Statistical Learning and Data Mining
CS 363D/ SSC 358

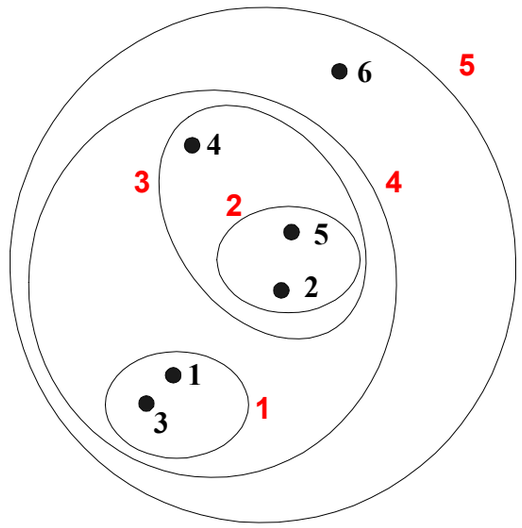
Lecture: Hierarchical Clustering

Prof. Pradeep Ravikumar
pradeepr@cs.utexas.edu

Adapted From: Pang-Ning Tan, Steinbach, Kumar

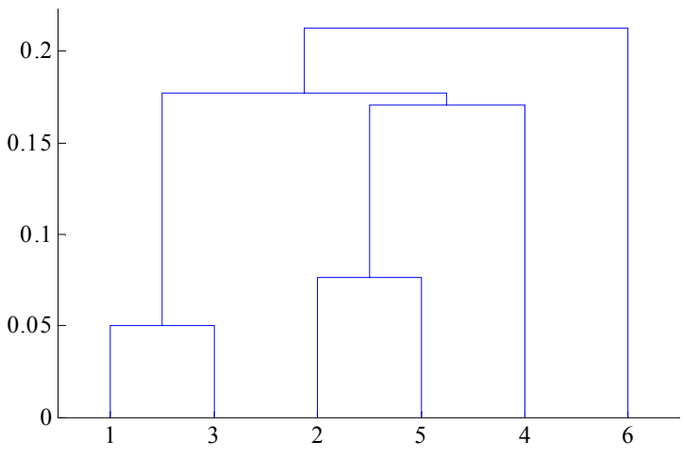
Hierarchical Clustering

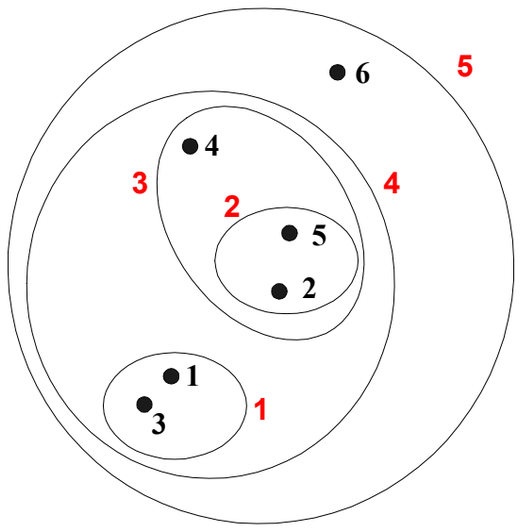
- Produces a set of nested clusters organized as a hierarchical tree



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

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)

Hierarchical Clustering

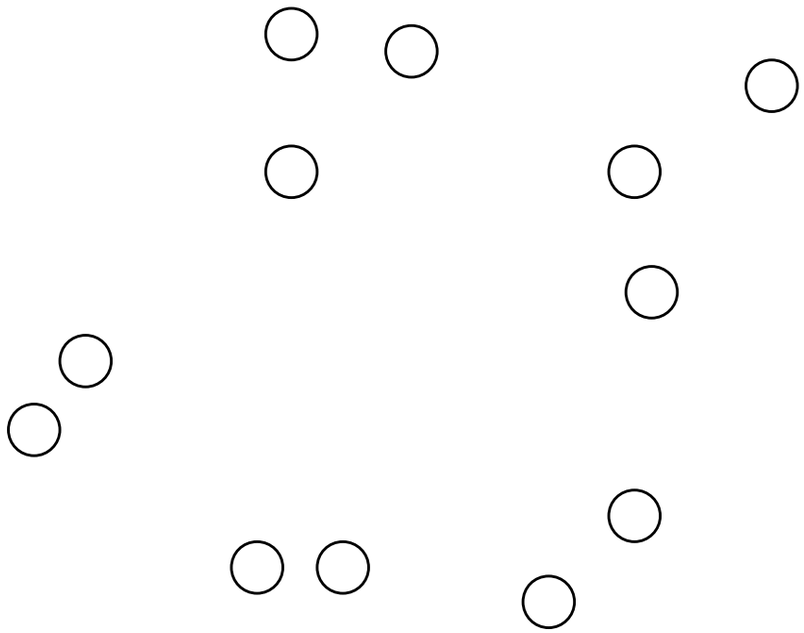
- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster 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

