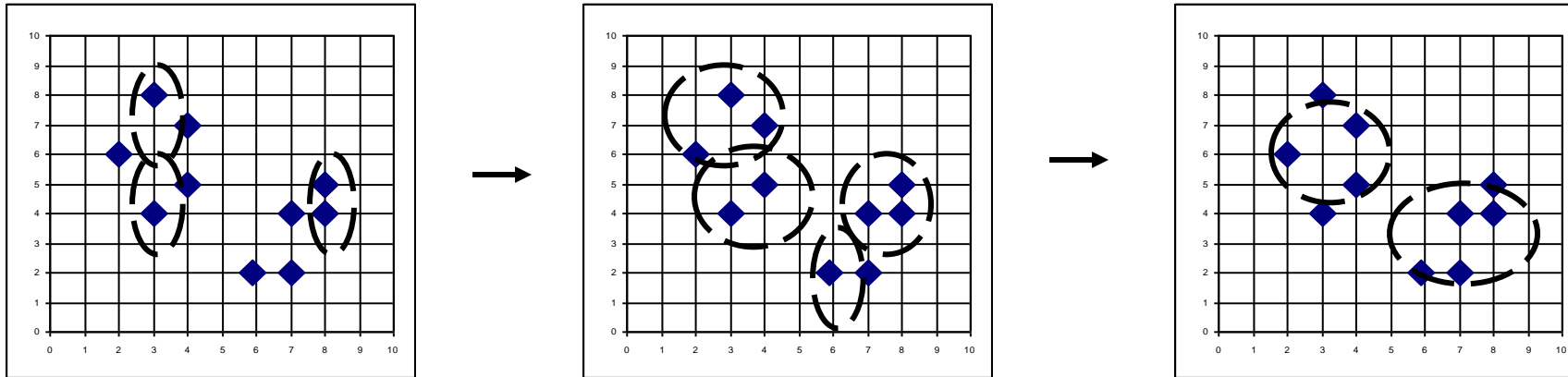


# **Session 2: Agglomerative Clustering Algorithms**

# Agglomerative Clustering Algorithm

- AGNES (AGglomerative NESting) (Kaufmann and Rousseeuw, 1990)

- Use the **single-link** method and the dissimilarity matrix
- Continuously merge nodes that have the least dissimilarity
- Eventually all nodes belong to the same cluster



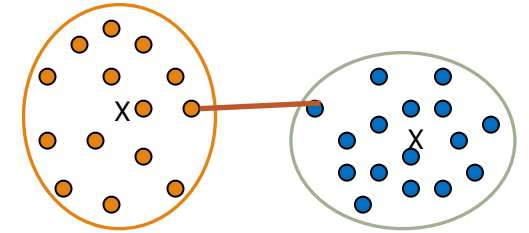
- Agglomerative clustering varies on different similarity measures among clusters

- Single link (nearest neighbor)
- Complete link (diameter)
- Average link (group average)
- Centroid link (centroid similarity)

# Single Link vs. Complete Link in Hierarchical Clustering

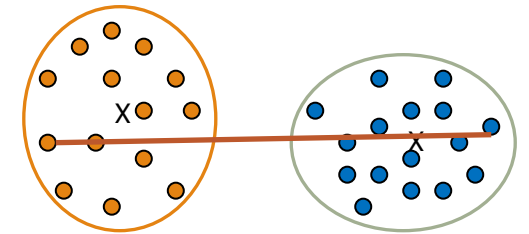
## □ Single link (nearest neighbor)

- The similarity between two clusters is the similarity between their most similar (nearest neighbor) members
- Local similarity-based: Emphasizing more on close regions, ignoring the overall structure of the cluster
- Capable of clustering non-elliptical shaped group of objects
- Sensitive to noise and outliers



## □ Complete link (diameter)

- The similarity between two clusters is the similarity between their most dissimilar members
- Merge two clusters to form one with the smallest diameter
- Nonlocal in behavior, obtaining compact shaped clusters
- Sensitive to outliers





# Agglomerative Clustering: Average vs. Centroid Links

- Agglomerative clustering with **average link**

- Average link:** The average distance between an element in one cluster and an element in the other (i.e., all pairs in two clusters)

- Expensive to compute

- Agglomerative clustering with **centroid link**

- Centroid link:** The distance between the centroids of two clusters

- Group Averaged Agglomerative Clustering (GAAC)**

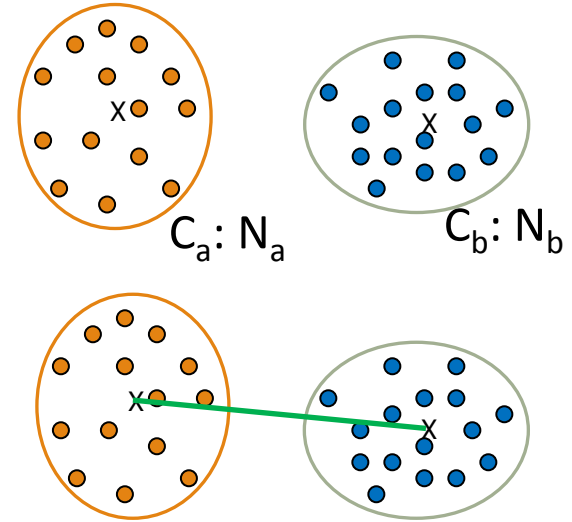
- Let two clusters  $C_a$  and  $C_b$  be merged into  $C_{a \cup b}$ . The new centroid is:

- $N_a$  is the cardinality of cluster  $C_a$ , and  $c_a$  is the centroid of  $C_a$

