Data Mining
Cluster Analysis: Advanced Concepts and Algorithms

Dr. Meng Qu
Rutgers University

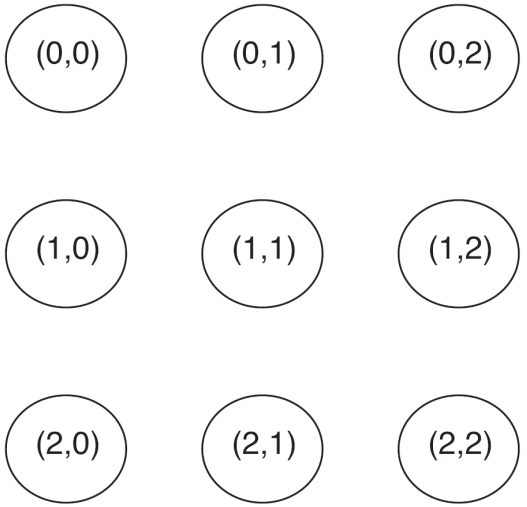
THE STATE UNIVERSITY OF NEW JERSEY
RUTGERS

Outline

- Prototype-based
  - Fuzzy c-means
  - Mixture Model Clustering
  - Self-Organizing Maps
- Density-based
  - Grid-based clustering
  - Subspace clustering
- Graph-based
  - Chameleon
  - Jarvis-Patrick
  - Shared Nearest Neighbor (SNN)
- Characteristics of Clustering Algorithms

SOM: Self-Organizing Maps

- Self-organizing maps (SOM)
  - Centroid based clustering scheme
  - Like K-means, a fixed number of clusters are specified
  - However, the spatial relationship of clusters is also specified, typically as a grid
  - Points are considered one by one
  - Each point is assigned to the closest centroid
  - Other centroids are updated based on their nearness to the closest centroid



Kohonen, Teuvo, and Self-Organizing Maps. "Springer series in information sciences." *Self-organizing maps* 30 (1995).

SOM: Self-Organizing Maps

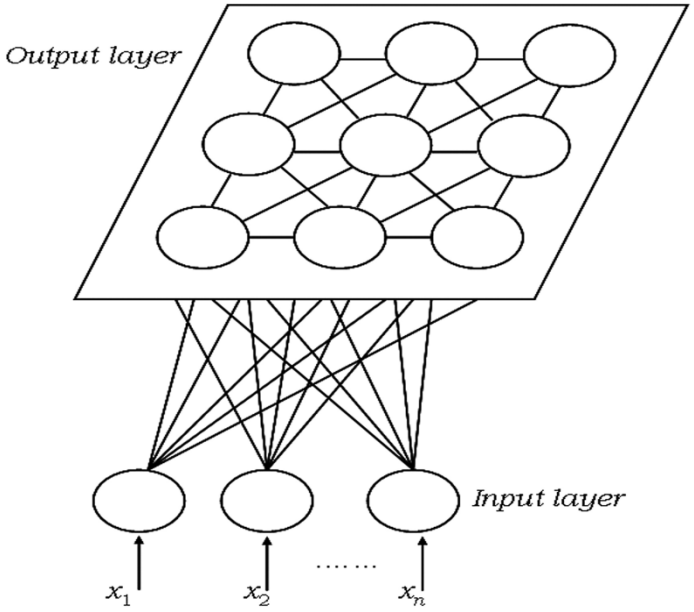
Algorithm 9.3 Basic SOM Algorithm.

1: Initialize the centroids.
2: **repeat**
3: Select the next object.
4: Determine the closest centroid to the object.
5: Update this centroid and the centroids that are close, i.e., in a specified neighborhood.
6: **until** The centroids don't change much or a threshold is exceeded.
7: Assign each object to its closest centroid and return the centroids and clusters.

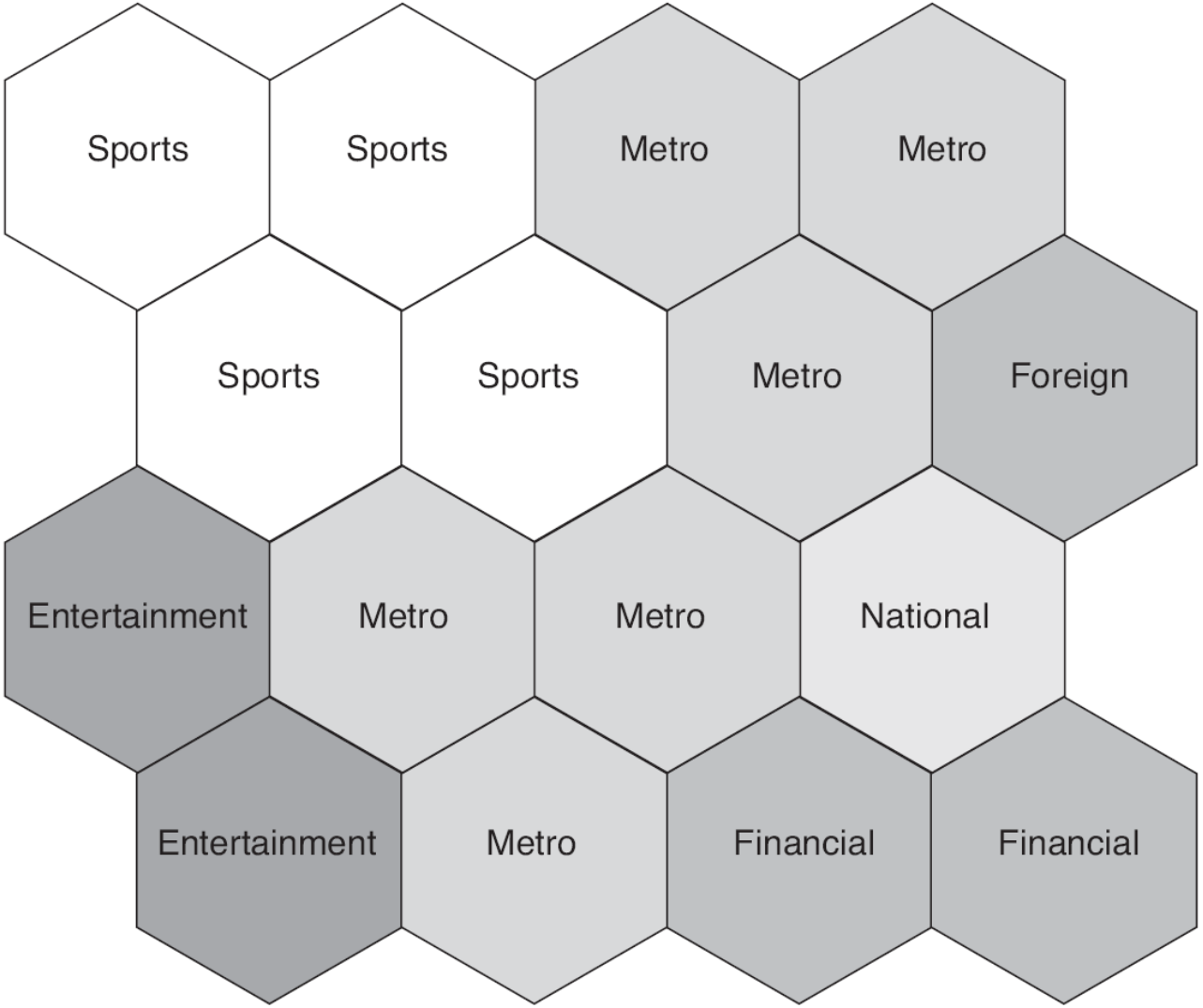
- Updates are weighted by distance
  - Centroids farther away are affected less
- The impact of the updates decreases with each time
  - At some point the centroids will not change much

