Clustering
(Part II)

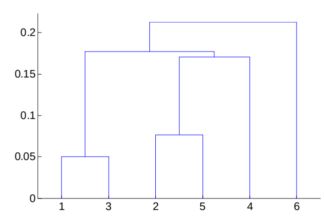
nanopoulos@ismll.de

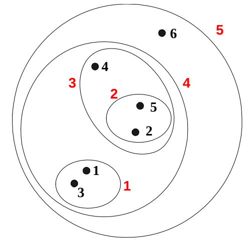
Hierarchical Clustering

Produces a set of nested clusters organized as a hierarchical tree

Can be visualized as a dendrogram

- A tree like diagram that records the sequences of merges or splits





Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by 'cutting' the dendogram at the proper level
- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)

Hierarchical Clustering

- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time

Agglomerative Clustering Algorithm

More popular hierarchical clustering technique

Basic algorithm is straightforward

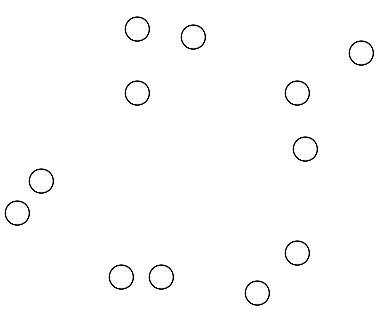
1. Compute the proximity matrix
2. Let each data point be a cluster
3. **Repeat**
4. Merge the two closest clusters
5. Update the proximity matrix
6. **Until** only a single cluster remains

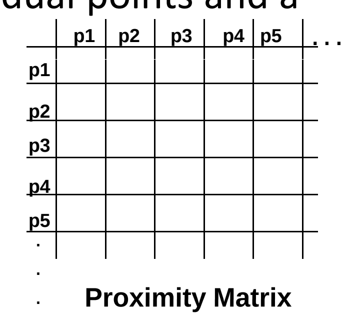
Key operation is the computation of the proximity of two clusters

Different approaches to defining the distance between clusters distinguish the different algorithms

Starting Situation

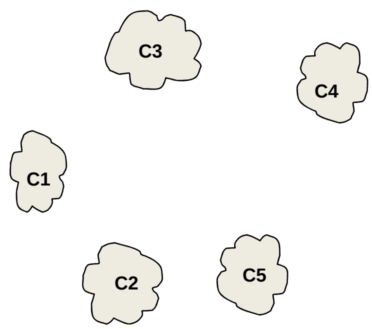
Start with clusters of individual points and a proximity matrix





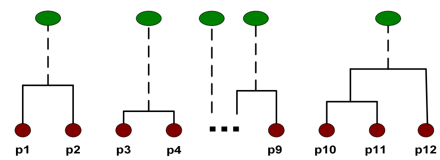
Intermediate Situation

After some merging steps, we have some clusters



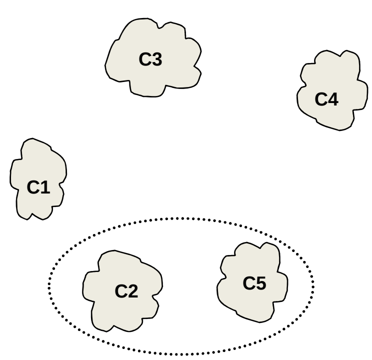
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



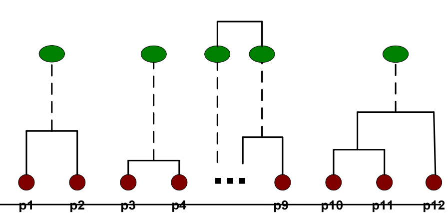
Intermediate Situation

We want to merge the two closest clusters (C2 and C5) and update the proximity matrix.



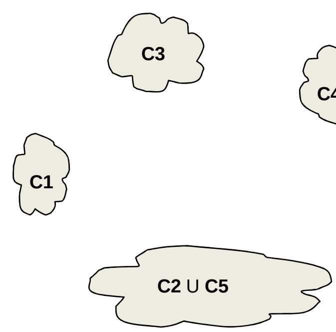
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



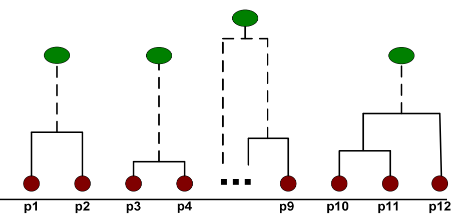
After Merging

The question is "How do we update the proximity matrix?"