- The similarity measure for GAAC is the average of their distances

- Agglomerative clustering with **Ward's criterion**

- Ward's criterion:** The increase in the value of the SSE criterion for the clustering obtained by merging them into  $C_a \cup C_b$ : 
$$W(C_{a \cup b}, c_{a \cup b}) - W(C, c) = \frac{N_a N_b}{N_a + N_b} d(c_a, c_b)$$

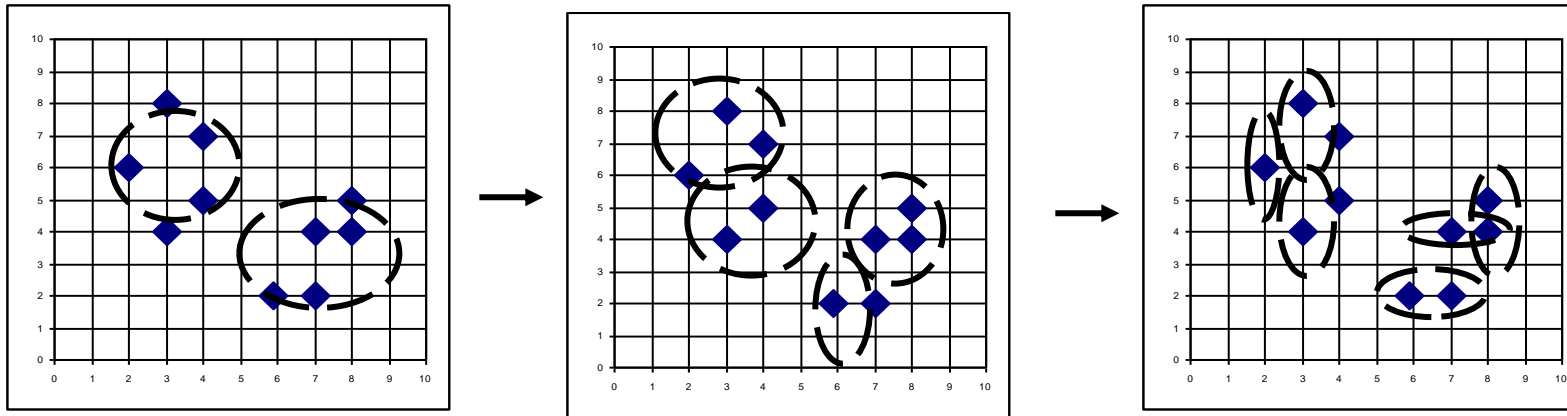




# **Session 3: Divisive Clustering Algorithms**

# Divisive Clustering

- ❑ DIANA (Divisive Analysis) (Kaufmann and Rousseeuw, 1990)
  - ❑ Implemented in some statistical analysis packages, e.g., Splus
- ❑ Inverse order of AGNES: Eventually each node forms a cluster on its own



- ❑ Divisive clustering is a top-down approach
  - ❑ The process starts at the root with all the points as one cluster
  - ❑ It recursively splits the higher level clusters to build the dendrogram
  - ❑ Can be considered as a global approach
  - ❑ More efficient when compared with agglomerative clustering



# More on Algorithm Design for Divisive Clustering

---

- ❑ Choosing which cluster to split
  - ❑ Check the sums of squared errors of the clusters and choose the one with the largest value
- ❑ Splitting criterion: Determining how to split
  - ❑ One may use Ward's criterion to chase for greater reduction in the difference in the SSE criterion as a result of a split
  - ❑ For categorical data, Gini-index can be used
- ❑ Handling the noise
  - ❑ Use a threshold to determine the termination criterion (do not generate clusters that are too small because they contain mainly noises)



# **Session 4: Extensions to Hierarchical Clustering**

# Extensions to Hierarchical Clustering

---

- ❑ Major weaknesses of hierarchical 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
- ❑ Other hierarchical clustering algorithms
  - ❑ BIRCH (1996): Use CF-tree and incrementally adjust the quality of sub-clusters
  - ❑ CURE (1998): Represent a cluster using a set of well-scattered representative points
  - ❑ CHAMELEON (1999): Use graph partitioning methods on the K-nearest neighbor graph of the data



The background of the slide features a complex, abstract design. It includes a grid of small grey plus signs, a network of thin red lines connecting various points, and clusters of green and blue dots. A large, semi-transparent white trapezoidal shape is positioned behind the main title text. On the left side, there is a small inset image showing a dense cluster of orange and red dots with a horizontal bar chart overlaid.

# **Session 5: BIRCH: A Micro-Clustering-Based Approach**

# BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies)

---

- ❑ A multiphase clustering algorithm (Zhang, Ramakrishnan & Livny, SIGMOD'96)
- ❑ Incrementally construct a CF (Clustering Feature) tree, a hierarchical data structure for multiphase clustering
  - ❑ Phase 1: Scan DB to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data)
  - ❑ Phase 2: Use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree
- ❑ Key idea: Multi-level clustering
  - ❑ Low-level micro-clustering: Reduce complexity and increase scalability
  - ❑ High-level macro-clustering: Leave enough flexibility for high-level clustering
- ❑ *Scales linearly*: Find a good clustering with a single scan and improve the quality with a few additional scans

