



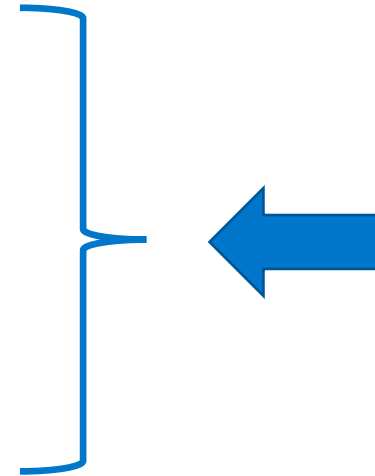
Kourosh Davoudi
kourosh@uoit.ca

Clustering: Advanced Concepts and
Algorithms

CSCI 4150U: Data Mining

Outline

- Soft Clustering
 - Fuzzy c-means
- CURE
- Graph-based Clustering
 - Chameleon
 - Jarvis-Patrick
 - SNN (Density) Clustering
- Characteristics of Clustering Algorithms



Graph-Based Clustering: General Concepts

- Graph-Based clustering uses the proximity graph
 - Start with the **proximity matrix**
 - Consider each **point** as a **node** in a graph
 - Each edge between two nodes has a **weight** which is the **proximity** between the two points
 - Initially the proximity graph is **fully connected**
 - MIN (single-link) and MAX (complete-link) can be viewed in graph terms
- In the simplest case, clusters are connected components in the graph.

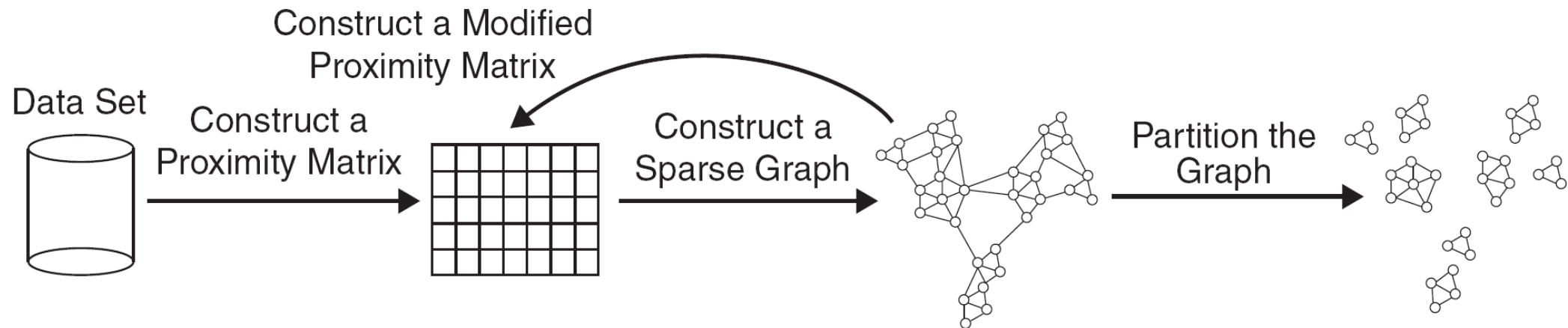
Graph-Based Clustering: Sparsification

- 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

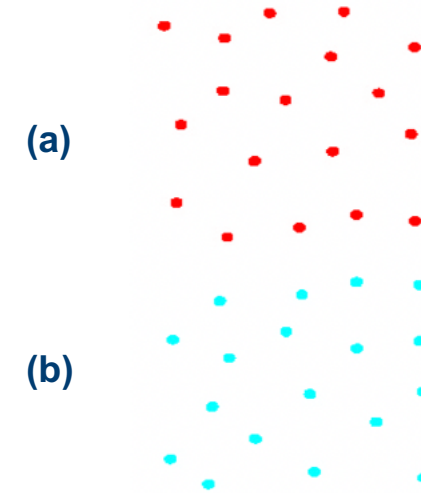
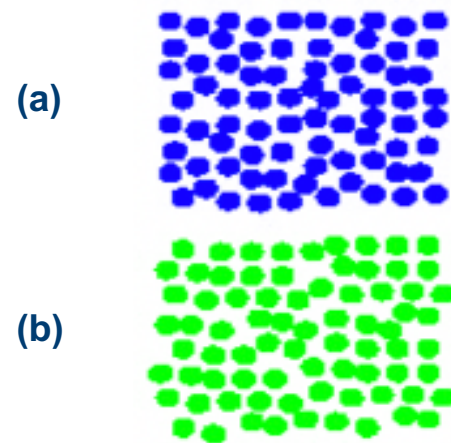
Sparsification in the Clustering Process



Limitations of Current Merging Schemes

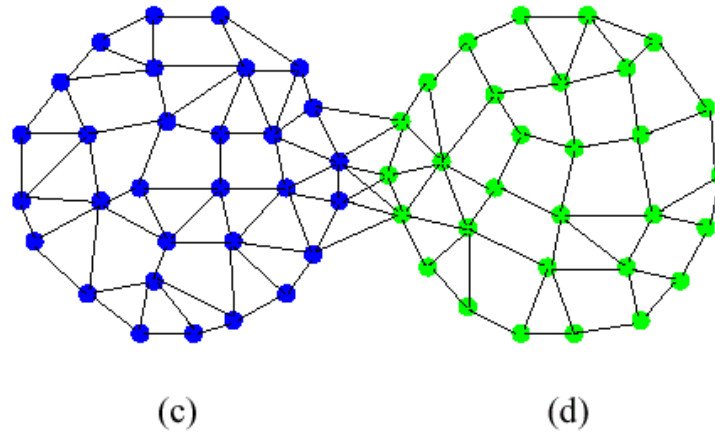
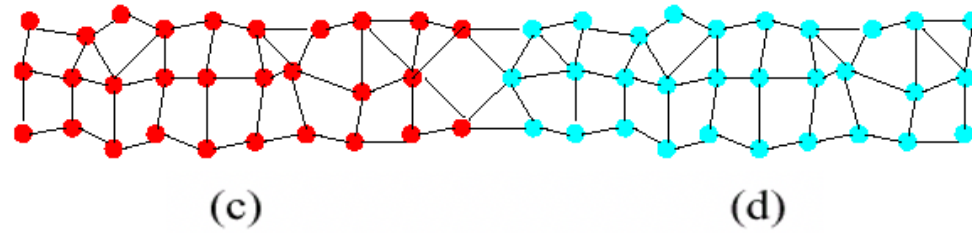
- Existing merging schemes in hierarchical clustering algorithms are static in nature
 - MIN or CURE:
 - Merge two clusters based on their closeness (or minimum distance)
 - GROUP-AVERAGE:
 - Merge two clusters based on their average connectivity

Limitations of Current Merging Schemes



Closeness schemes will merge (a) and (b)

Limitations of Current Merging Schemes



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

Chameleon: Clustering Using 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 property is 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*

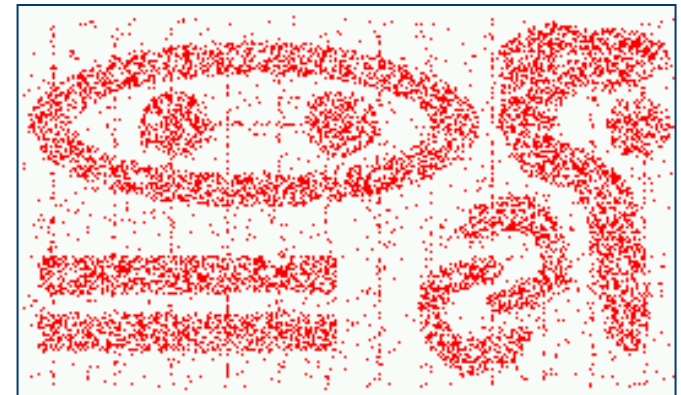


- One of the areas of application is *spatial data*

Characteristics of Spatial Data Sets

- Clusters are defined as densely populated regions of the space
- Clusters have arbitrary shapes, orientation, and non-uniform sizes
- Difference in densities across clusters and variation in density within clusters
- Existence of special artifacts (*streaks*) and noise

The clustering algorithm must address the above characteristics and also require minimal supervision.