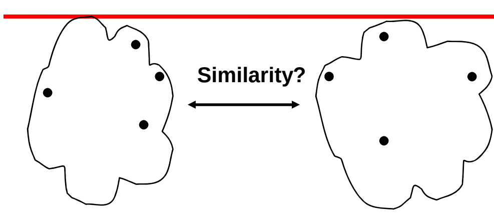


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



How to Define Inter-Cluster Similarity

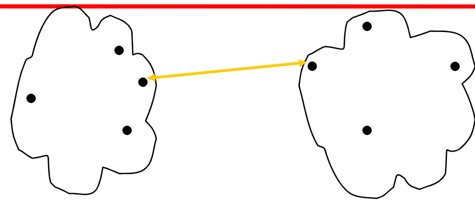


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

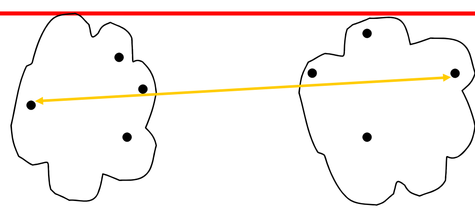


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

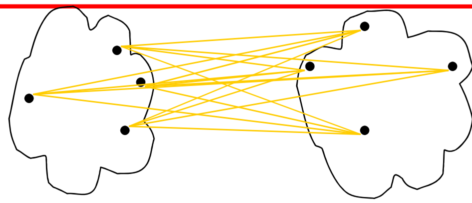


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

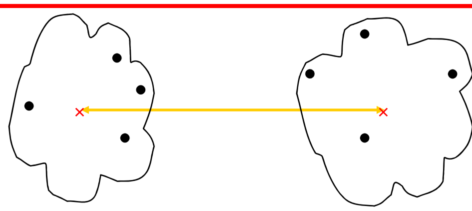


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

How to Define Inter-Cluster Similarity



- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

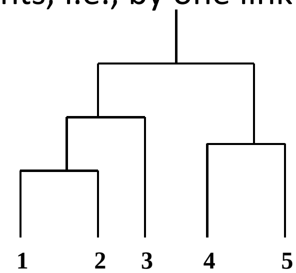
Proximity Matrix

Cluster Similarity: MIN or Single Link

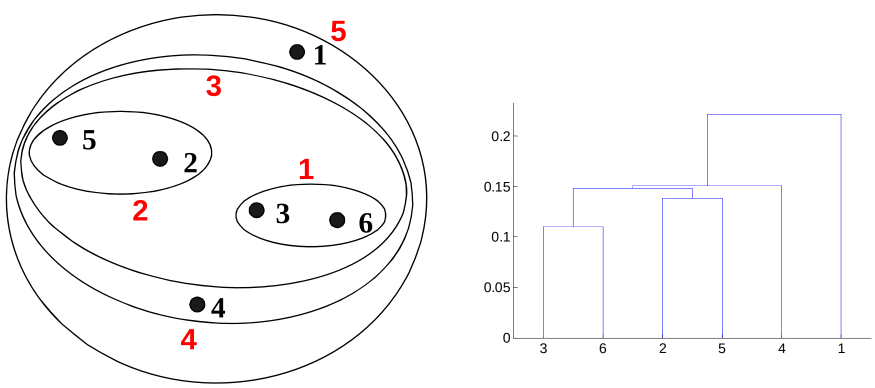
Similarity of two clusters is based on the two most similar (closest) points in the different clusters

- Determined by one pair of points, i.e., by one link in the proximity graph.