SOM: Self-Organizing Maps

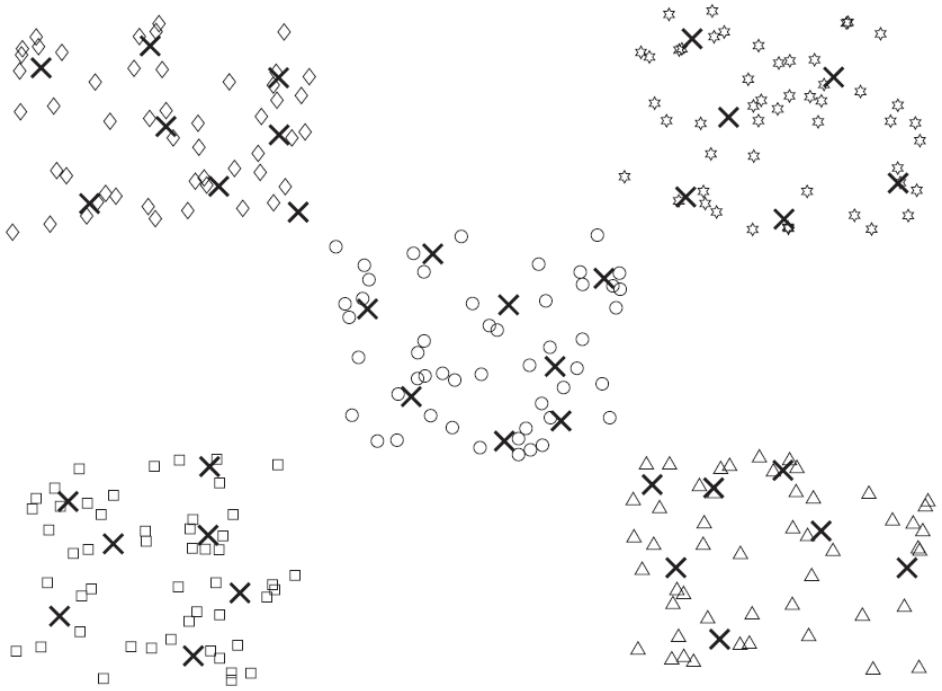
- SOM can be viewed as a type of dimensionality reduction
- If a two-dimensional grid is used, the results can be visualized



SOM Clusters of LA Times Document Data



Another SOM Example: 2D Points



(a) Distribution of SOM reference vectors (**X**'s) for a two-dimensional point set.

diamond	diamond	diamond	hexagon	hexagon	hexagon
diamond	diamond	diamond	circle	hexagon	hexagon
diamond	diamond	circle	circle	circle	hexagon
square	square	circle	circle	triangle	triangle
square	square	circle	circle	triangle	triangle
square	square	square	triangle	triangle	triangle

(b) Classes of the SOM centroids.

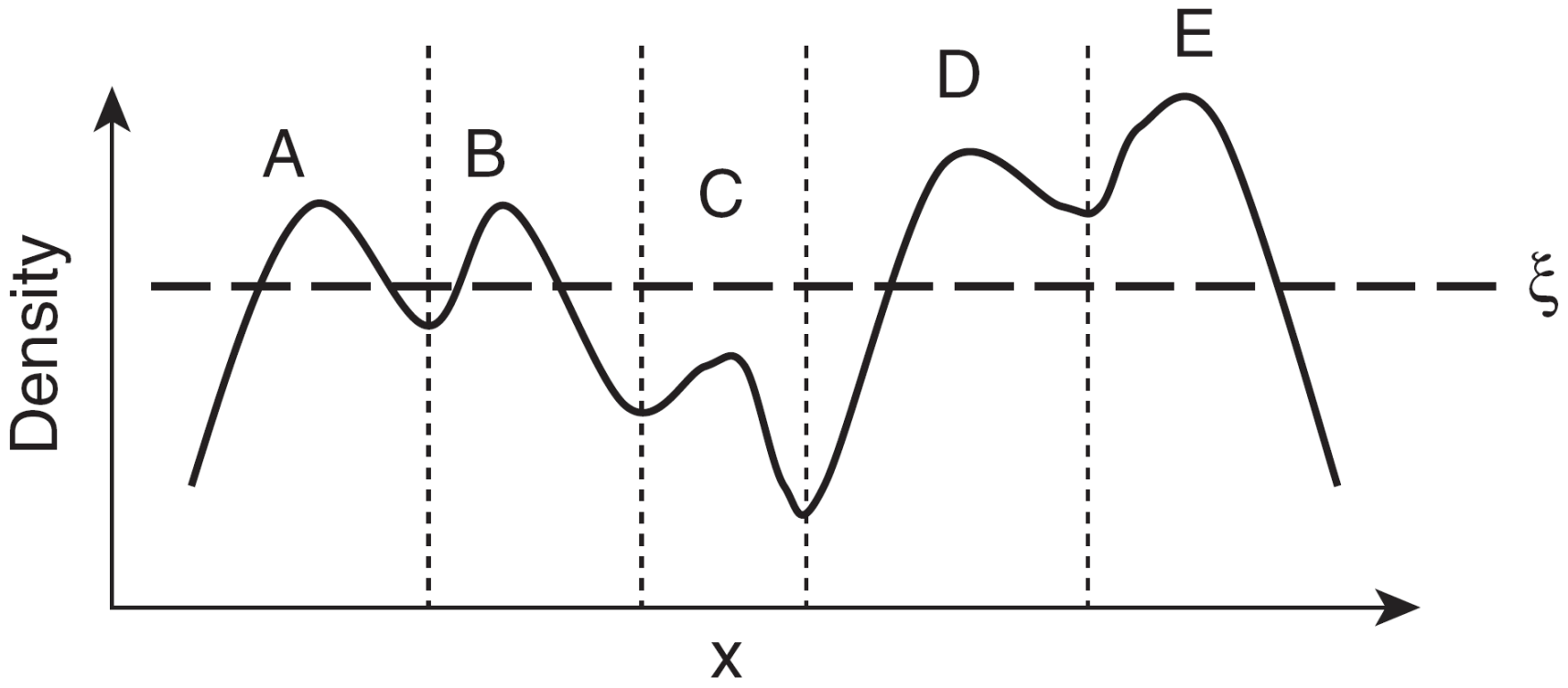
Issues with SOM

- Computational complexity
- Locally optimal solution
- Grid is somewhat arbitrary

Denclue (DENsity CLUstering)

- DENCLUE is a density based clustering approach that models the overall density of a set of points as the sum of influence functions associated with each point.
- The resulting overall density function will have local peaks.
- We can use the local peaks to define clusters.
- For each data point, a hill climbing procedure find the nearest peak associated with that point.
- The set of all data points associated with a particular peak (local density attractor) becomes as cluster.