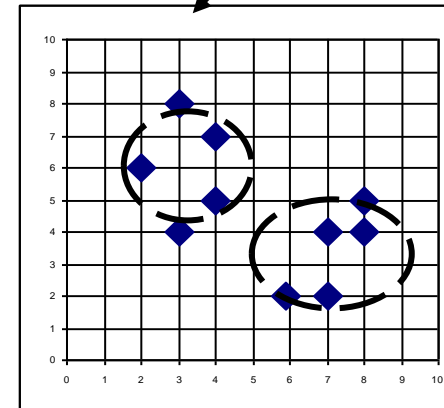
# Clustering Feature Vector in BIRCH

❑ Clustering Feature (CF):  $CF = (N, LS, SS)$

❑  $N$ : Number of data points

❑  $LS$ : linear sum of  $N$  points:  $\sum_{i=1}^N X_i$

❑  $SS$ : square sum of  $N$  points:  $\sum_{i=1}^N X_i^2$



$CF = (5, (16,30), (54,190))$

(3,4)

(2,6)

(4,5)

(4,7)

(3,8)

❑ Clustering feature:

❑ Summary of the statistics for a given sub-cluster: the 0-th, 1st, and 2nd moments of the sub-cluster from the statistical point of view

❑ Registers crucial measurements for computing cluster and utilizes storage efficiently



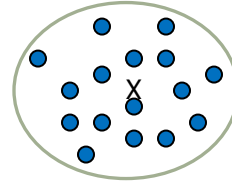
# Measures of Cluster: Centroid, Radius and Diameter

□ Centroid:  $\vec{x}_0$

□ the “middle” of a cluster

□  $n$ : number of points in a cluster

□  $\vec{x}_i$  is the  $i$ -th point in the cluster



$$\vec{x}_0 = \frac{\sum_i^n \vec{x}_i}{n}$$

□ Radius:  $R$

□ Average distance from member objects to the centroid

□ The square root of average distance from any point of the cluster to its centroid

$$R = \sqrt{\frac{\sum_i^n (\vec{x}_i - \vec{x}_0)^2}{n}}$$

□ Diameter:  $D$

□ Average pairwise distance within a cluster

□ The square root of average mean squared distance between all pairs of points in the cluster

$$D = \sqrt{\frac{\sum_i^n \sum_j^n (\vec{x}_i - \vec{x}_j)^2}{n(n-1)}}$$

# The CF Tree Structure in BIRCH

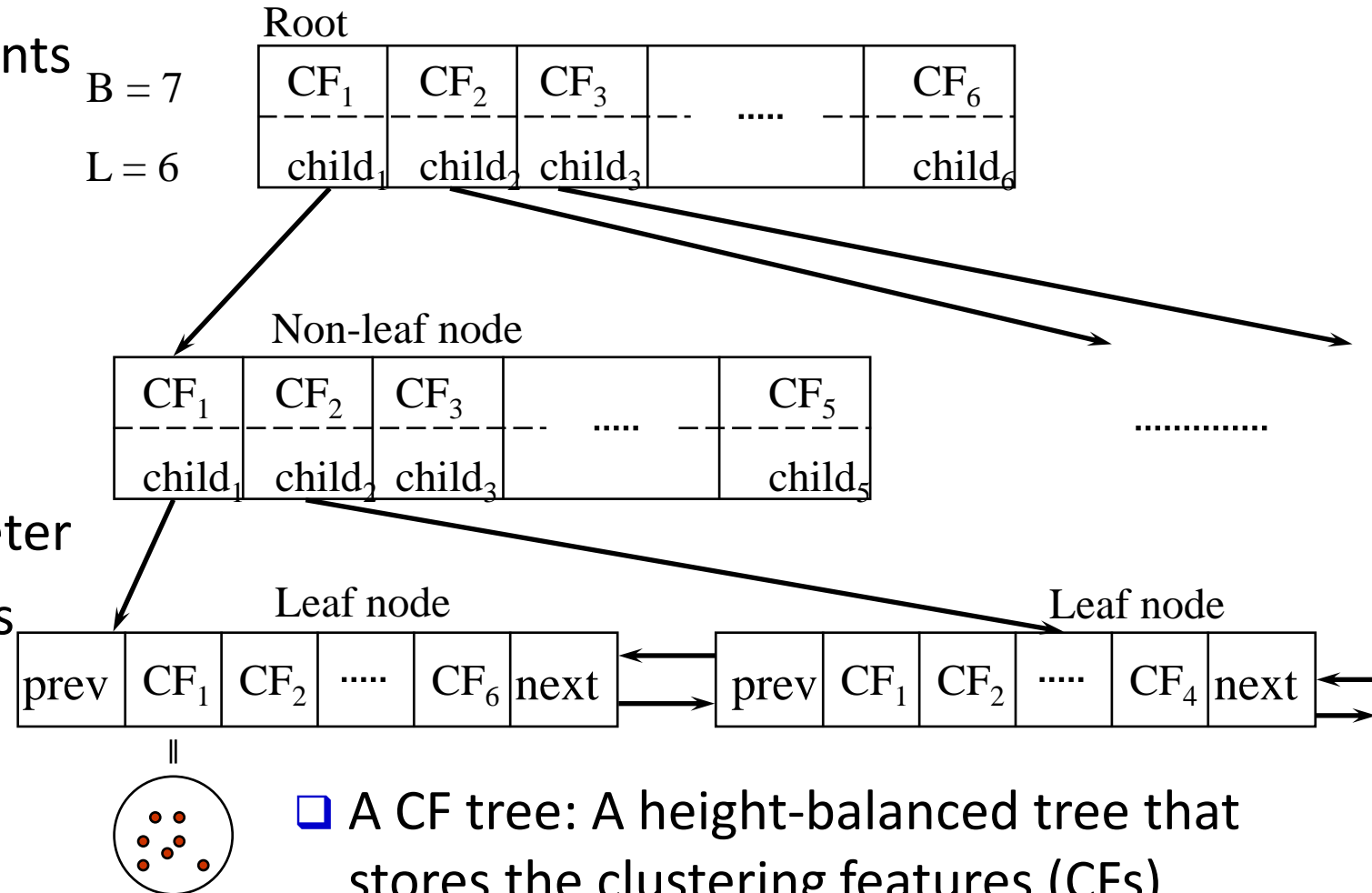
- Incremental insertion of new points (similar to B+-tree)