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00

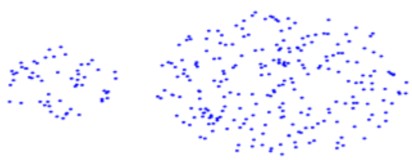


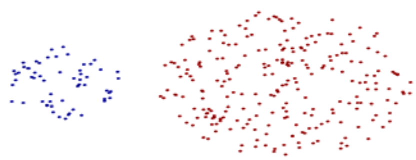
Hierarchical Clustering: MIN



Nested Clusters **Dendrogram**

Strength of MIN

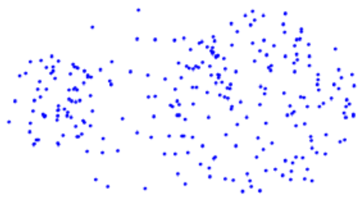


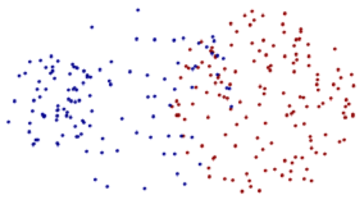


Original Points **Two Clusters**

- **Can handle non-elliptical shapes**

Limitations of MIN





Original Points **Two Clusters**

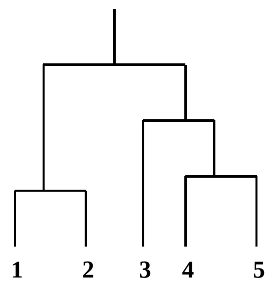
- **Sensitive to noise and outliers**

Cluster Similarity: MAX or Complete Linkage

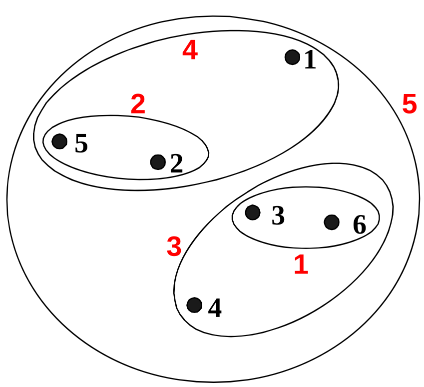
Similarity of two clusters is based on the two least similar (most distant) points in the different clusters

Determined by all pairs of points in the two clusters

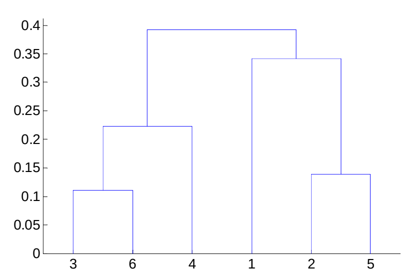
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: MAX



Nested Clusters



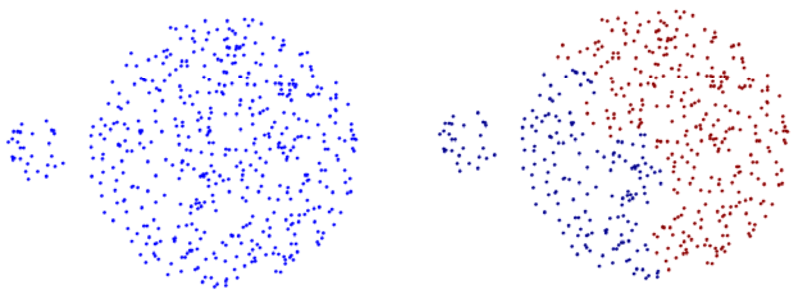
Dendrogram

Strength of MAX

Original Points   Two Clusters

• Less susceptible to noise and outliers

Limitations of MAX



Original Points   Two Clusters

•Tends to break large clusters

•Biased towards globular clusters

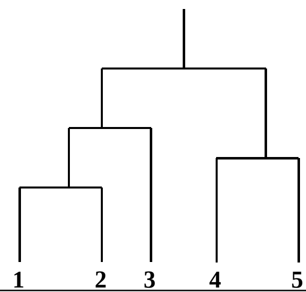
Cluster Similarity: Group Average

Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

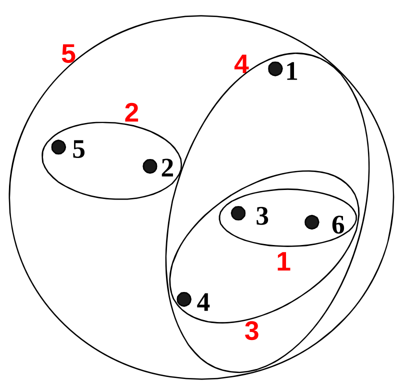
$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum\limits_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| * |\text{Cluster}_j|}$$

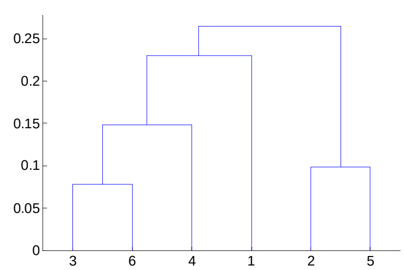
Need to use average connectivity for scalability since total proximity favors large clusters

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: Group Average





Nested Clusters **Dendrogram**

Hierarchical Clustering: Group Average

Compromise between Single and Complete Link

Strengths

- Less susceptible to noise and outliers

Limitations

- Biased towards globular clusters

Cluster Similarity: Ward's Method

Similarity of two clusters is based on the increase in squared error when two clusters are merged