DENCLUE Algorithm



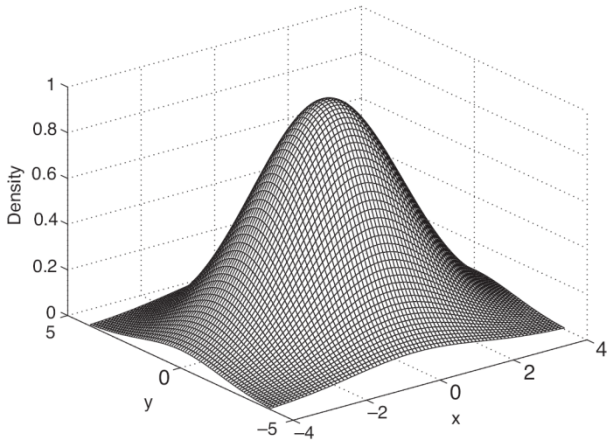
DENCLUE Algorithm

- Find the density function
- Identify local maxima (density attractors)
- Assign each point to the density attractor
  - Follow direction of maximum increase in density
- Define clusters as groups consisting of points associated with density attractor
- Discard clusters whose density attractor has a density less than a user specified minimum, ξ
- Combine clusters connected by paths of points that are connected by points with density above ξ

Denclue (DENsity CLUstering)

- Based on the notion of kernel-density estimation
  - Contribution of each point to the density is given by an influence or kernel function

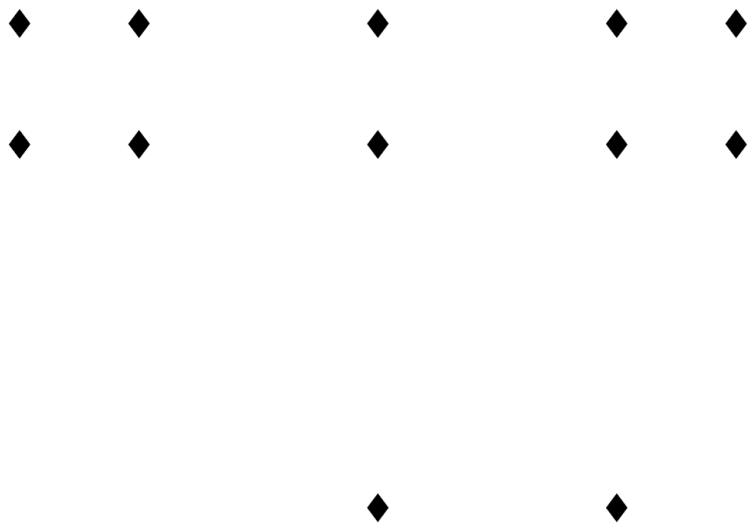
$$K(y) = e^{-distance(\mathbf{x},\mathbf{y})^2/2\sigma^2}$$



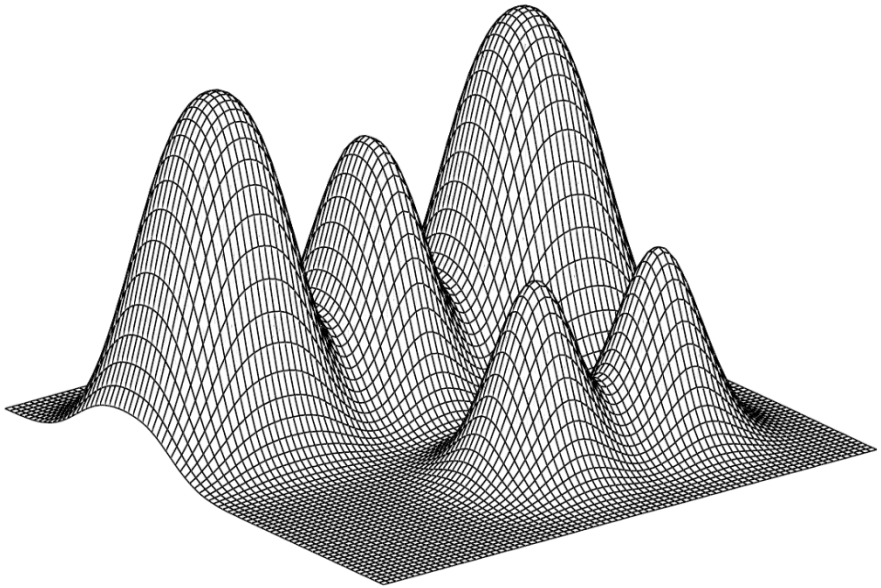
Formula and plot of Gaussian Kernel

  - Overall density is the sum of the contributions of all points

Example of Density from Gaussian Kernel



Set of 12 points.

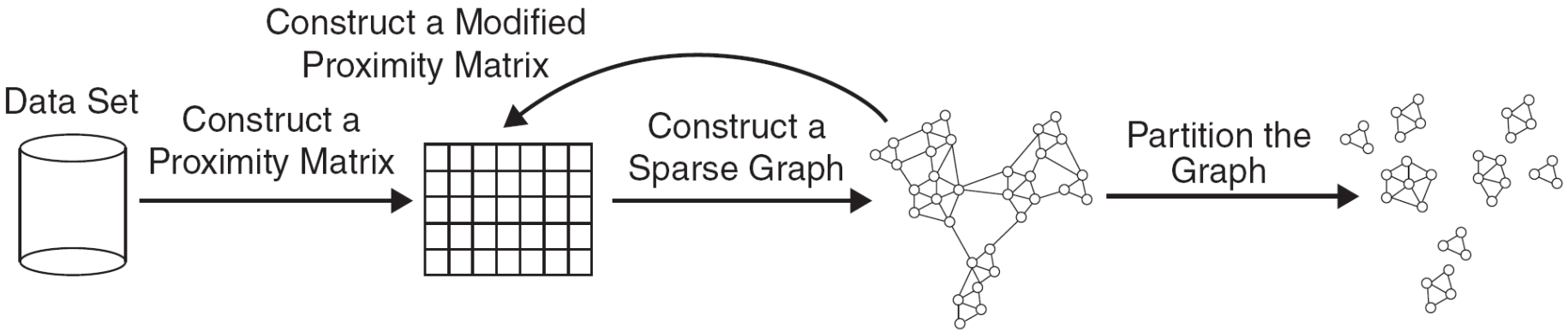


Overall density—surface plot.

Graph-Based Clustering: Chameleon

- Based on several key ideas
  - Sparsification of the proximity graph to keep only the connections of an object with its nearest neighbors. This is useful for handling noise and outliers.
  - Partitioning the data into clusters that are relatively pure subclusters of the “true” clusters
  - Merging based on preserving characteristics of clusters

Sparsification in the Clustering Process



Graph-Based Clustering: Sparsification

- The amount of data that needs to be processed is drastically reduced
  - Sparsification can eliminate more than 99% of the entries in a proximity matrix
  - The amount of time required to cluster the data is drastically reduced
  - The size of the problems that can be handled is increased

Graph-Based Clustering: Sparsification ...