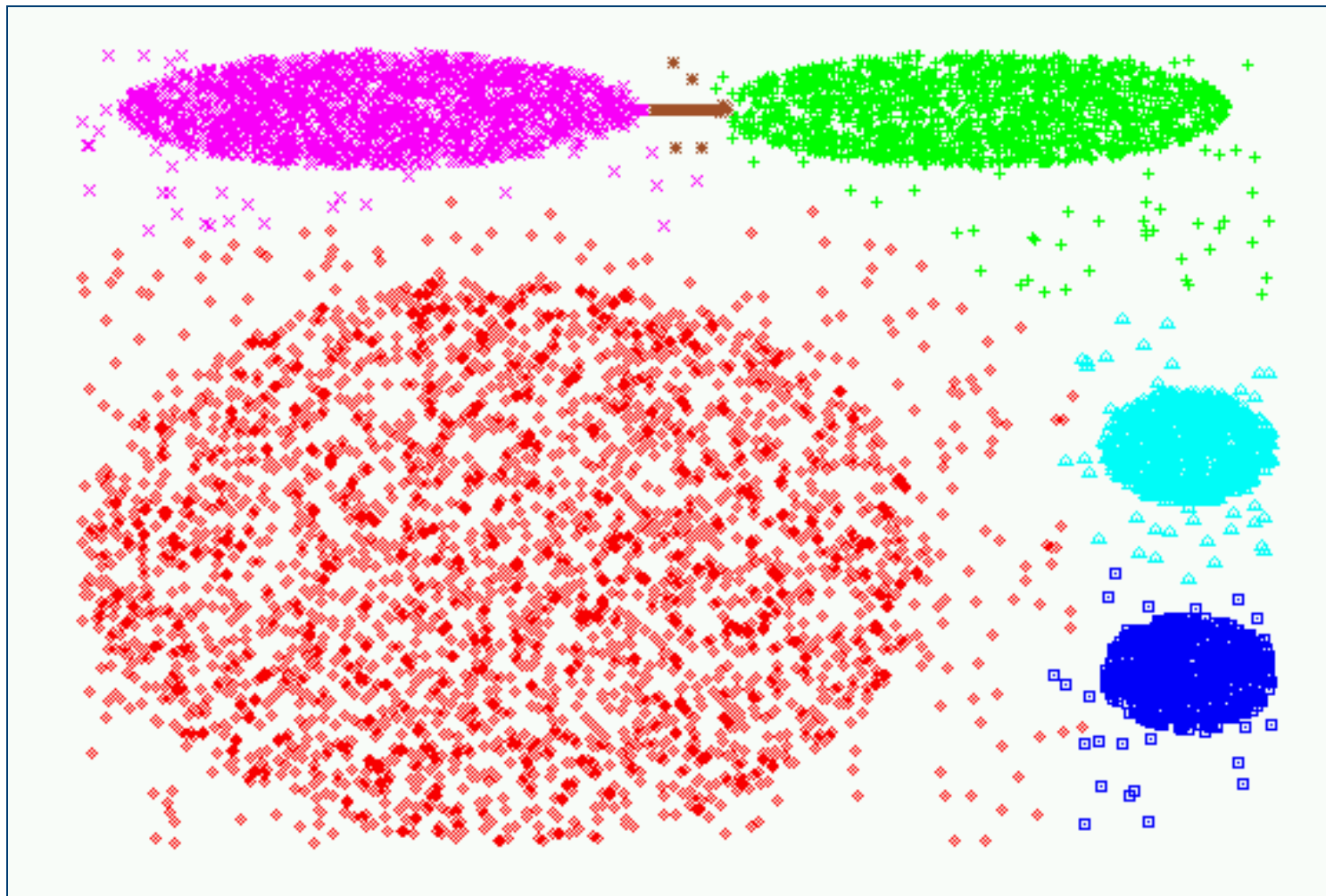
Chameleon: Steps

- **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
 - Concept of neighborhood is captured dynamically (even if region is sparse)
- **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

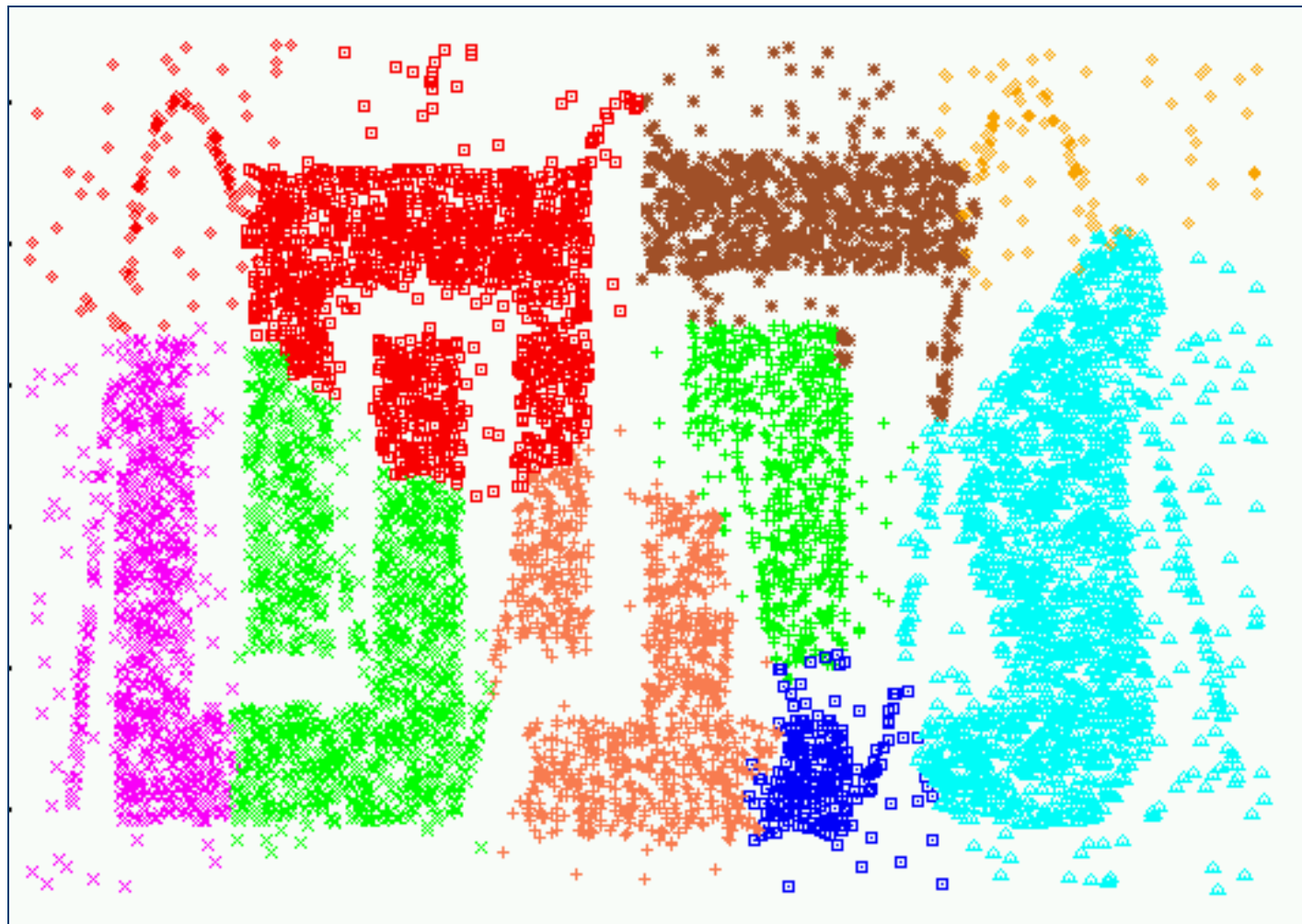
Chameleon: Steps ...