- Similar to group average if distance between points is distance squared

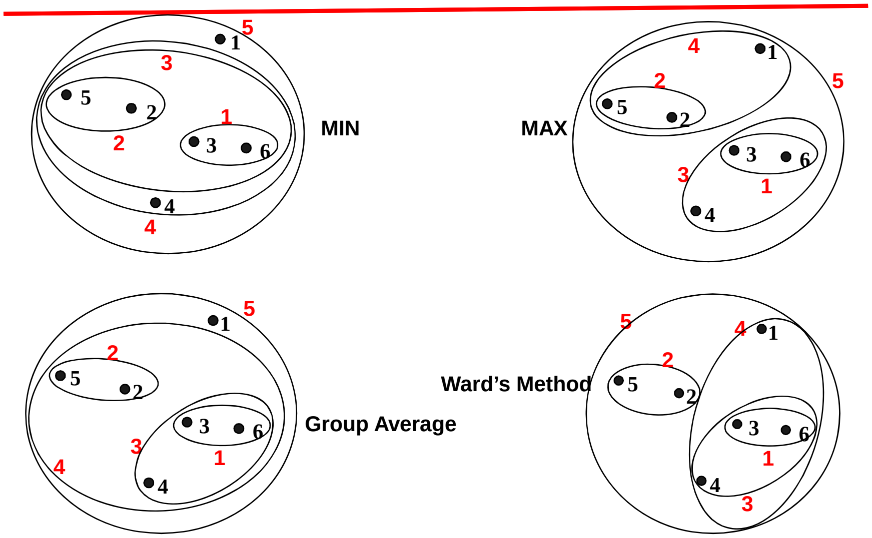
Less susceptible to noise and outliers

Biased towards globular clusters

Hierarchical analogue of K-means

- Can be used to initialize K-means

Hierarchical Clustering: Comparison



Hierarchical Clustering: Time and Space requirements

$O(N^2)$ space since it uses the proximity matrix.

- N is the number of points.

$O(N^3)$ time in many cases

- There are N steps and at each step the size, N^2, proximity matrix must be updated and searched
- Complexity can be reduced to $O(N^2 \log(N))$ time for some approaches

Hierarchical Clustering: Problems and Limitations

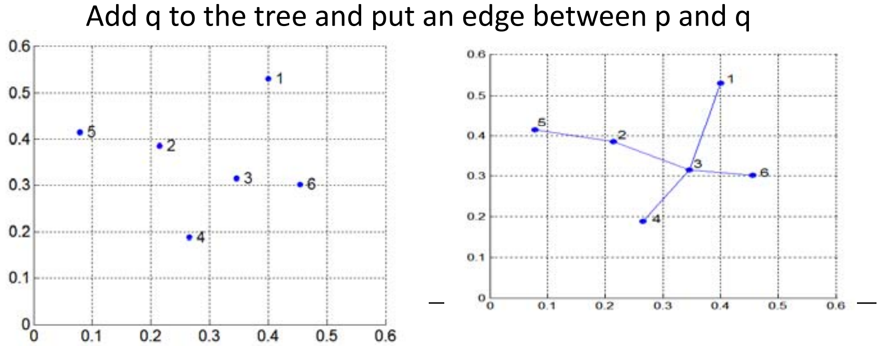
Once a decision is made to combine two clusters, it cannot be undone

No objective function is directly minimized

Different schemes have problems with one or more of the following:
- Sensitivity to noise and outliers
- Difficulty handling different sized clusters and convex shapes
- Breaking large clusters

MST: Divisive Hierarchical Clustering

Build MST (Minimum Spanning Tree)
- Start with a tree that consists of any point
- In successive steps, look for the closest pair of points (p, q) such that one point (p) is in the current tree but the other (q) is not
- Add q to the tree and put an edge between p and q



MST: Divisive Hierarchical Clustering

Use MST for constructing hierarchy of clusters

Algorithm 7.5 MST Divisive Hierarchical Clustering Algorithm

1: Compute a minimum spanning tree for the proximity graph.
2: **repeat**
3: Create a new cluster by breaking the link corresponding to the largest distance (smallest similarity).
4: **until** Only singleton clusters remain

Hierarchical Clustering: Revisited

Creates nested clusters

Agglomerative clustering algorithms vary in terms of how the proximity of two clusters are computed

- MIN (single link): susceptible to noise/outliers
- MAX/GROUP AVERAGE:
  may not work well with non-globular clusters

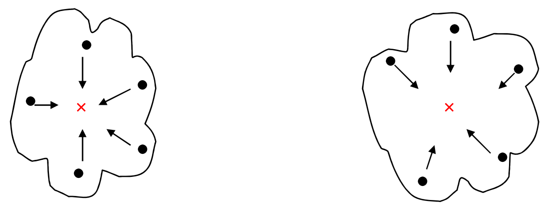
CURE algorithm tries to handle both problems