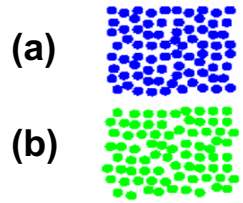
- Clustering may work better
  - Sparsification techniques keep the connections to the most similar (nearest) neighbors of a point while breaking the connections to less similar points.
  - The nearest neighbors of a point tend to belong to the same class as the point itself.
  - This reduces the impact of noise and outliers and sharpens the distinction between clusters.

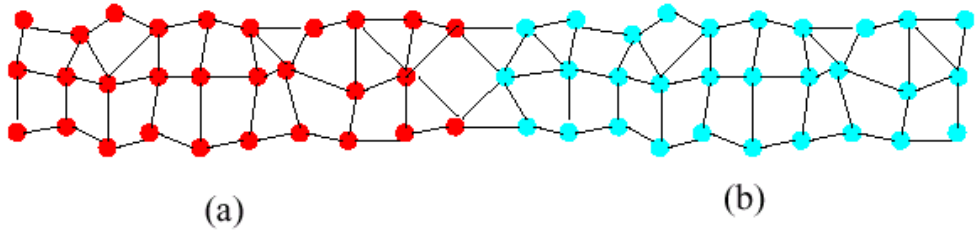
- Sparsification facilitates the use of graph partitioning algorithms (or algorithms based on graph partitioning algorithms)
  - Chameleon and Hypergraph-based Clustering

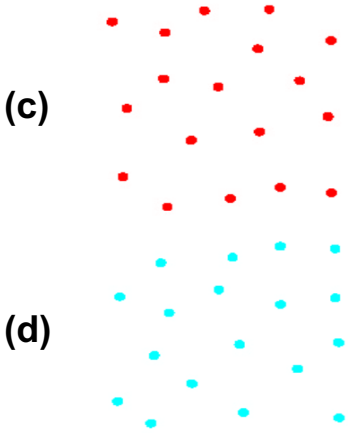
Limitations of Current Merging Schemes

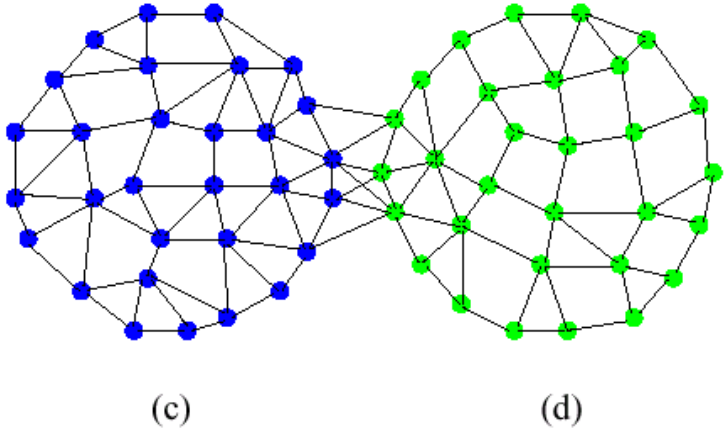
- Existing merging schemes in hierarchical clustering algorithms are static in nature
  - MIN or CURE:
    - Merge two clusters based on their *closeness* (or minimum distance)
  - GROUP-AVERAGE:
    - Merge two clusters based on their average *connectivity*

Limitations of Current Merging Schemes







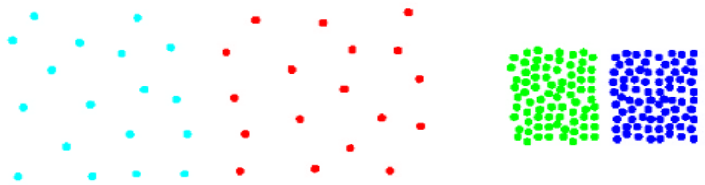


Closeness schemes will merge (a) and (b)

Average connectivity schemes will merge (c) and (d)

Chameleon: Clustering Using Dynamic Modeling

- Hierarchical Clustering with Dynamic Modeling
- Adapt to the characteristics of the data set to find the natural clusters
- Use a dynamic model to measure the similarity between clusters
  - Main properties are the relative closeness and relative inter-connectivity of the cluster
  - Two clusters are combined if the resulting cluster shares certain *properties* with the constituent clusters
  - The merging scheme preserves *self-similarity*



Relative Closeness

- **Relative Closeness (RC)** is the absolute closeness of two clusters normalized by the internal closeness of the clusters. Two clusters are combined only if the points in the resulting cluster are almost as close to each other as in each of the original clusters. Mathematically,

$$RC = \frac{\bar{S}_{EC}(C_i, C_j)}{\frac{m_i}{m_i+m_j}\bar{S}_{EC}(C_i) + \frac{m_j}{m_i+m_j}\bar{S}_{EC}(C_j)}, \tag{9.17}$$

where m_i and m_j are the sizes of clusters C_i and C_j, respectively, $\bar{S}_{EC}(C_i, C_j)$ is the average weight of the edges (of the k-nearest neighbor graph) that connect clusters C_i and C_j; $\bar{S}_{EC}(C_i)$ is the average weight of edges if we bisect cluster C_i; and $\bar{S}_{EC}(C_j)$ is the average weight of edges if we bisect cluster C_j. (EC stands for edge cut.)

Relative Interconnectivity

- **Relative Interconnectivity (RI)** is the absolute interconnectivity of two clusters normalized by the internal connectivity of the clusters. Two clusters are combined if the points in the resulting cluster are almost as strongly connected as points in each of the original clusters. Mathematically,

$$RI = \frac{EC(C_i, C_j)}{\frac{1}{2}(EC(C_i) + EC(C_j))}, \tag{9.18}$$

where $EC(C_i, C_j)$ is the sum of the edges (of the k-nearest neighbor graph) that connect clusters C_i and C_j; $EC(C_i)$ is the minimum sum of the cut edges if we bisect cluster C_i; and $EC(C_j)$ is the minimum sum of the cut edges if we bisect cluster C_j.