- 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

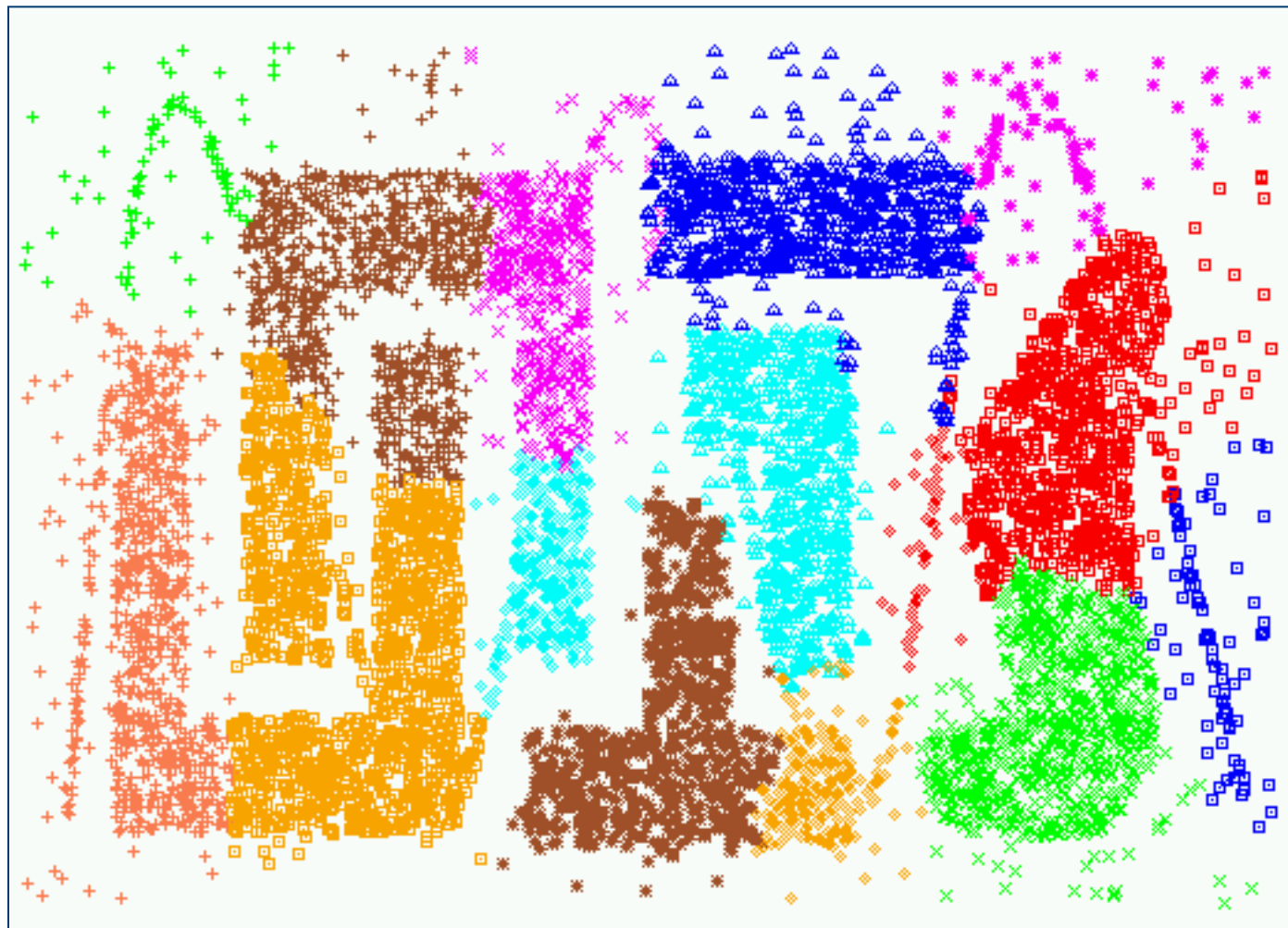
Experimental Results: CHAMELEON



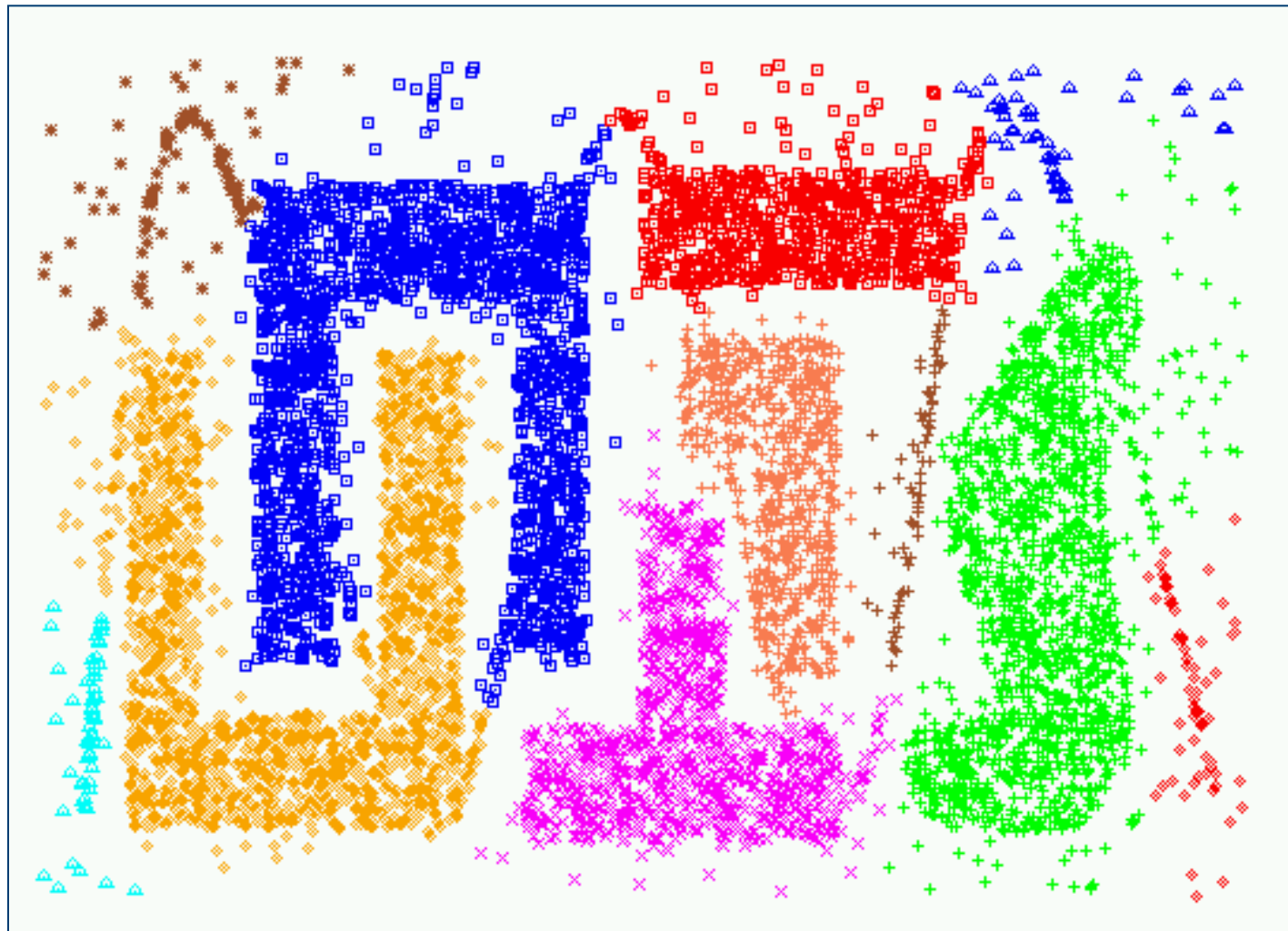
Experimental Results: CURE (10 clusters)