Often starts with a proximity matrix

A type of graph-based algorithm

CURE: Another Hierarchical Approach

Uses a number of points to represent a cluster



Representative points are found by selecting a constant number of points from a cluster and then "shrinking" them toward the center of the cluster

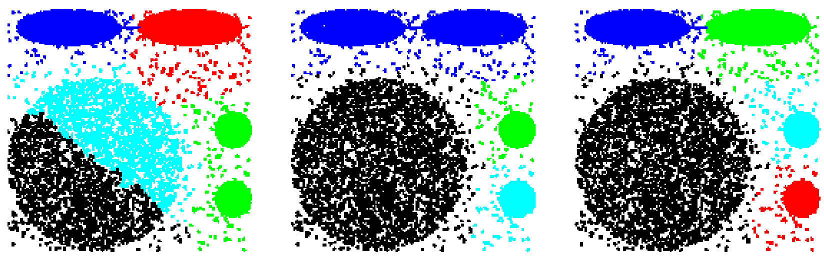
Cluster similarity is the similarity of the closest pair of representative points from different clusters

CURE

Shrinking representative points toward the center helps avoid problems with noise and outliers

CURE is better able to handle clusters of arbitrary shapes and sizes

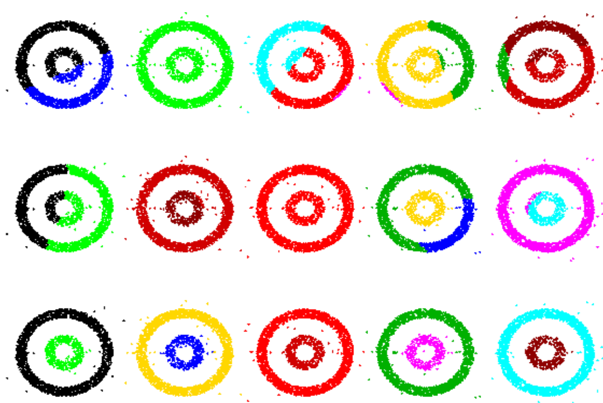
Experimental Results: CURE



a) BIRCH b) MST METHOD c) CURE

Picture from *CURE*, Guha, Rastogi, Shim.

Experimental Results: CURE



a) BIRCH

(centroid)

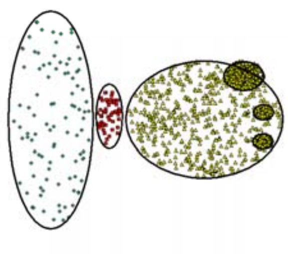
b) MST METHOD

(single link)

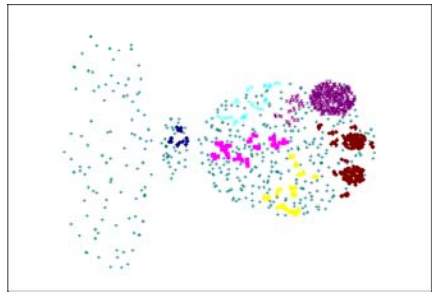
c) CURE

Picture from *CURE*, Guha, Rastogi, Shim.

CURE Cannot Handle Differing Densities



Original Points



CURE

Graph-Based Clustering

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 as starting with this graph

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 …

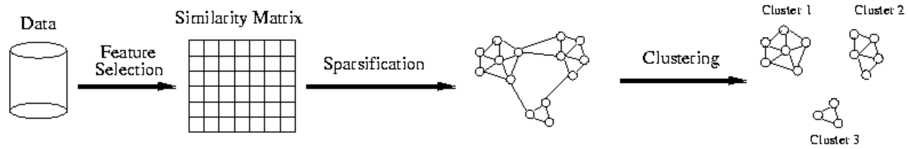
Clustering may work better

- 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

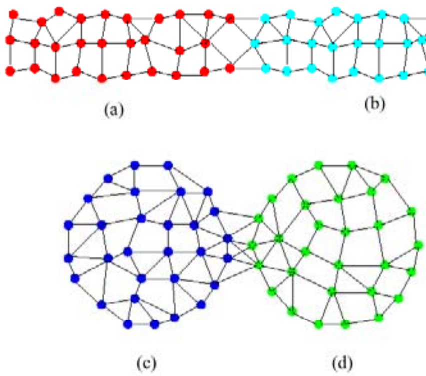
MIN or CURE:

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)

