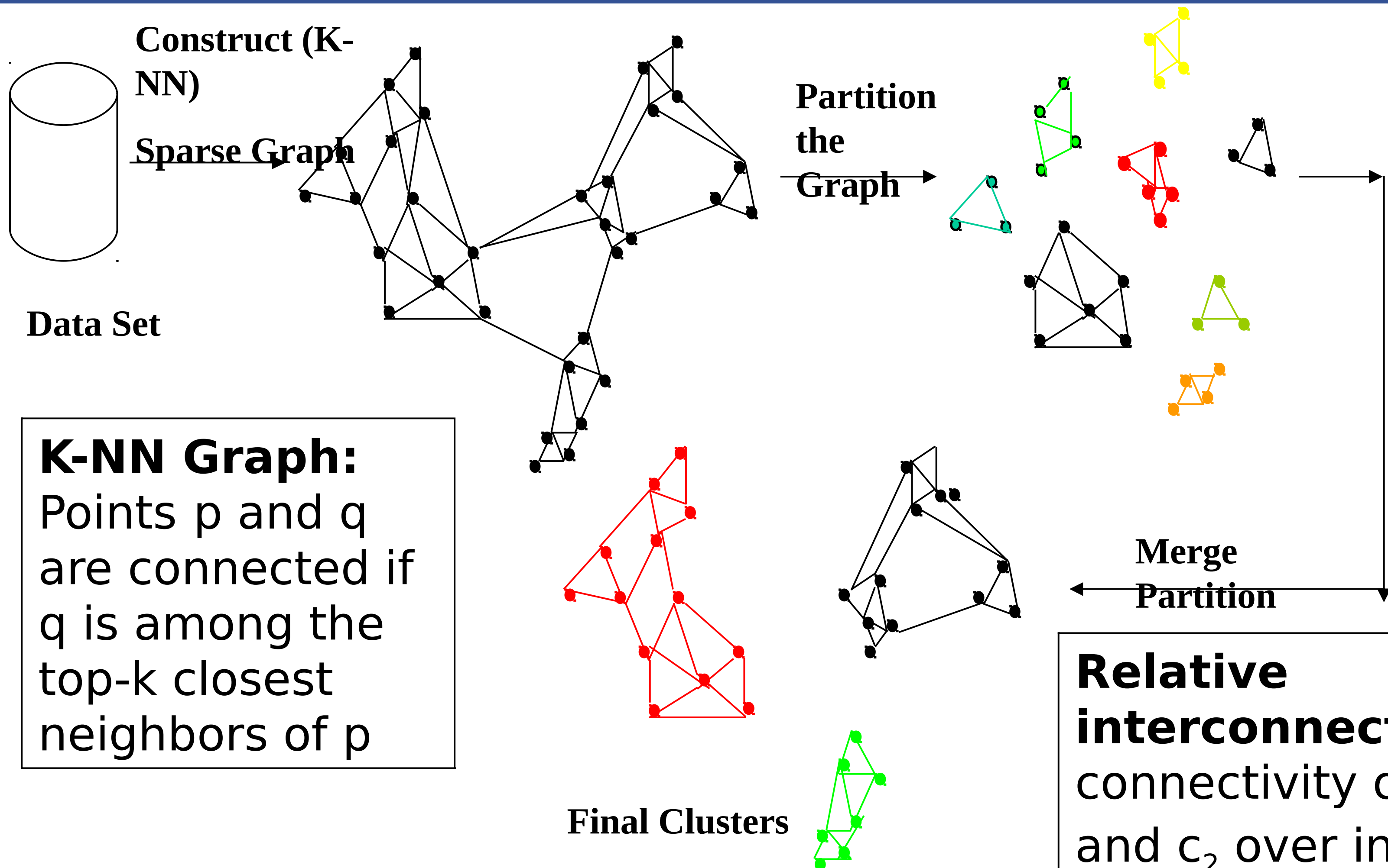
CHAMELEON: Hierarchical Clustering

CHAMELEON: Hierarchical Clustering Using Dynamic Modeling

- CHAMELEON: A graph partitioning approach that uses dynamic model to determine the similarity between pair of clusters.
- Cluster similarity is assessed based on
  - How well connected objects within the cluster
  - The proximity of clusters.
- Two clusters are merged only if the interconnectivity and closeness (proximity) between two clusters are high.
- Chameleon adapt to internal characteristics of the clusters being merged