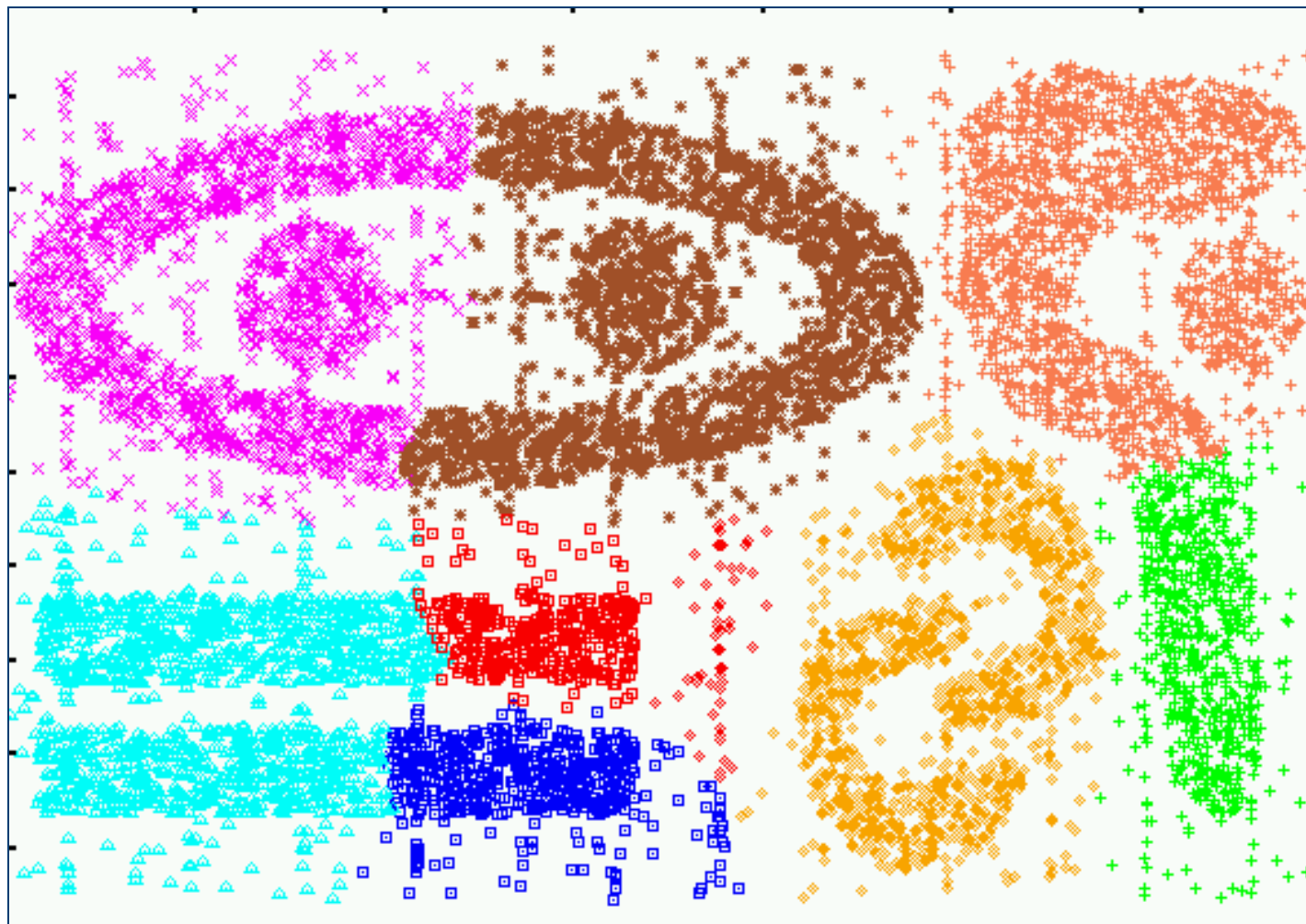
Experimental Results: CURE (15 clusters)