- For each point in the input

- Find closest leaf entry
- Add point to leaf entry and update CF
- If entry diameter  $>$  max\_diameter
  - split leaf, and possibly parents

- A CF tree has two parameters

- Branching factor: Maximum number of children
- Maximum diameter of sub-clusters stored at the leaf nodes



- A CF tree: A height-balanced tree that stores the clustering features (CFs)
- The non-leaf nodes store sums of the CFs of their children

# BIRCH: A Scalable and Flexible Clustering Method

---

- ❑ An integration of agglomerative clustering with other (flexible) clustering methods
  - ❑ Low-level micro-clustering
    - ❑ Exploring CP-feature and BIRCH tree structure
    - ❑ Preserving the inherent clustering structure of the data
  - ❑ Higher-level macro-clustering
    - ❑ Provide sufficient flexibility for integration with other clustering methods
- ❑ Impact to many other clustering methods and applications
- ❑ Concerns
  - ❑ Sensitive to insertion order of data points
  - ❑ Due to the fixed size of leaf nodes, clusters may not be so natural
  - ❑ Clusters tend to be spherical given the radius and diameter measures

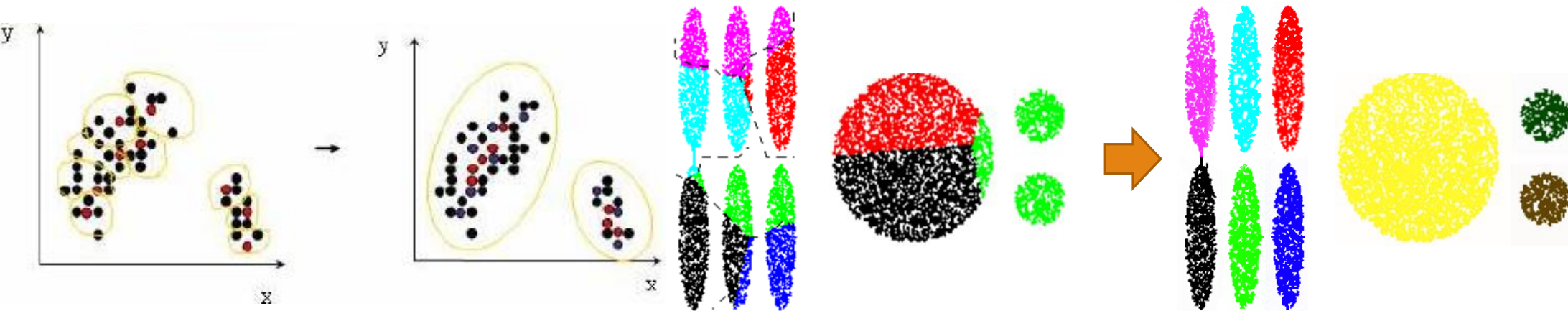




# **Session 6: CURE: Clustering Using Well-Scattered Representatives**

# CURE: Clustering Using Representatives

- **CURE** (Clustering Using REpresentatives) (S. Guha, R. Rastogi, and K. Shim, 1998)
  - Represent a cluster using a set of well-scattered representative points
- Cluster distance: Minimum distance between the representative points chosen
  - This incorporates features of both single link and average link
- Shrinking factor  $\alpha$ : The points are shrunk towards the centroid by a factor  $\alpha$ 
  - Far away points are shrunk more towards the center: More robust to outliers
- Choosing scattered points helps CURE capture clusters of arbitrary shapes