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

Chameleon: Clustering Using Dynamic Modeling

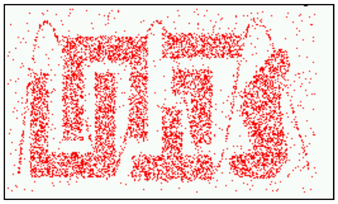
- Adapt to the characteristics of the data set to find the natural clusters
- Use a dynamic model to measure the similarity between clusters
  - Main property is the relative closeness and relative inter-connectivity of the cluster
  - Two clusters are combined if the resulting cluster shares certain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

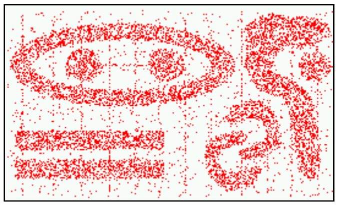
- One of the areas of application is spatial data

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., is a sub-cluster of a “real” cluster

Chameleon: Steps ...

Phase 2: Use Hierarchical Agglomerative Clustering to merge sub-clusters

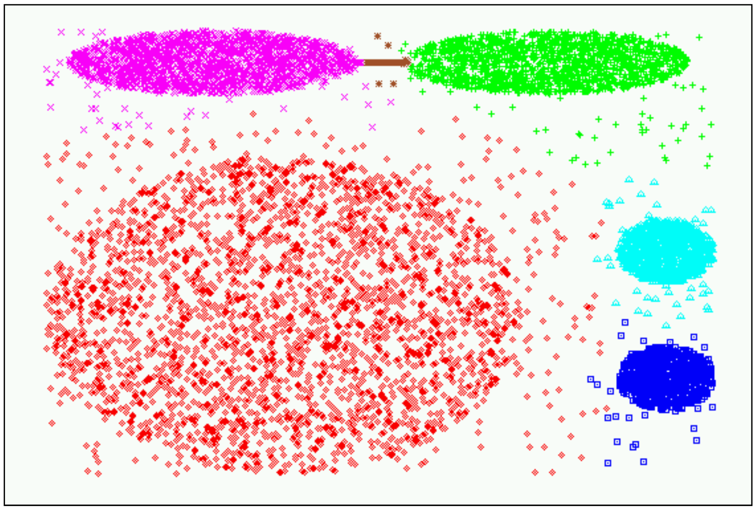
Two clusters are combined if the *resulting cluster shares certain properties with the constituent clusters*

Two key properties used to model cluster similarity:

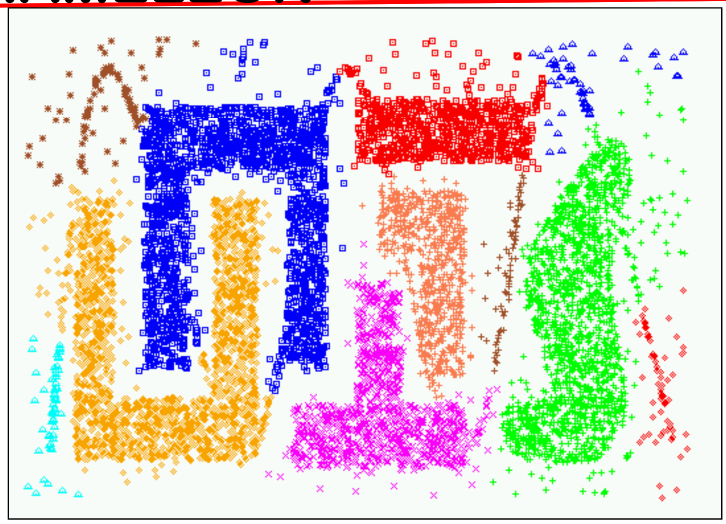
Relative Interconnectivity: Absolute interconnectivity of two clusters normalized by the internal connectivity of the clusters

Relative Closeness: Absolute closeness of two clusters normalized by the internal closeness of the clusters