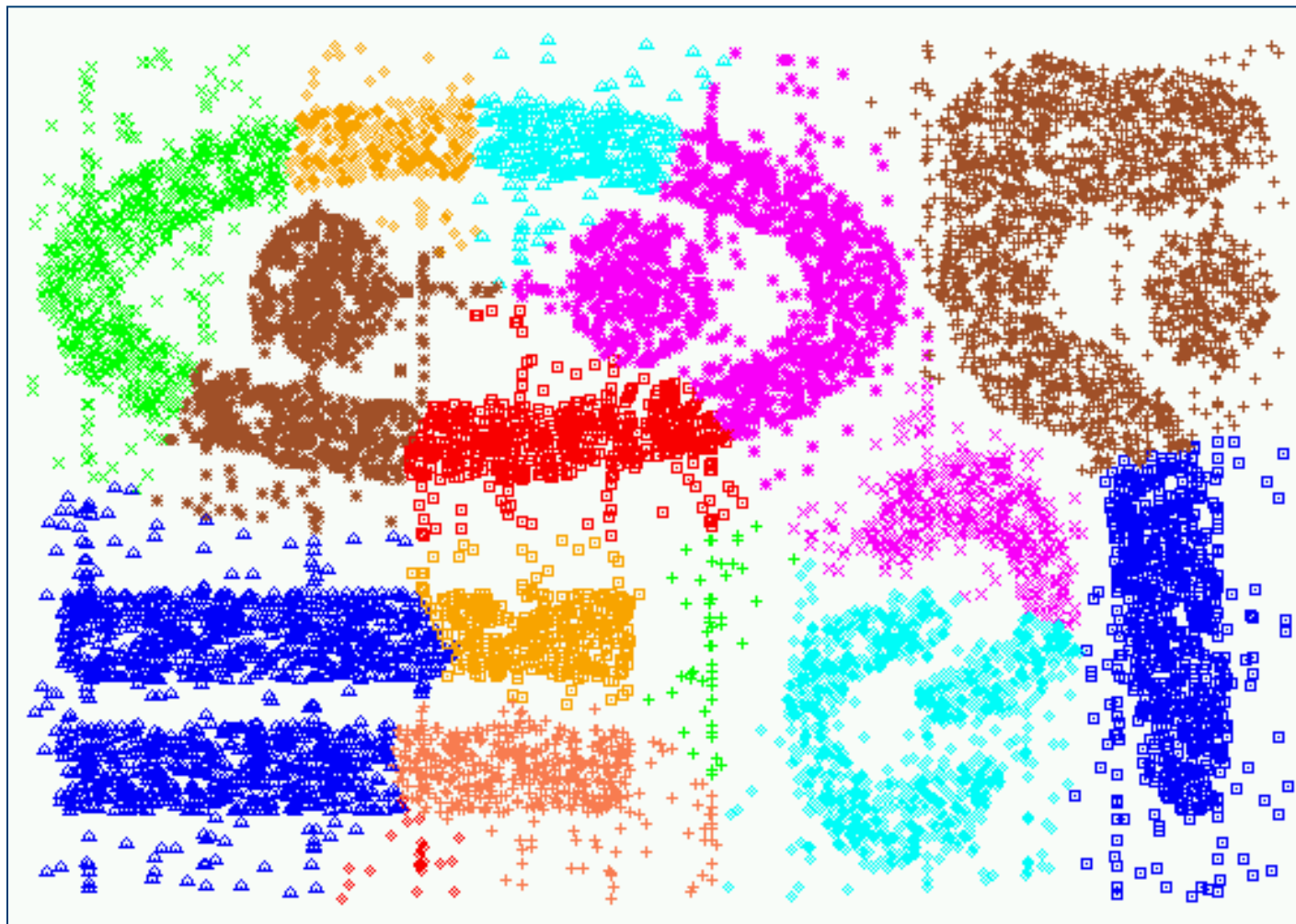
Experimental Results: CHAMELEON



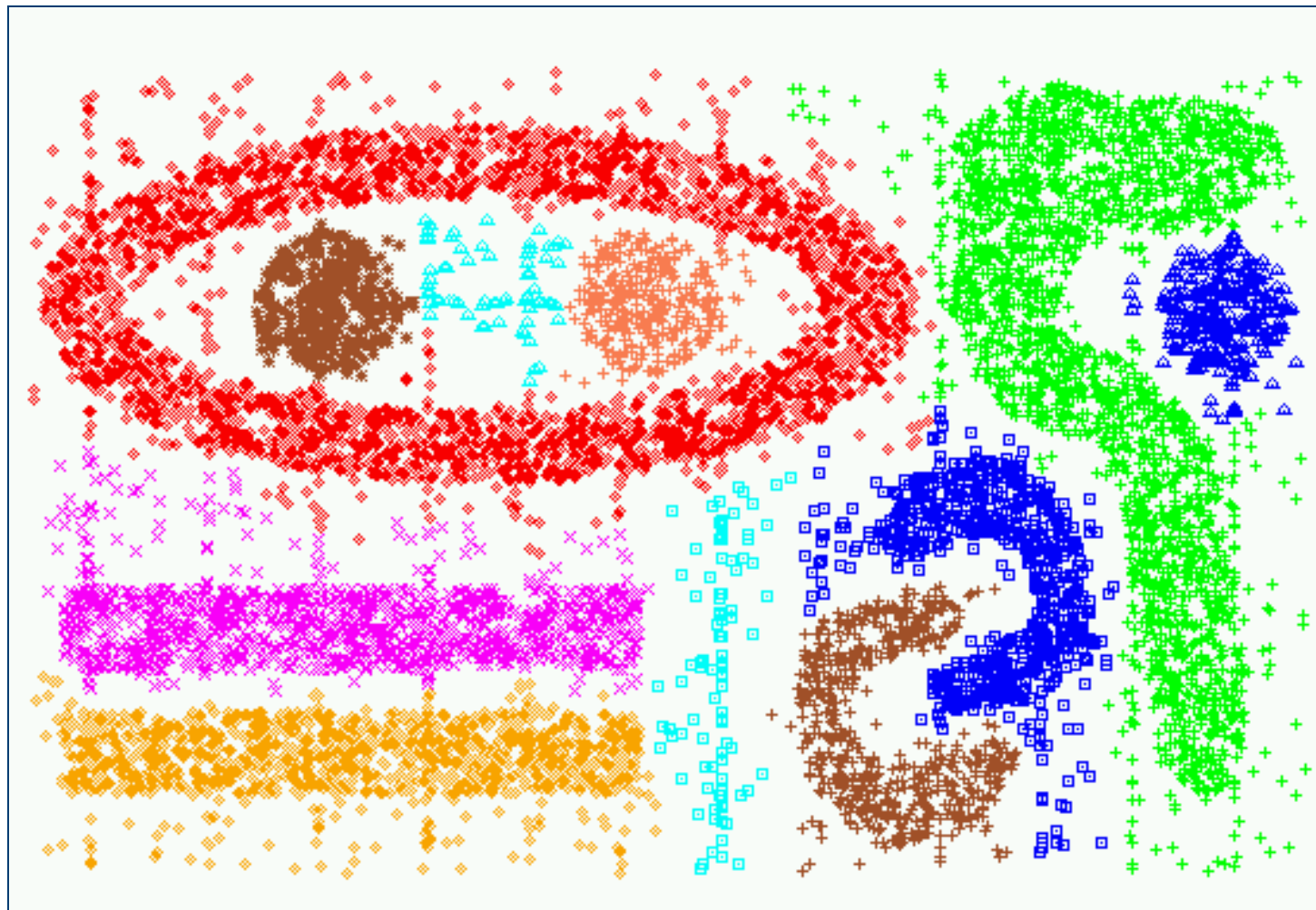
Experimental Results: CURE (9 clusters)



Experimental Results: CURE (15 clusters)

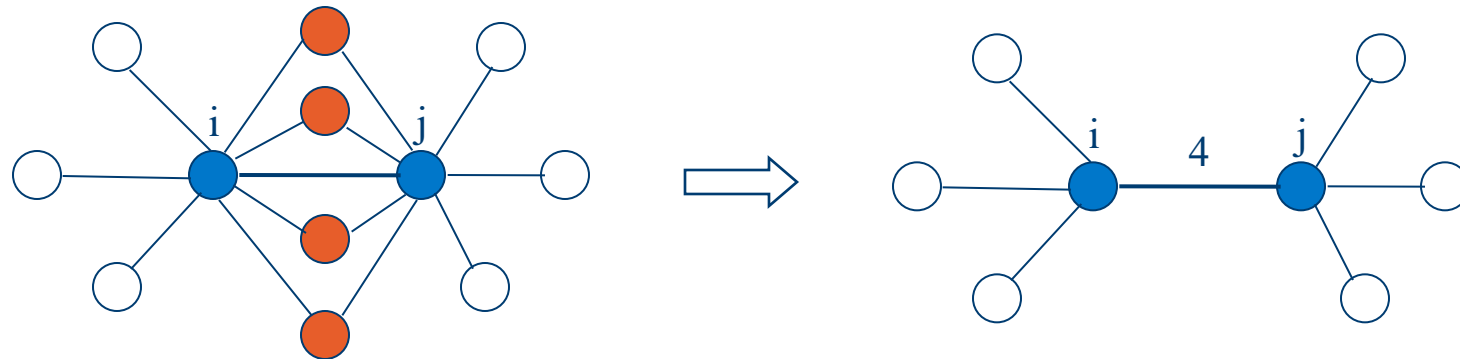


Experimental Results: CHAMELEON

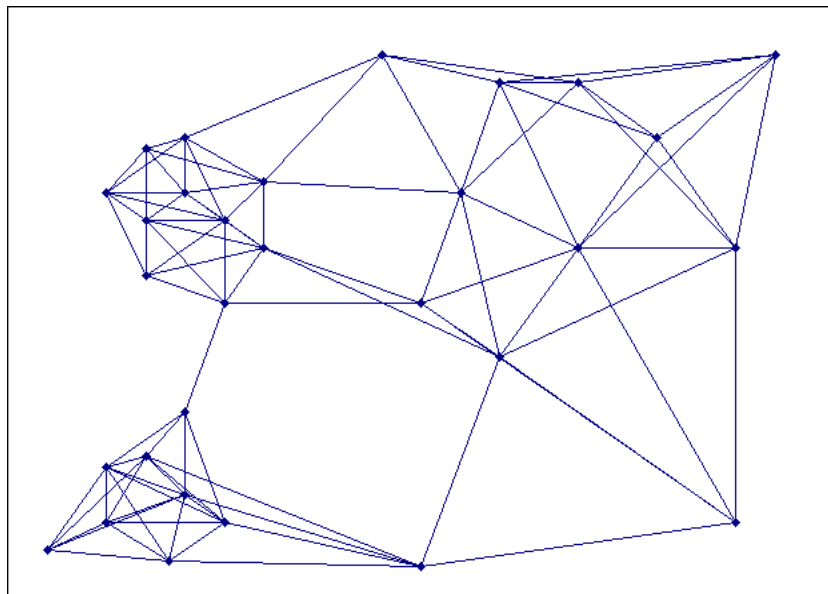


Graph-Based Clustering: SNN Approach

- **Shared Nearest Neighbor (SNN) graph**: the weight of an edge is the number of shared neighbors between vertices given that the vertices are connected