Courtesy: Kyuseok Shim@SNU.KR





# **Session 7: CHAMELEON: Graph Partitioning on the KNN Graph of the Data**

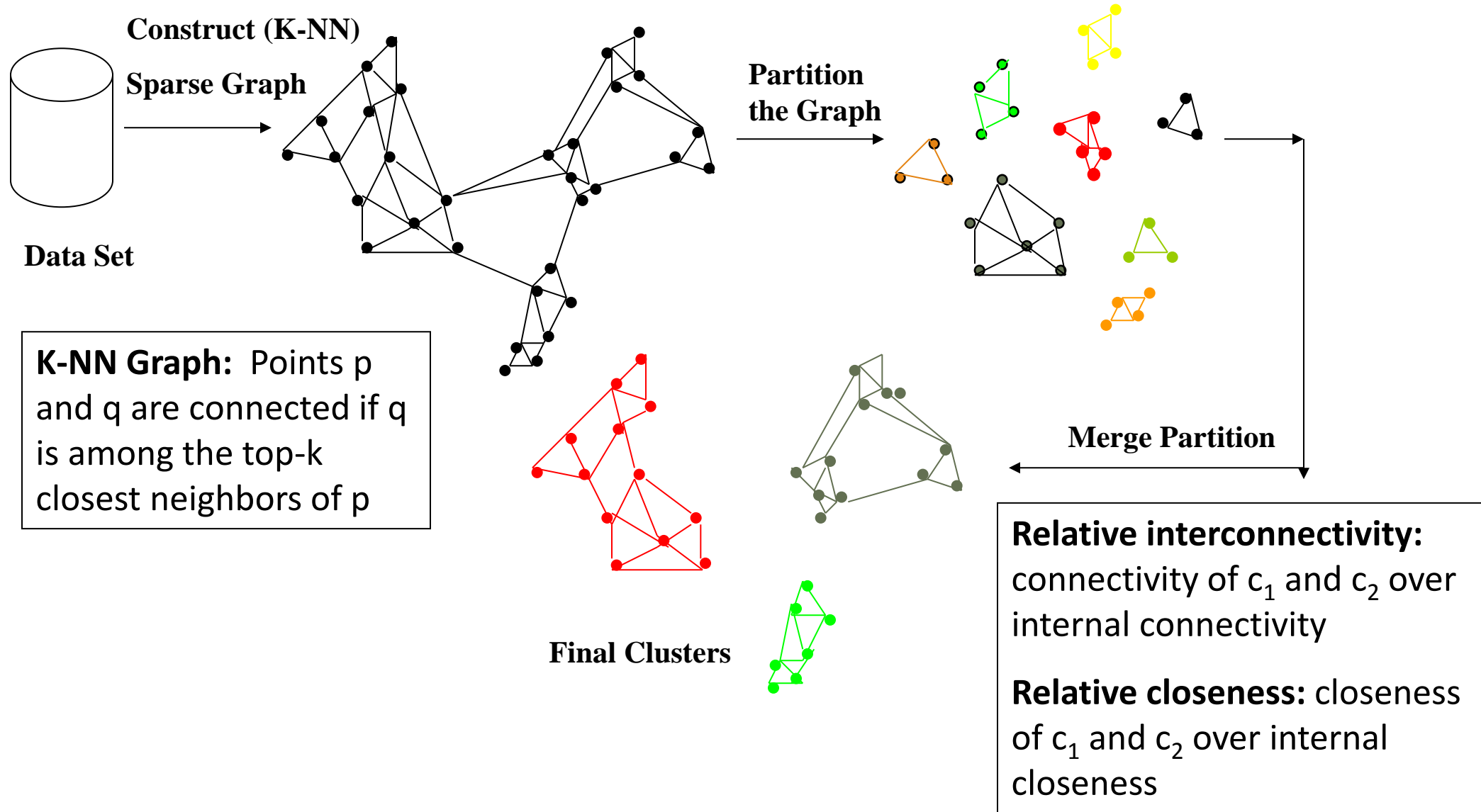
# CHAMELEON: Hierarchical Clustering Using Dynamic Modeling

---

- ❑ CHAMELEON: A graph partitioning approach (G. Karypis, E. H. Han, and V. Kumar, 1999)
- ❑ 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
- ❑ A graph-based, 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

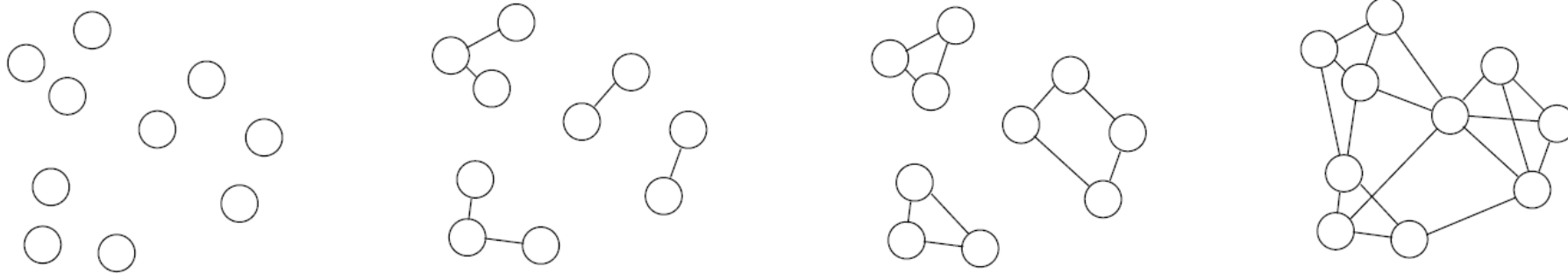


# Overall Framework of CHAMELEON



# KNN Graphs and Interconnectivity

- K-nearest neighbor (KNN) graphs from an original data in 2D:



(a) Original Data in 2D

(b) 1-nearest neighbor graph

(c) 2-nearest neighbor graph

(d) 3-nearest neighbor graph

- $EC_{\{C_i, C_j\}}$ : The absolute interconnectivity between  $C_i$  and  $C_j$ :

- *The sum of the weight of the edges that connect vertices in  $C_i$  to vertices in  $C_j$*

- Internal interconnectivity of a cluster  $C_i$ : *The size of its min-cut bisector  $EC_{C_i}$  (i.e., the weighted sum of edges that partition the graph into two roughly equal parts)*

- Relative Interconnectivity (RI):