Chameleon: Steps

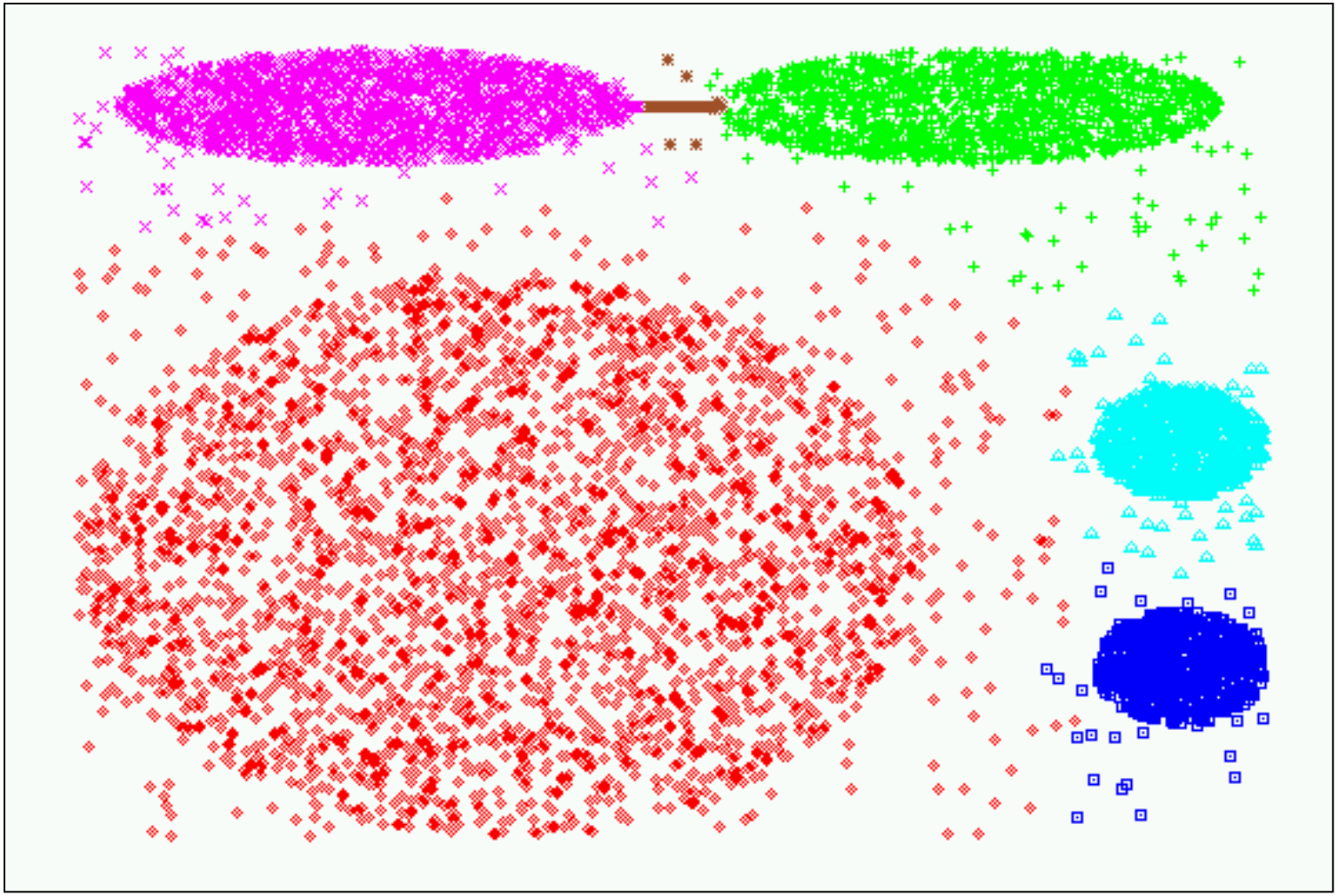
- Preprocessing Step:
  Represent the data by a Graph
  - Given a set of points, construct the k-nearest-neighbor (k-NN) graph to capture the relationship between a point and its k nearest neighbors

- Phase 1: Use a multilevel graph partitioning algorithm on the graph to find a large number of clusters of well-connected vertices
  - Each cluster should contain mostly points from one “true” cluster, i.e., be a sub-cluster of a “real” cluster

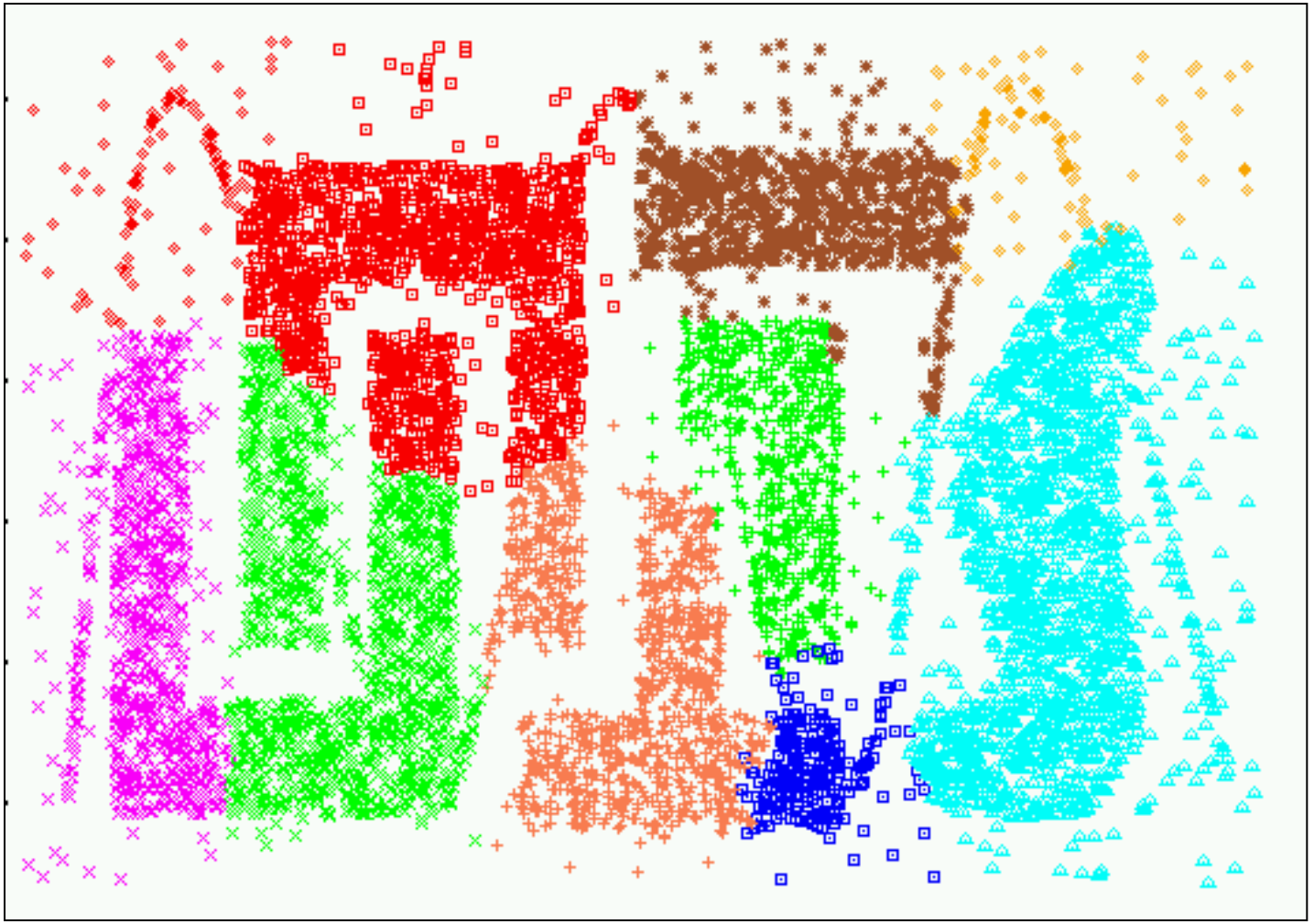
Chameleon: Steps …

- Phase 2: Use Hierarchical Agglomerative Clustering to merge sub-clusters
  - Two clusters are combined if the *resulting cluster shares certain properties with the constituent clusters*
  - Two key properties used to model cluster similarity:
    - Relative Interconnectivity: Absolute interconnectivity of two clusters normalized by the internal connectivity of the clusters
    - Relative Closeness: Absolute closeness of two clusters normalized by the internal closeness of the clusters