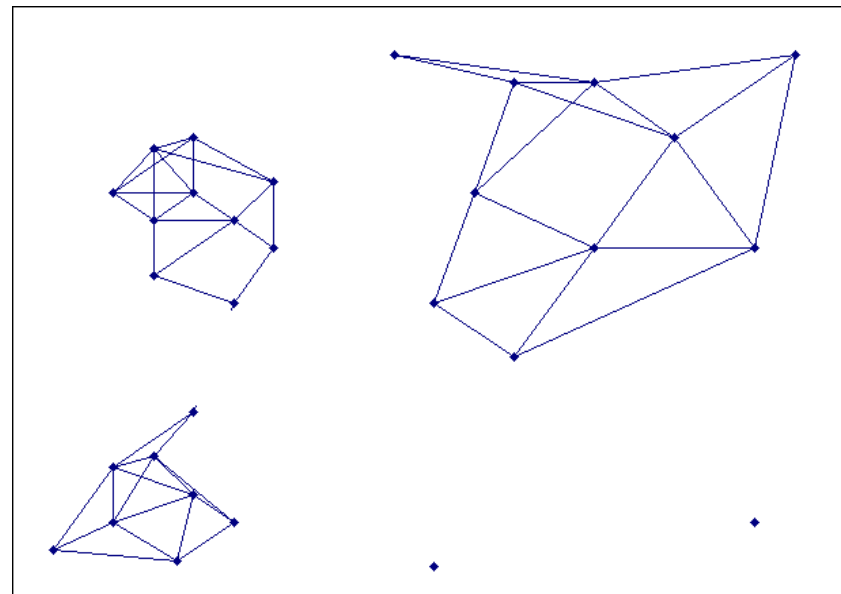


Creating the SNN Graph



Sparse Graph

Link weights are similarities between neighboring points



Shared Near Neighbor Graph

Link weights are **number of Shared Nearest** Neighbors

Jarvis-Patrick Clustering

- First, the **k-nearest neighbors of all points** are found
 - In graph terms this can be regarded as breaking all but the k strongest links from a point to other points in the proximity graph
- A pair of points is put in the same cluster if
 - any two points **share more than T neighbors** and
 - the two points **are in each others k nearest neighbor** list
- For instance, we might choose a nearest neighbor list of size $k=20$ and put points in the same cluster if they share more than $T=10$ near neighbors
- Jarvis-Patrick clustering is too brittle

Example: SNN similarity

- $K=3$

| | A | B | C | D | E | F |
|---|-------|-------|-------|-------|-------|-------|
| A | 1.000 | 0.809 | 0.874 | 0.697 | 0.681 | 0.696 |
| B | 0.809 | 1.000 | 0.746 | 0.673 | 0.590 | 0.594 |
| C | 0.874 | 0.746 | 1.000 | 0.603 | 0.680 | 0.771 |
| D | 0.697 | 0.673 | 0.603 | 1.000 | 0.514 | 0.543 |
| E | 0.681 | 0.590 | 0.680 | 0.514 | 1.000 | 0.622 |
| F | 0.696 | 0.594 | 0.771 | 0.543 | 0.622 | 1.000 |

Similarity Matrix

| | 1st NN | 2nd NN | 3rd NN |
|---|--------|--------|--------|
| A | C | B | D |
| B | A | C | D |
| C | A | F | B |
| D | A | B | C |
| E | A | C | F |
| F | C | A | E |

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | 0 | 2 | 1 | 2 | 0 | 0 |
| B | | 0 | 1 | 2 | 0 | 0 |
| C | | | 0 | 0 | 0 | 1 |
| D | | | | 0 | 0 | 0 |
| E | | | | | 0 | 2 |
| F | | | | | | 0 |

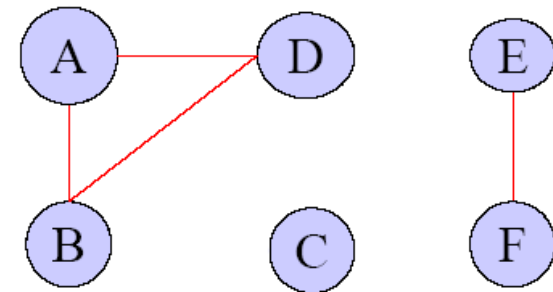
SNN for Jarvis-Patrick Clustering

Example: Jarvis-Patrick Clustering

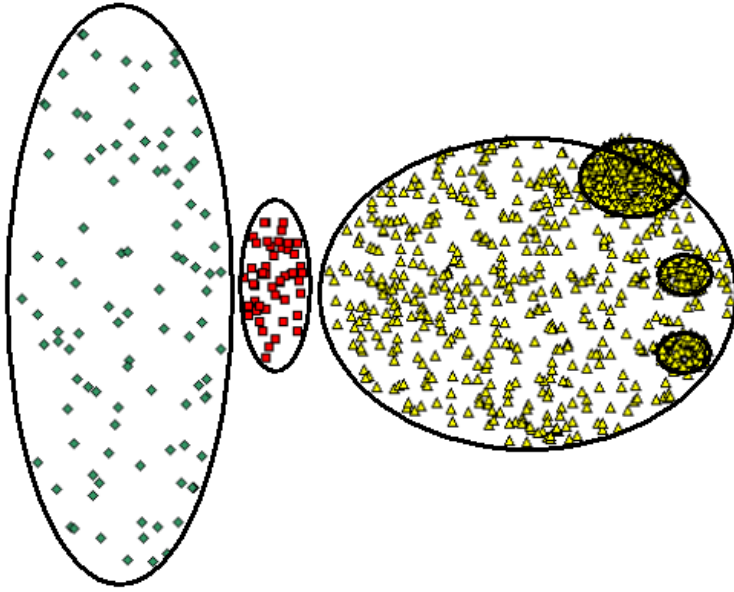
- Jarvis-Patrick Clustering for $K=3$ and $T=2$

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | 0 | 2 | 1 | 2 | 0 | 0 |
| B | | 0 | 1 | 2 | 0 | 0 |
| C | | | 0 | 0 | 0 | 1 |
| D | | | | 0 | 0 | 0 |
| E | | | | | 0 | 2 |
| F | | | | | | 0 |

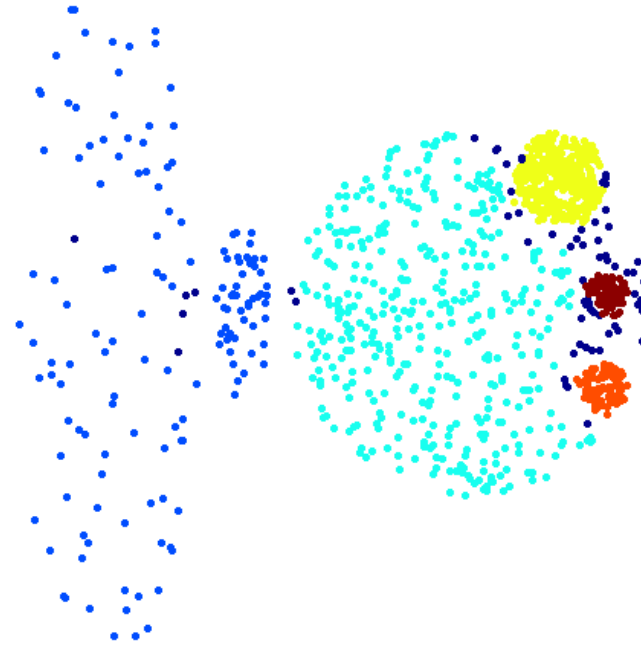
- Thus, three clusters are obtained: $\{A, B, D\}$, $\{C\}$, $\{E, F\}$