$$RI(C_i, C_j) = \frac{|EC_{\{C_i, C_j\}}|}{\frac{|EC_{C_i}| + |EC_{C_j}|}{2}}$$

# Relative Closeness & Merge of Sub-Clusters

---

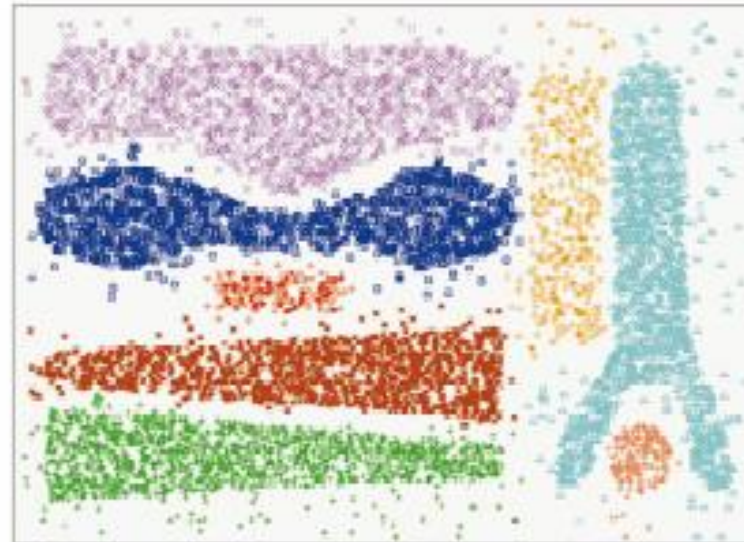
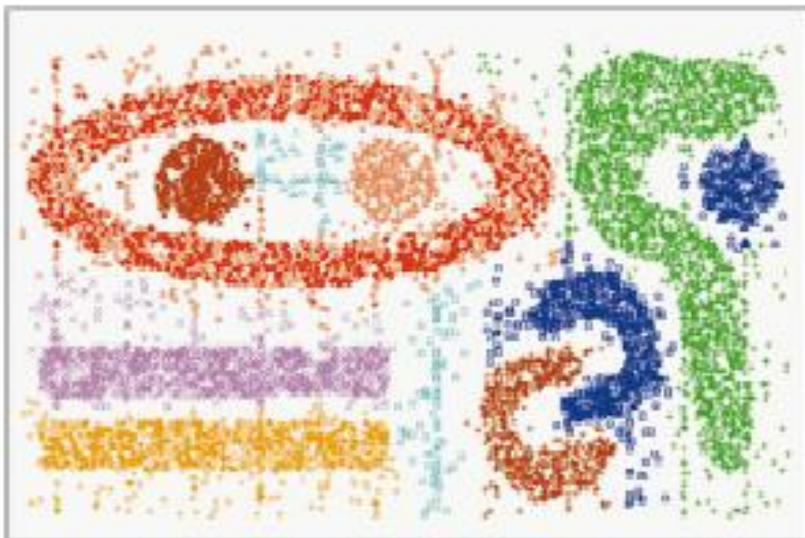
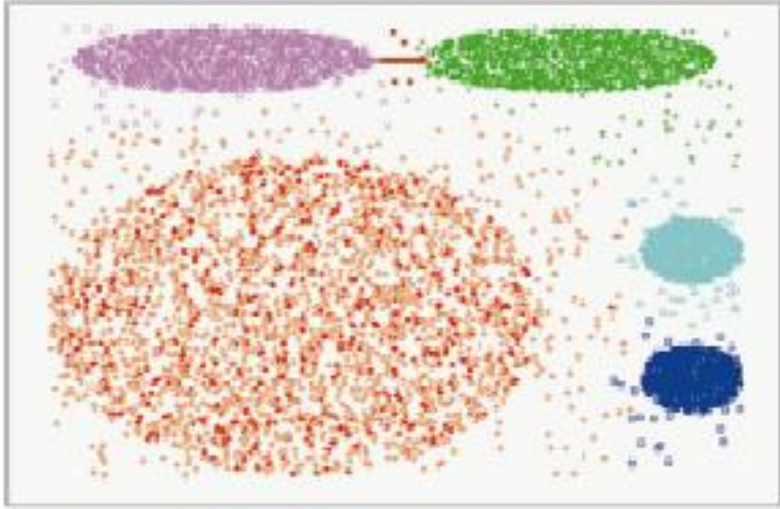
- ❑ **Relative closeness** between a pair of clusters  $C_i$  and  $C_j$ : *The absolute closeness between  $C_i$  and  $C_j$  normalized w.r.t. the internal closeness of the two clusters  $C_i$  and  $C_j$*

$$RC(C_i, C_j) = \frac{\bar{S}_{EC\{C_i, C_j\}}}{\frac{|C_i|}{|C_i|+|C_j|}\bar{S}_{EC_{C_i}} + \frac{|C_j|}{|C_i|+|C_j|}\bar{S}_{EC_{C_j}}}$$

- ❑ where  $\bar{S}_{EC_{C_i}}$  and  $\bar{S}_{EC_{C_j}}$  are the average weights of the edges that belong to the min-cut bisector of clusters  $C_i$  and  $C_j$ , respectively, and  $\bar{S}_{EC\{C_i, C_j\}}$  is the average weight of the edges that connect vertices in  $C_i$  to vertices in  $C_j$
- ❑ **Merge Sub-Clusters:**
  - ❑ Merges only those pairs of clusters whose RI and RC are both above some user-specified thresholds
  - ❑ Merge those maximizing the function that combines RI and RC

# CHAMELEON: Clustering Complex Objects

---



CHAMELEON is capable to generate quality clusters at clustering complex objects



The background of the slide is a collage of abstract data visualizations. It features several network graphs with nodes and edges in various colors (red, green, blue, orange). There are also scatter plots with points of different colors (green, blue, orange, purple) and some blurred, low-resolution images of what might be biological structures or data clusters. The overall aesthetic is technical and data-driven.