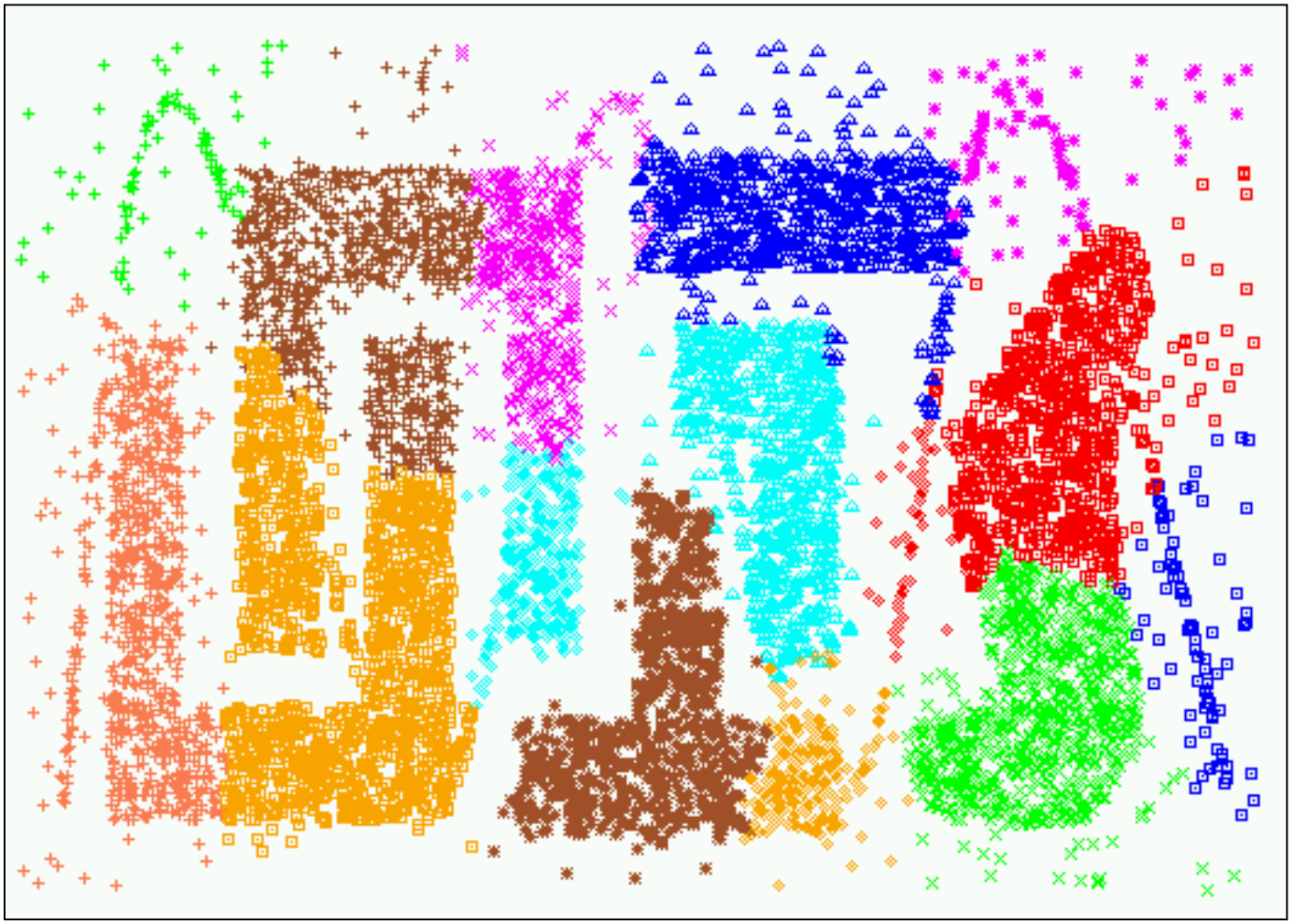
Experimental Results: CHAMELEON



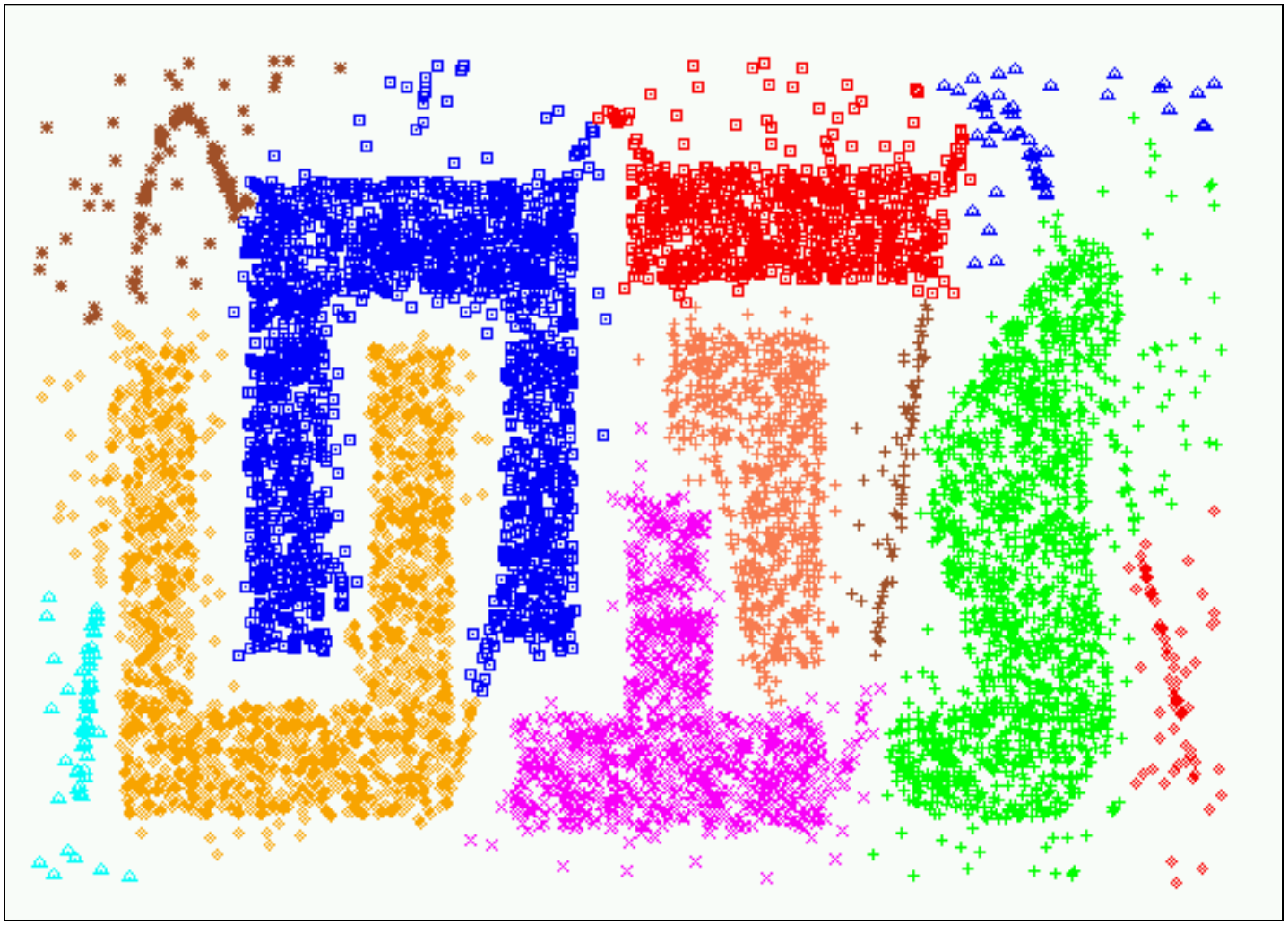
Experimental Results: CURE (*10 clusters*)