When Jarvis-Patrick Works Reasonably Well



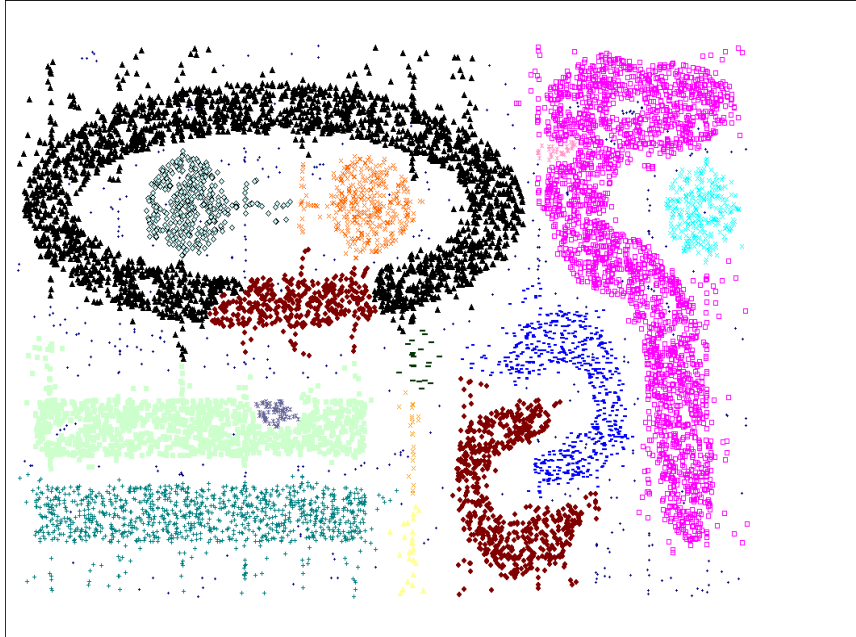
Original Points



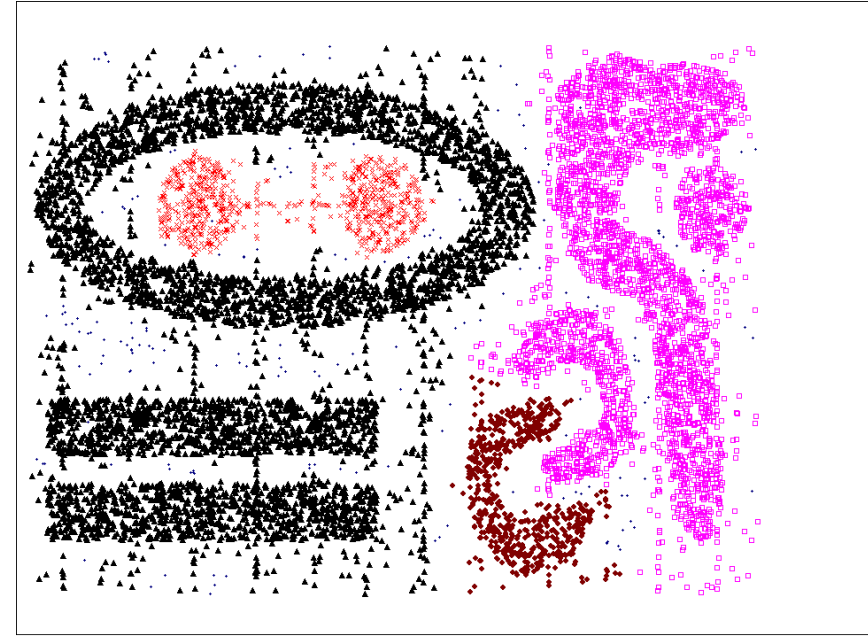
Jarvis Patrick Clustering

6 shared neighbors out of 20

When Jarvis-Patrick Does NOT Work Well



Smallest threshold, T , that does not merge clusters.



Threshold of $T = 1$

SNN Density-Based Clustering

- Combines:
 - Graph based clustering (similarity definition based on number of shared nearest neighbors)
 - Density based clustering (DBSCAN-like approach)
- SNN density measures whether a point is surrounded by similar points (with respect to its nearest neighbors)

SNN Density-Based Clustering

1. Compute the similarity matrix

This corresponds to a similarity graph with data points for nodes and edges whose weights are the similarities between data points

2. Sparsify the similarity matrix by keeping only the k most similar neighbors

This corresponds to only keeping the k strongest links of the similarity graph

3. Construct the shared nearest neighbor graph from the sparsified similarity matrix.

At this point, we could apply a similarity threshold and find the connected components to obtain the clusters (Jarvis-Patrick algorithm)

4. Find the SNN density of each Point.

Using a user specified parameters, *Eps*, find the number points that have an SNN similarity of *Eps* or greater to each point. This is the SNN density of the point



SNN Density-Based Clustering...

5. Find the core points

Using a user specified parameter, *MinPts*, find the core points, i.e., all points that have an SNN density greater than *MinPts*

6. Form clusters from the core points

If two core points are within a “radius”, *Eps*, of each other they are placed in the same cluster

7. Discard all noise points

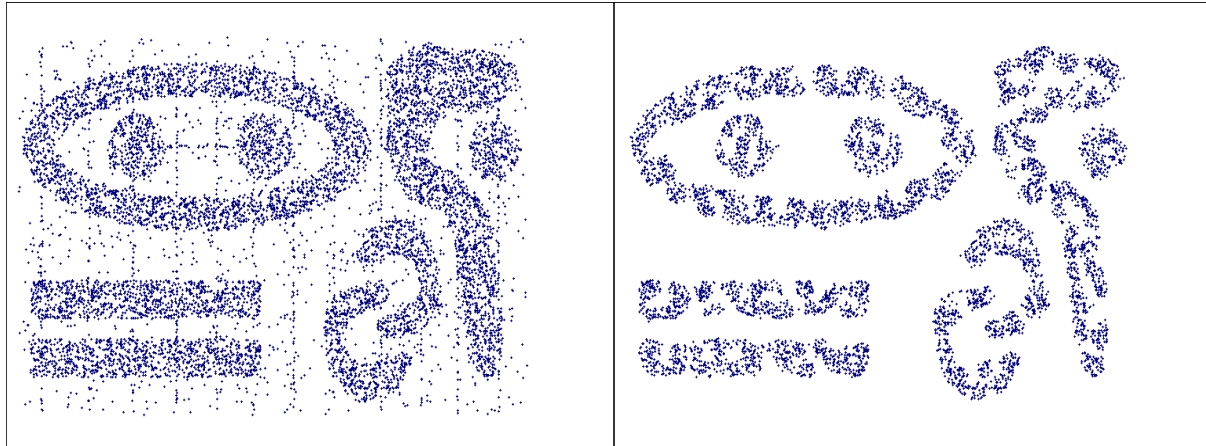
All non-core points that are not within a “radius” of *Eps* of a core point are discarded

8. Assign all non-noise, non-core points to clusters

This can be done by assigning such points to the nearest core point

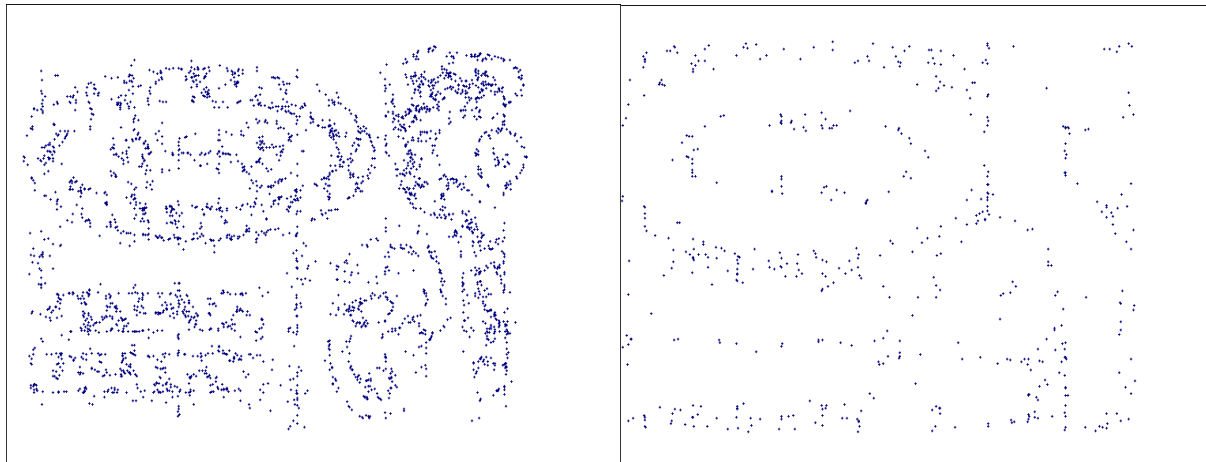
(Note that steps 4-8 are DBSCAN)