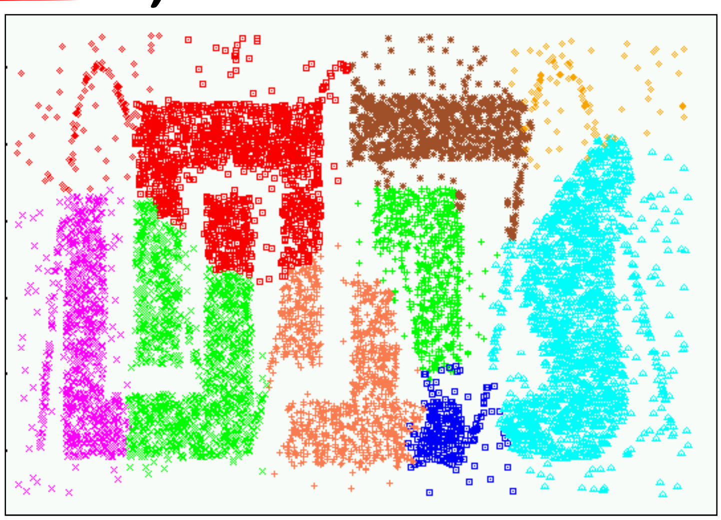
Experimental Results: **CHAMELEON**



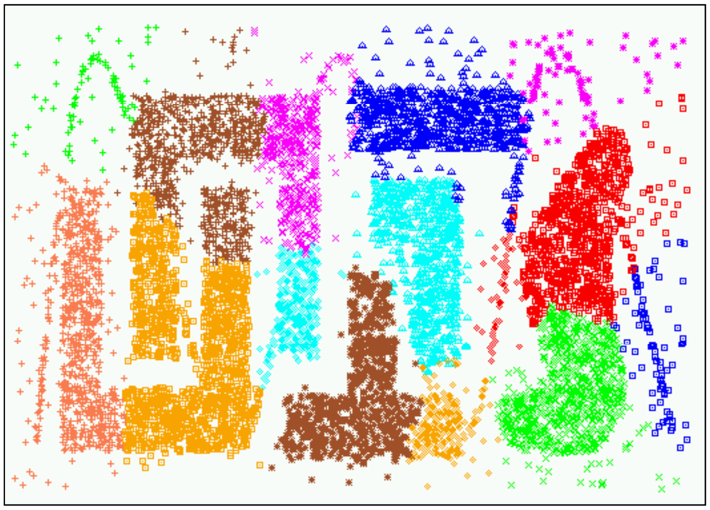
Experimental Results: CHAMELEON



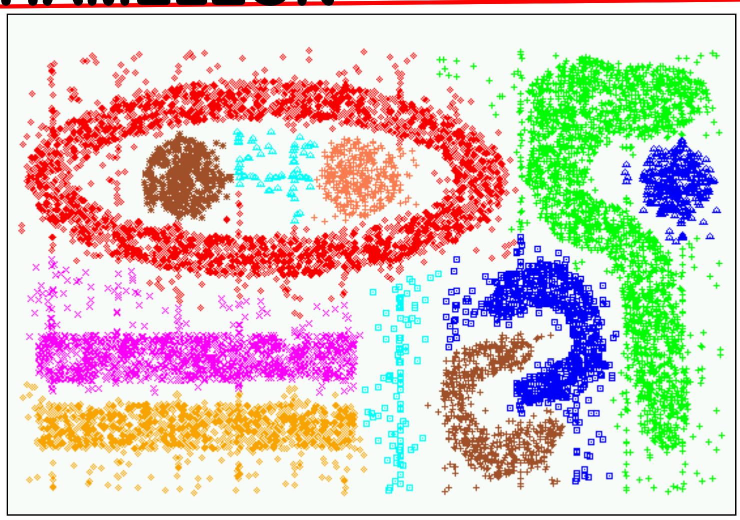
Experimental Results: CURE (*10 clusters*)



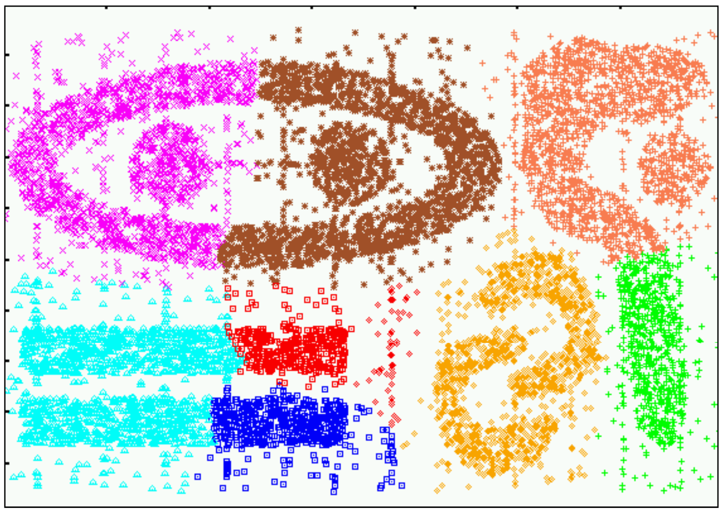
Experimental Results: CURE (*15 clusters*)