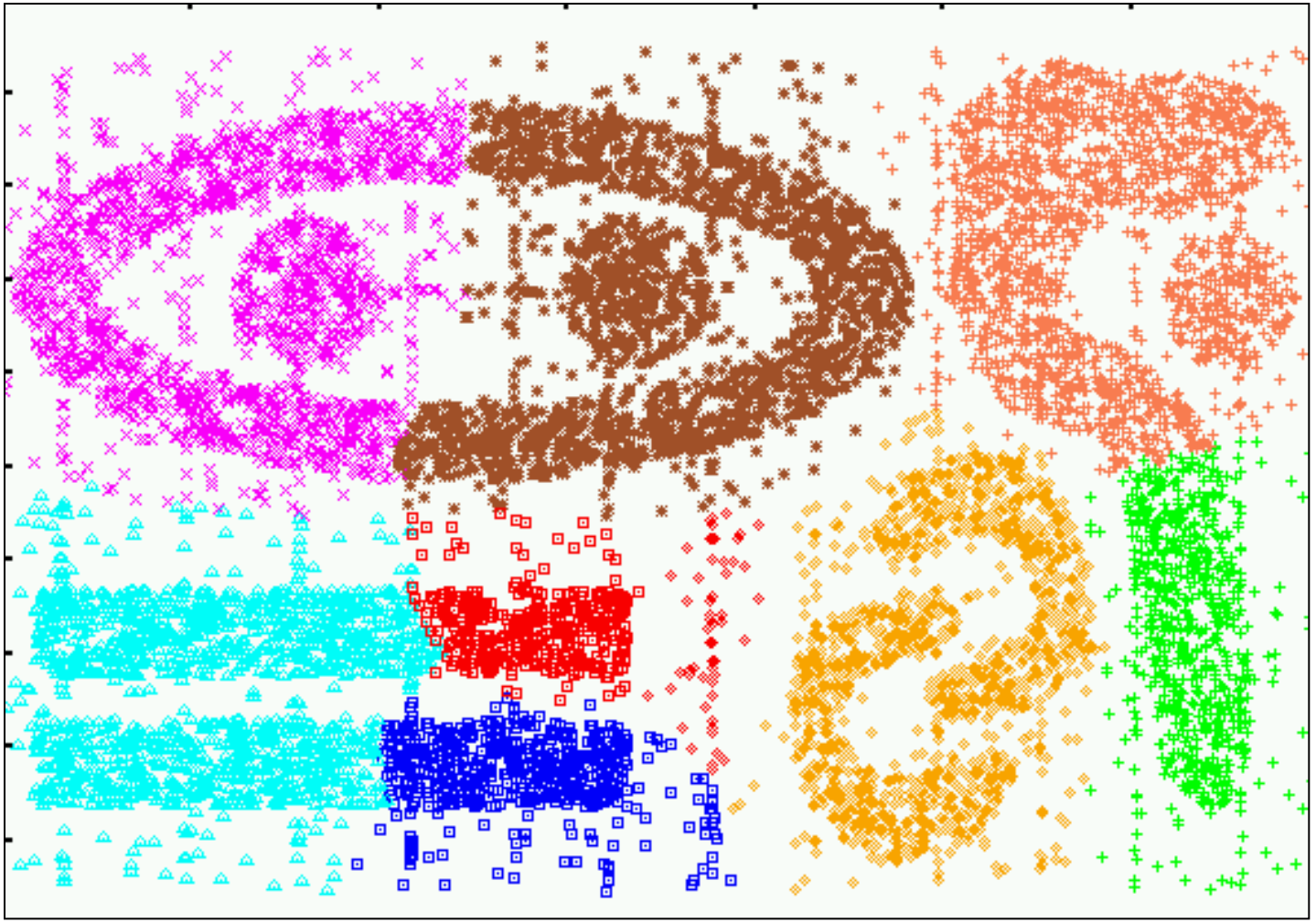
Experimental Results: CURE (*15 clusters*)