Overall Framework of CHAMELEON



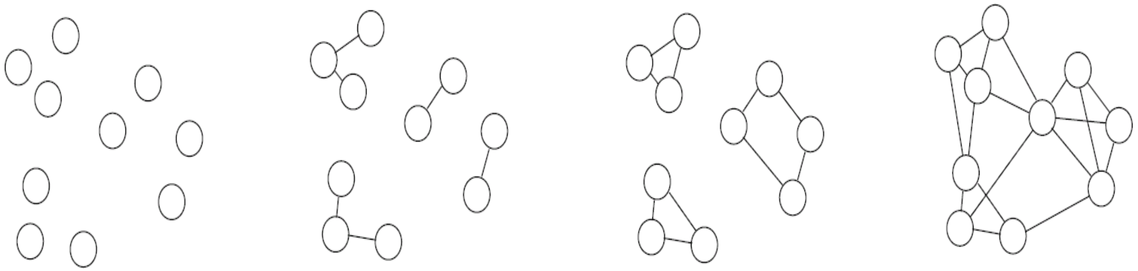
K-NN Graph: Points p and q are connected if q is among the top-k closest neighbors of p

Relative interconnectivity: connectivity of c_1 and c_2 over internal connectivity

Relative closeness:

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

- Each vertex of the graph represents a data object there exists an edge between vertices if one object is among k-most similar objects to the other.

- The edges are weighted to reflect the similarity between objects.

CHAMELEON: Hierarchical Clustering Using Dynamic Modeling

- A graph-based, two-phase algorithm
  - **Use a graph-partitioning algorithm:** Cluster objects into a large number of relatively small sub-clusters
  - **Use an agglomerative hierarchical clustering algorithm:** Find the genuine clusters by repeatedly combining these sub-clusters

CHAMELEON: Partitioning the Graph

- Uses a graph-partitioning algorithm to partition k-nearest neighbor graph into large number of relatively small sub-clusters.
- The cluster C is partitioned into subclusters C_i and Cj so as to minimize the weight of the edges that would be cut hence C be bisected into C_i and Cj.
- It asses the absolute interconnectivity between clusters C_i and Cj.
- $EC_{\{Ci ,Cj\}}$: 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

CHAMELEON: Merging of Sub-Clusters

- Uses Agglomerative hierarchical clustering algorithm that iteratively merges subclusters based on their similarity.
- More similar subclusters are made based on the account of their **relative interconnectivity (RI)** and their **relative closeness of the clusters (RC).**
- **Relative Interconnectivity**(RI): $EC_{\{Ci,Cj\}}$:The absolute interconnectivity between C_i and C_j normalized with respect to the internal interconnectivity of two clusters C_i and C_j.
- **Internal Interconnectivity:** The size of the min-cut bisector ECci, the weighted sum of the edges that partition the graph into two roughly equal parts

CHAMELEON: Merging of Sub-Clusters

Relative Interconnectivity (RI): $EC_{\{Ci\ ,Cj\ \}}$:

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

Absolute interconnectivity defined for cluster C_i and C_j normalized by the average of the respective internal interconnectivities

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

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

where $\overline{S}_{EC_{C_i}}$ and $\overline{S}_{EC_{C_j}}$ are the average weights of the edges that belong to the min-cut bisector of clusters C_i and C_j , respectively, and is the average $\overline{S}_{EC_{\{C_i,C_j\}}}$ weight of the edges that connect vertices in C_i to vertices C_j

Relative Closeness & Merge of Sub-Clusters

- ***Merge Sub-Clusters:***
  - Merges only those pairs of clusters whose RI and RC are both above some user-specified thresholds
- *Discovers arbitrarily shaped clusters of high quality*
- *The processing cost for high-dimensional data may require $O(n^2)$ time for n objects in worst case.*

CHAMELEON: Clustering Complex Objects