SNN Density



a) All Points

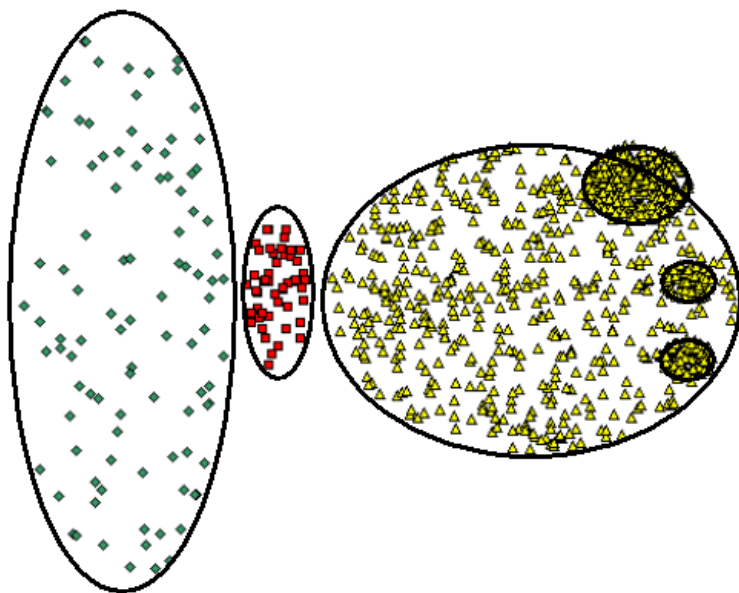
b) High SNN Density



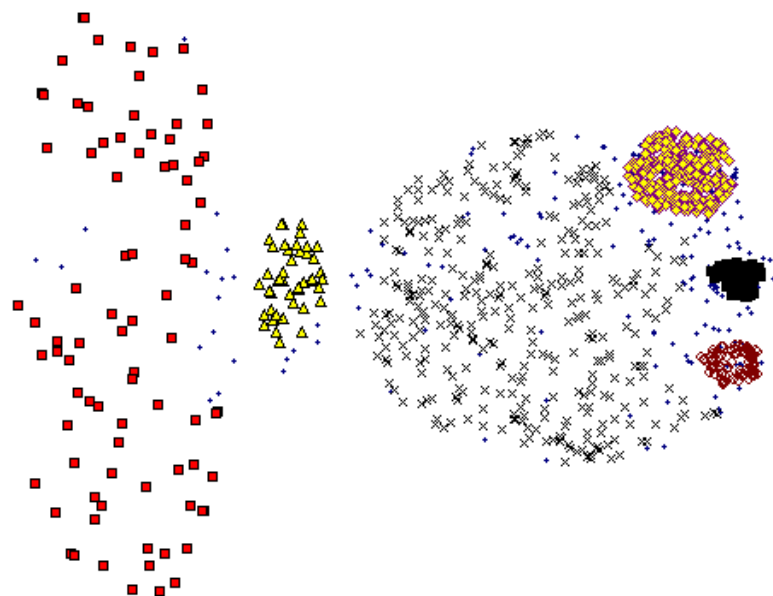
c) Medium SNN Density

d) Low SNN Density

SNN Clustering Can Handle Differing Densities

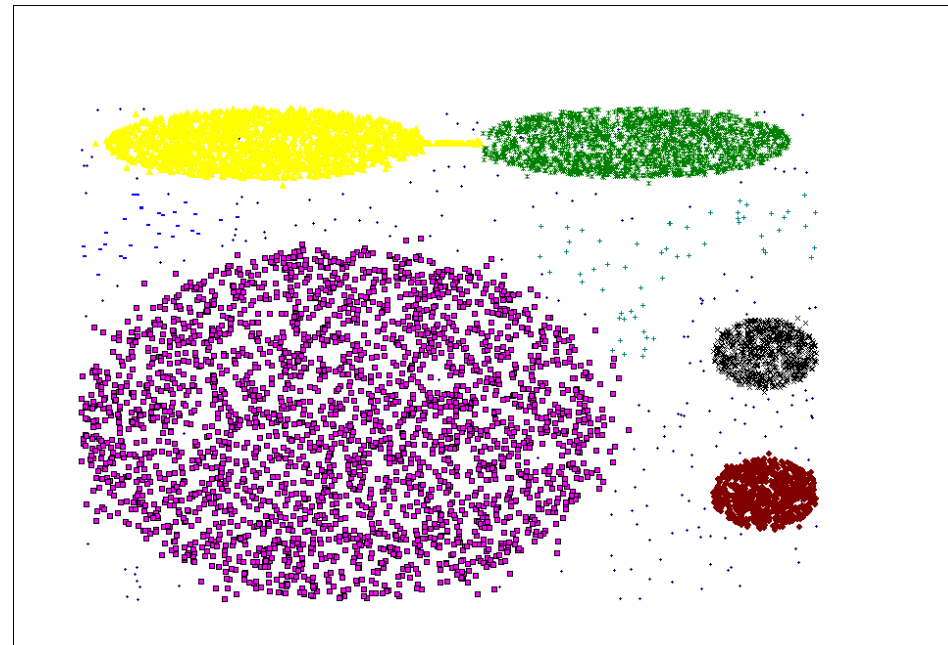
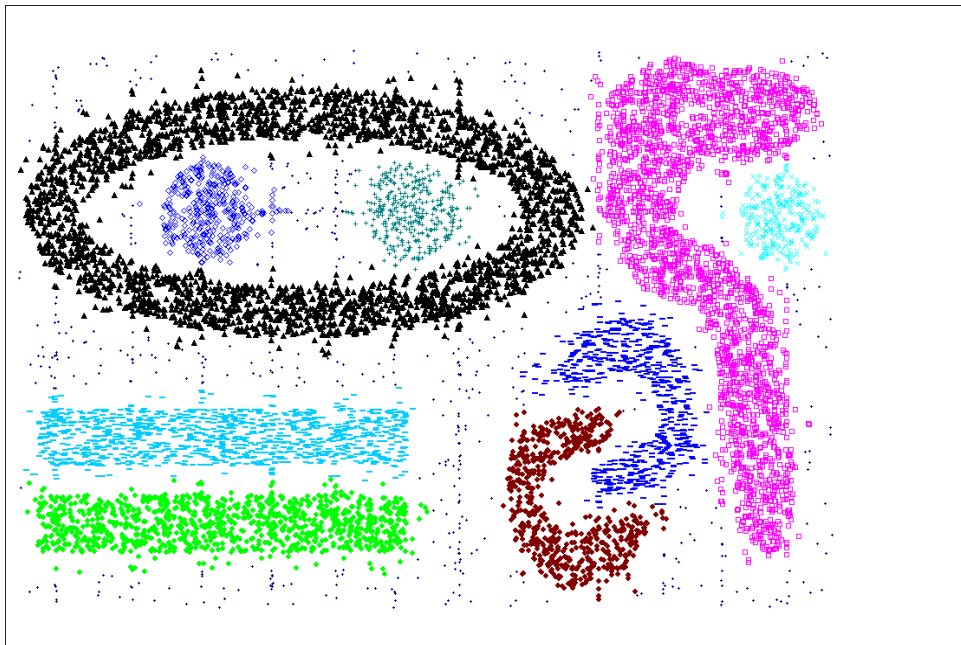


Original Points



SNN Clustering

SNN Clustering Can Handle Other Difficult Situations



Limitations of SNN Clustering

- Does not cluster all the points
- Complexity of SNN Clustering is high
 - $O(n * \text{time to find numbers of neighbor within Eps})$
 - In worst case, this is $O(n^2)$
 - For lower dimensions, there are more efficient ways to find the nearest neighbors
 - R* Tree
 - k-d Trees
- Parameterization is not easy

Characteristics of Data, Clusters, and Clustering Algorithms

- A cluster analysis is affected by characteristics of
 - Data
 - Clusters
 - Clustering algorithms
- Looking at these characteristics gives us a number of dimensions that you can use to describe clustering algorithms and the results that they produce