# **Session 8: Probabilistic Hierarchical Clustering**

# Probabilistic Hierarchical Clustering

---

- ❑ Algorithmic hierarchical clustering
  - ❑ Nontrivial to choose a good distance measure
  - ❑ Hard to handle missing attribute values
  - ❑ Optimization goal not clear: heuristic, local search
- ❑ Probabilistic hierarchical clustering
  - ❑ Use probabilistic models to measure distances between clusters
  - ❑ Generative model: Regard the set of data objects to be clustered as a sample of the underlying data generation mechanism to be analyzed
  - ❑ Easy to understand, same efficiency as algorithmic agglomerative clustering method, can handle partially observed data
- ❑ In practice, assume the generative models adopt common distribution functions, e.g., Gaussian distribution or Bernoulli distribution, governed by parameters

# Generative Model

- Given a set of 1-D points  $X = \{x_1, \dots, x_n\}$  for clustering analysis & assuming they are generated by a Gaussian distribution:

$$\mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- The probability that a point  $x_i \in X$  is generated by the model:

$$P(x_i | \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i - \mu)^2}{2\sigma^2}}$$

- The likelihood that  $X$  is generated by the model:

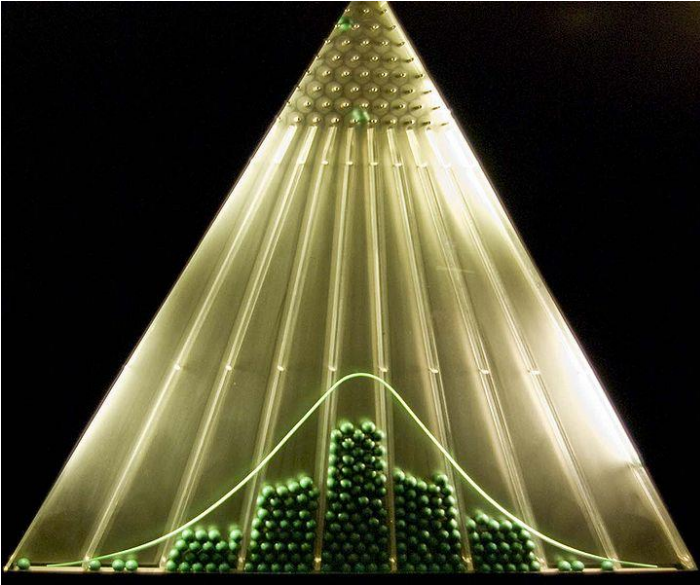
$$L(\mathcal{N}(\mu, \sigma^2) : X) = P(X | \mu, \sigma^2) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i - \mu)^2}{2\sigma^2}}$$

- The task of learning the generative model: find the parameters  $\mu$  and  $\sigma^2$  such that

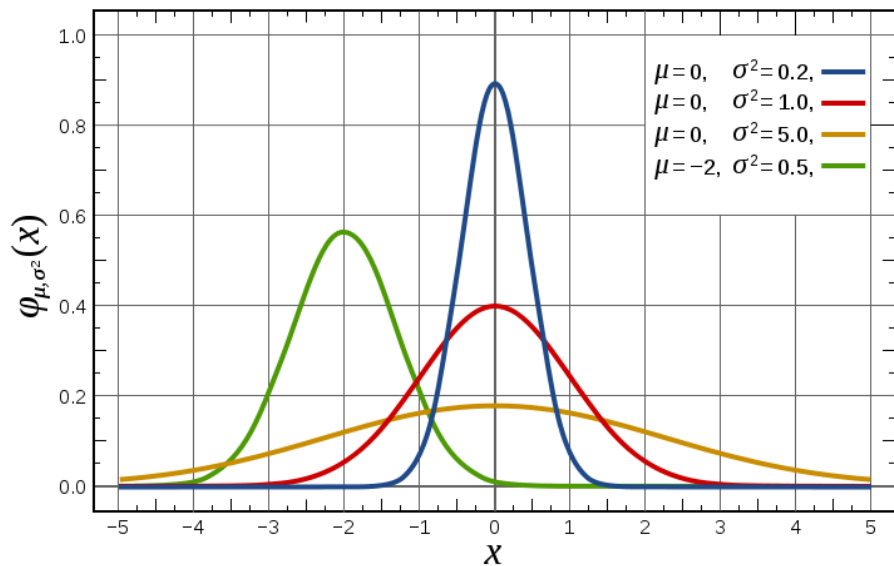
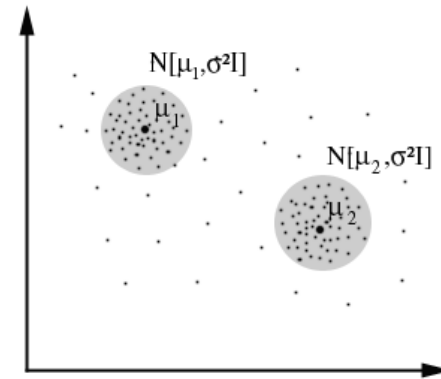
the maximum likelihood

$$\mathcal{N}(\mu_0, \sigma_0^2) = \arg \max \{L(\mathcal{N}(\mu, \sigma^2) : X)\}$$

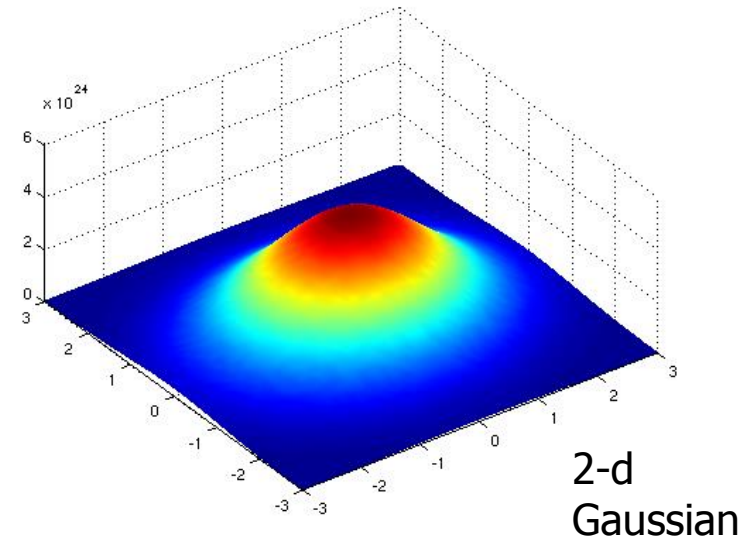
# Gaussian Distribution



Bean  
machine:  
drop ball  
with pins



1-d  
Gaussian



2-d  
Gaussian

From wikipedia and <http://home.dei.polimi.it>



# A Probabilistic Hierarchical Clustering Algorithm

- For a set of objects partitioned into  $m$  clusters  $C_1, \dots, C_m$ , the quality can be measured by,