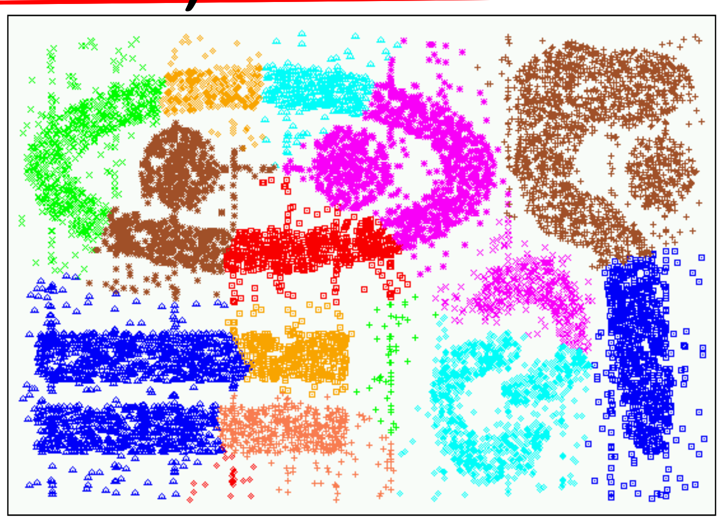
Experimental Results: CHAMELEON



Experimental Results: CURE (*9 clusters*)

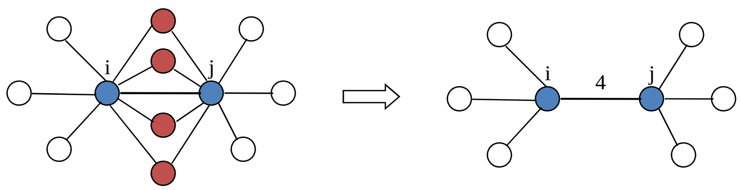


Experimental Results: CURE (*15 clusters*)

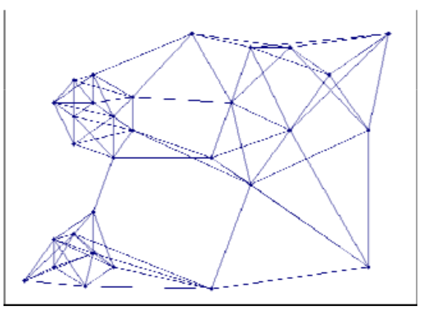


Shared Near Neighbor Approach

SNN graph: the weight of an edge is the number of shared neighbors between vertices given that the vertices are connected

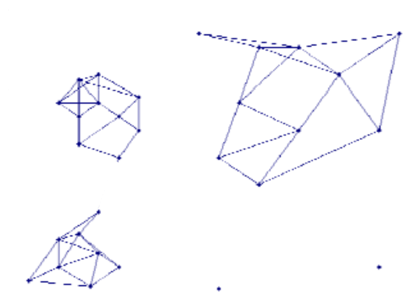


Creating the SNN Graph



Sparse Graph

Link weights are similarities between neighboring points



Shared Near Neighbor Graph

Link weights are number of Shared Nearest Neighbors

Jarvis-Patrick Clustering

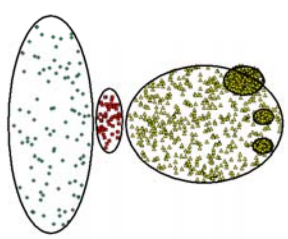
First, the k-nearest neighbors of all points are found
- In graph terms this can be regarded as breaking all but the k strongest links from a point to other points in the proximity graph

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

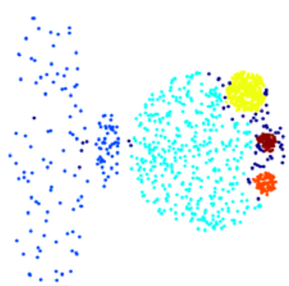
For instance, we might choose a nearest neighbor list of size 20 and put points in the same cluster if they share more than 10 near neighbors

Jarvis-Patrick clustering is too brittle

When Jarvis-Patrick Works Reasonably Well



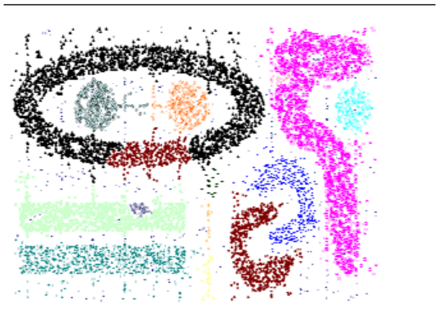
Original Points



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