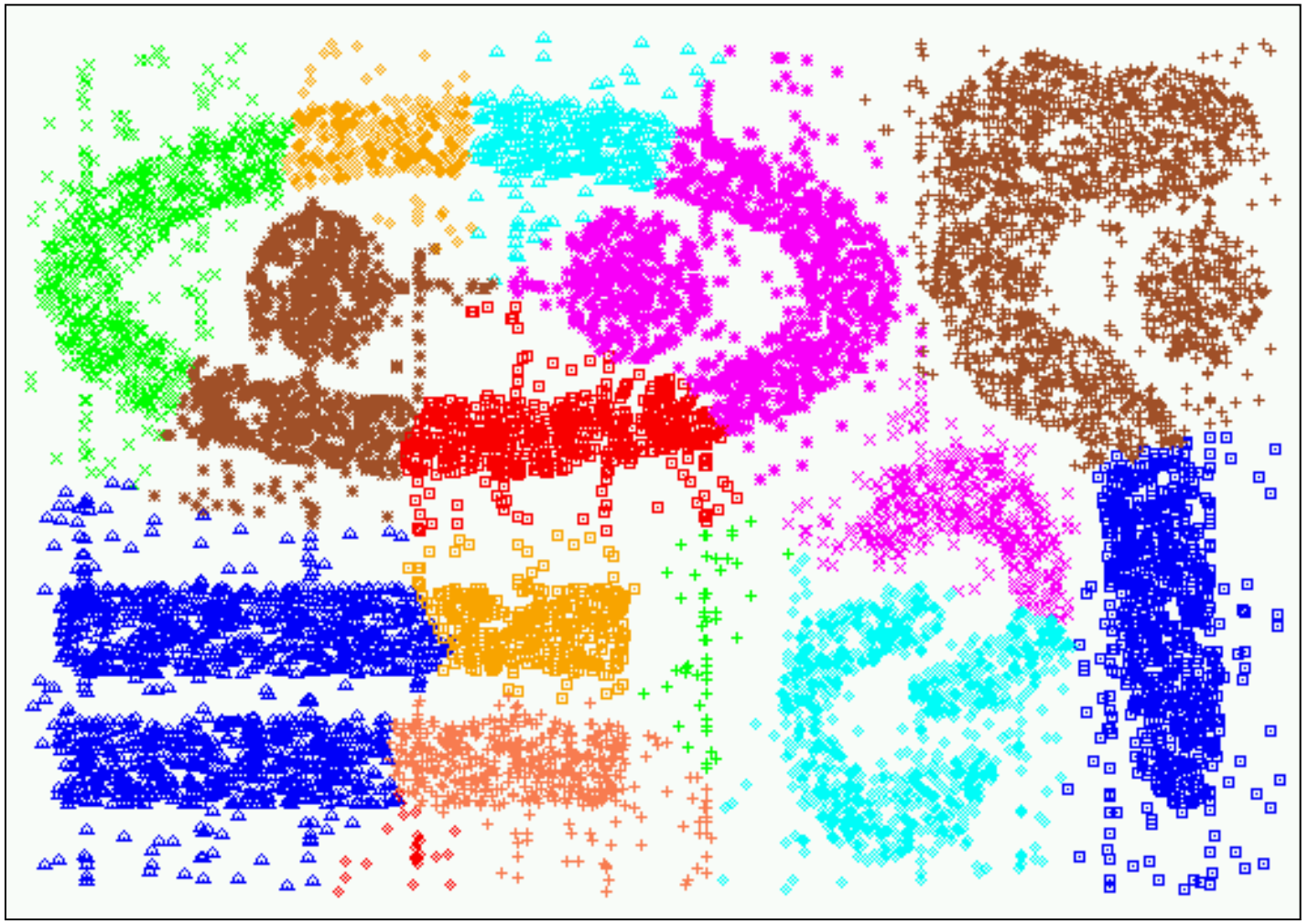
Experimental Results: CHAMELEON



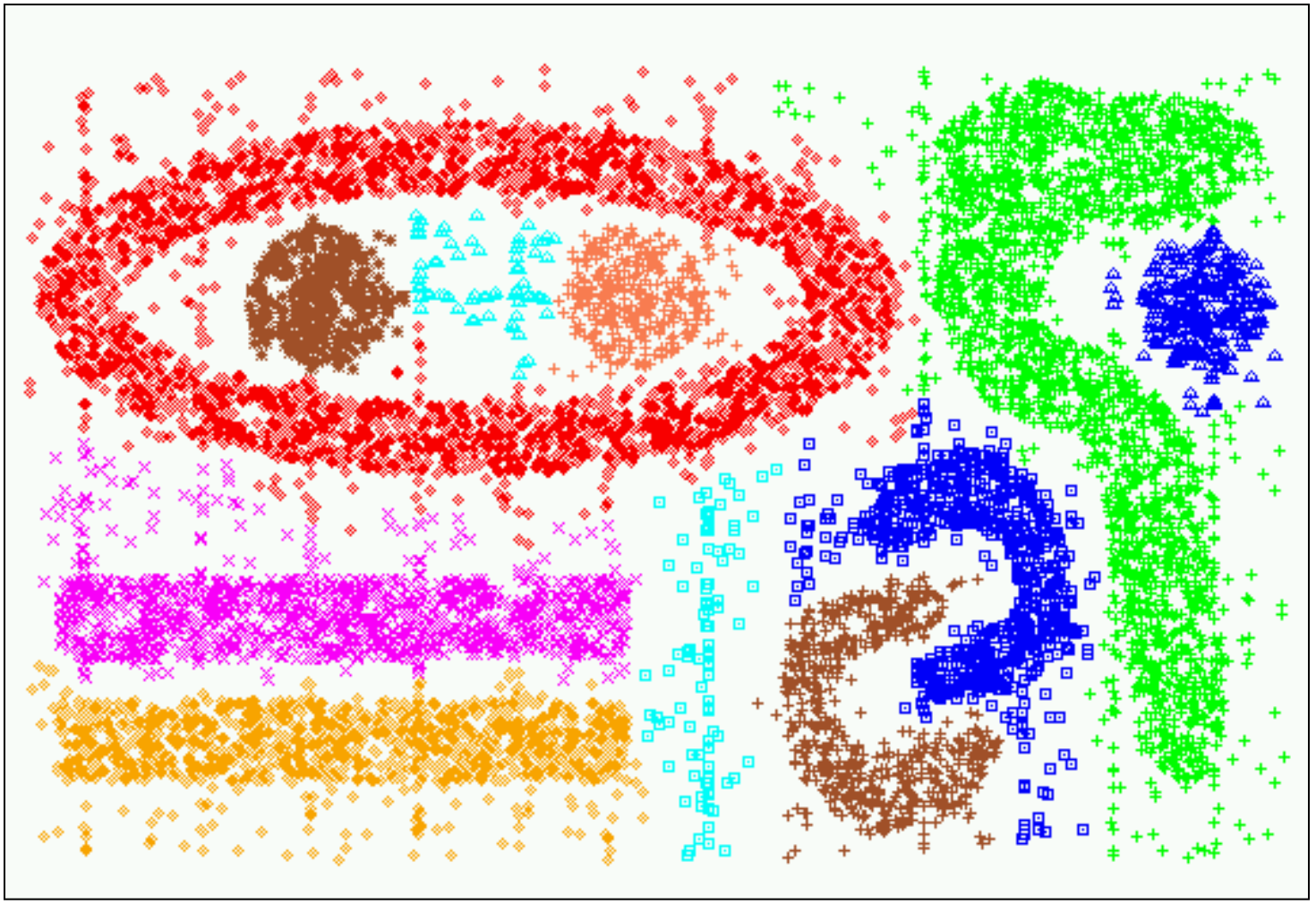
Experimental Results: CURE (*9 clusters*)



Experimental Results: CURE (*15 clusters*)



Experimental Results: CHAMELEON



Comparison of MIN and EM-Clustering

- *We assume EM clustering using the Gaussian (normal) distribution.*
- MIN is hierarchical, EM clustering is partitional.
- Both MIN and EM clustering are complete.
- MIN has a graph-based (contiguity-based) notion of a cluster, while EM clustering has a prototype (or model-based) notion of a cluster.
- MIN will not be able to distinguish poorly separated clusters, but EM can manage this in many situations.
- MIN can find clusters of different shapes and sizes; EM clustering prefers globular clusters and can have trouble with clusters of different sizes.
- Min has trouble with clusters of different densities, while EM can often handle this.
- Neither MIN nor EM clustering finds subspace clusters.

Comparison of MIN and EM-Clustering

- MIN can handle outliers, but noise can join clusters; EM clustering can tolerate noise, but can be strongly affected by outliers.
- EM can only be applied to data for which a centroid is meaningful; MIN only requires a meaningful definition of proximity.
- EM will have trouble as dimensionality increases and the number of its parameters (the number of entries in the covariance matrix) increases as the square of the number of dimensions; MIN can work well with a suitable definition of proximity.
- EM is designed for Euclidean data, although versions of EM clustering have been developed for other types of data. MIN is shielded from the data type by the fact that it uses a similarity matrix.
- MIN makes no distribution assumptions; the version of EM we are considering assumes Gaussian distributions.

Comparison of MIN and EM-Clustering

- EM has an O(n) time complexity; MIN is $O(n^2 \log(n))$.
- Because of random initialization, the clusters found by EM can vary from one run to another; MIN produces the same clusters unless there are ties in the similarity matrix.
- Neither MIN nor EM automatically determine the number of clusters.
- MIN does not have any user-specified parameters; EM has the number of clusters and possibly the weights of the clusters.
- EM clustering can be viewed as an optimization problem; MIN uses a graph model of the data.
- Neither EM or MIN are order dependent.