$$Q(\{C_1, \dots, C_m\}) = \prod_{i=1}^m P(C_i)$$

where  $P()$  is the maximum likelihood

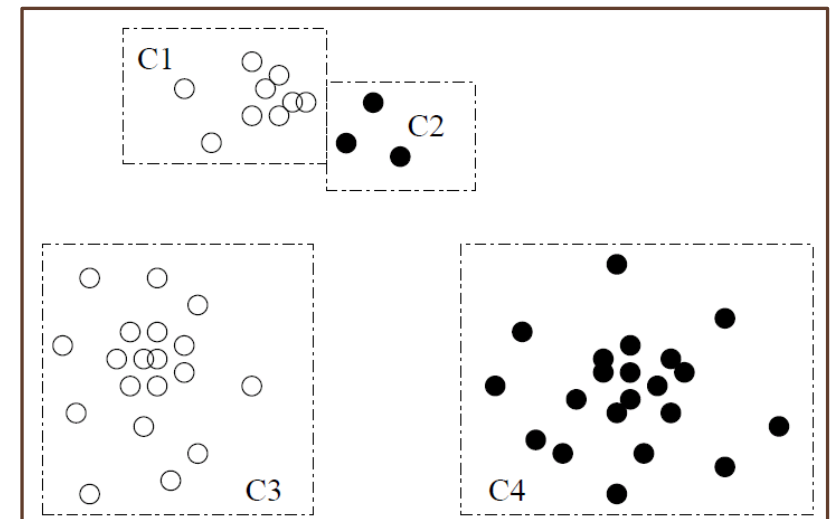
- If we merge two clusters  $C_{j1}$  and  $C_{j2}$  into a cluster  $C_{j1} \cup C_{j2}$ , the change in quality of the overall clustering is

$$\begin{aligned} & Q(\{C_1, \dots, C_m\} - \{C_{j1}, C_{j2}\} \cup \{C_{j1} \cup C_{j2}\}) - Q(\{C_1, \dots, C_m\}) \\ = & \frac{\prod_{i=1}^m P(C_i) \cdot P(C_{j1} \cup C_{j2})}{P(C_{j1})P(C_{j2})} - \prod_{i=1}^m P(C_i) \\ = & \prod_{i=1}^m P(C_i) \left( \frac{P(C_{j1} \cup C_{j2})}{P(C_{j1})P(C_{j2})} - 1 \right) \end{aligned}$$

- Distance between clusters  $C_1$  and  $C_2$ :

$$\text{dist}(C_i, C_j) = -\log \frac{P(C_1 \cup C_2)}{P(C_1)P(C_2)}$$

- If  $\text{dist}(C_i, C_j) < 0$ , merge  $C_i$  and  $C_j$



The background of the slide is a complex, abstract composition. It features a central white banner with a subtle geometric pattern of thin lines and small plus signs. This banner is flanked by two large, overlapping triangular shapes in a light grayish-blue color. The entire slide is framed by a dark, reddish-brown border. Within this border, there are several geometric and data-like elements: a network of thin red lines connecting green dots, a grid of small white plus signs, and a series of purple arrows pointing in various directions. In the bottom-left corner, there is a small, detailed image of a galaxy cluster with orange and red hues, overlaid with a grid of plus signs and a horizontal band of pink and white squares.

# Session 9: Summary

# Summary: Hierarchical Clustering Methods

---

- ❑ Basic Concepts of Hierarchical Algorithms
- ❑ Agglomerative Clustering Algorithms
- ❑ Divisive Clustering Algorithms
- ❑ Extensions to Hierarchical Clustering
- ❑ BIRCH: A Micro-Clustering-Based Approach
- ❑ CURE: Exploring Well-Scattered Representative Points
- ❑ CHAMELEON: Graph Partitioning on the KNN Graph of the Data
- ❑ Probabilistic Hierarchical Clustering
- ❑ Summary

# Recommended Readings

---

- ❑ A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Prentice Hall, 1988
- ❑ L. Kaufman and P. J. Rousseeuw. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, 1990
- ❑ T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: An Efficient Data Clustering Method for Very Large Databases. SIGMOD'96
- ❑ S. Guha, R. Rastogi, and K. Shim. Cure: An Efficient Clustering Algorithm for Large Databases. SIGMOD'98
- ❑ G. Karypis, E.-H. Han, and V. Kumar. CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling. *COMPUTER*, 32(8): 68-75, 1999.
- ❑ Jiawei Han, Micheline Kamber, and Jian Pei. Data Mining: Concepts and Techniques. Morgan Kaufmann, 3<sup>rd</sup> ed. , 2011 (Chap. 10)
- ❑ C. K. Reddy and B. Vinzamuri. A Survey of Partitional and Hierarchical Clustering Algorithms, in (Chap. 4) Aggarwal and Reddy (eds.), Data Clustering: Algorithms and Applications. CRC Press, 2014
- ❑ M. J. Zaki and W. Meira, Jr.. Data Mining and Analysis: Fundamental Concepts and Algorithms. Cambridge Univ. Press, 2014