- 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

Agglomerative Clustering: Initially

- Start with clusters of individual points and a proximity matrix

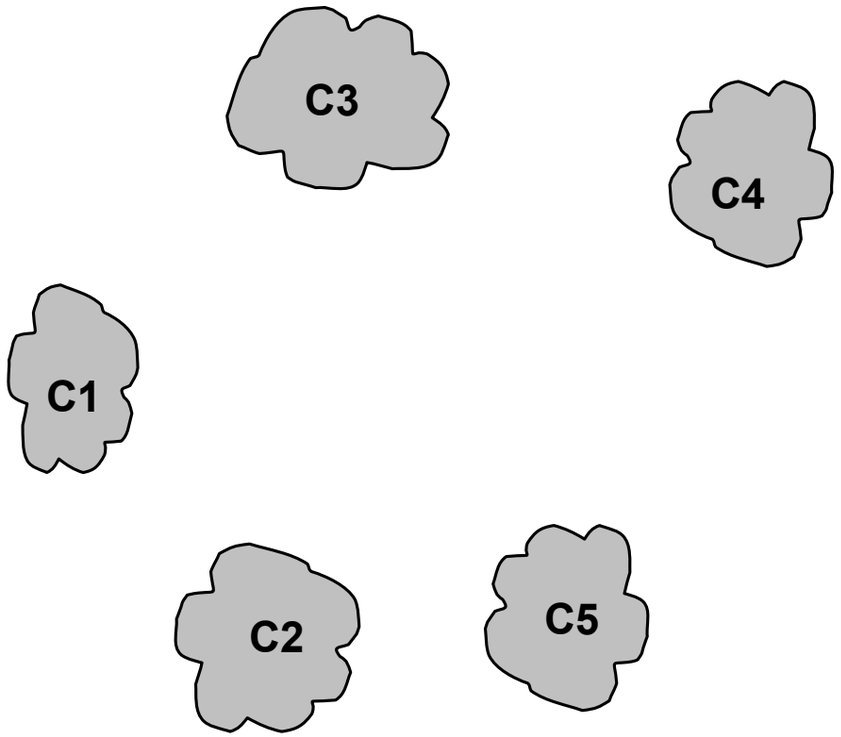


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

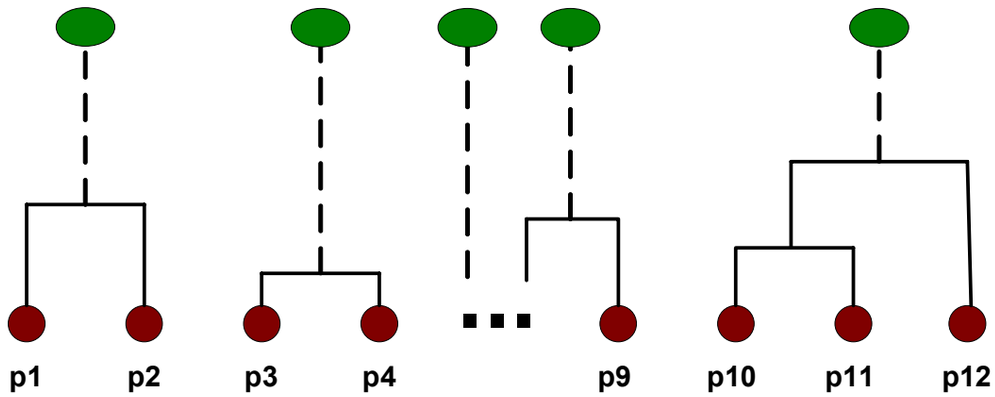
Agglomerative Clustering: Middle

- After some merging steps, we have some clusters



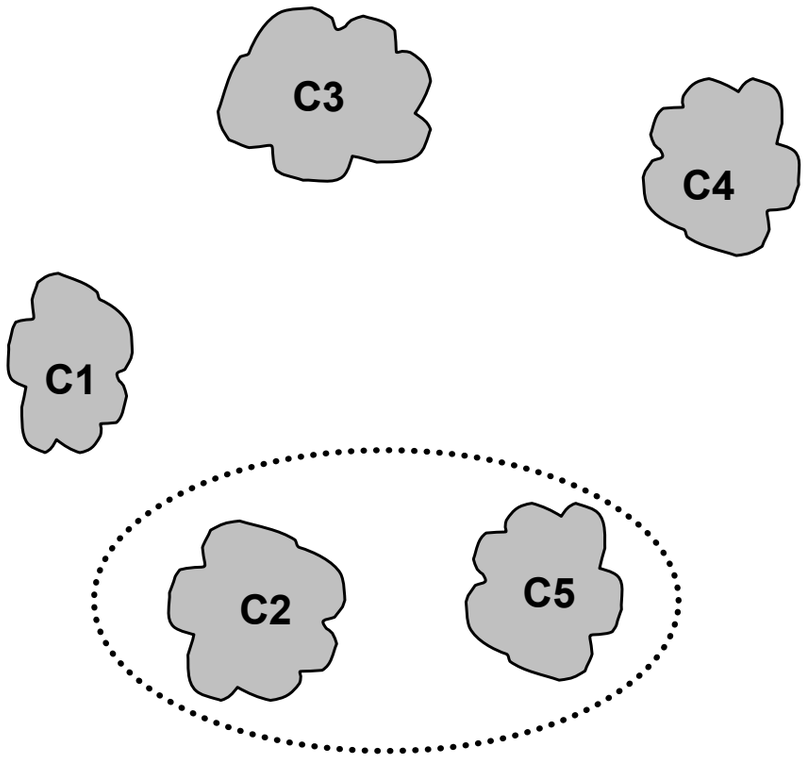
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

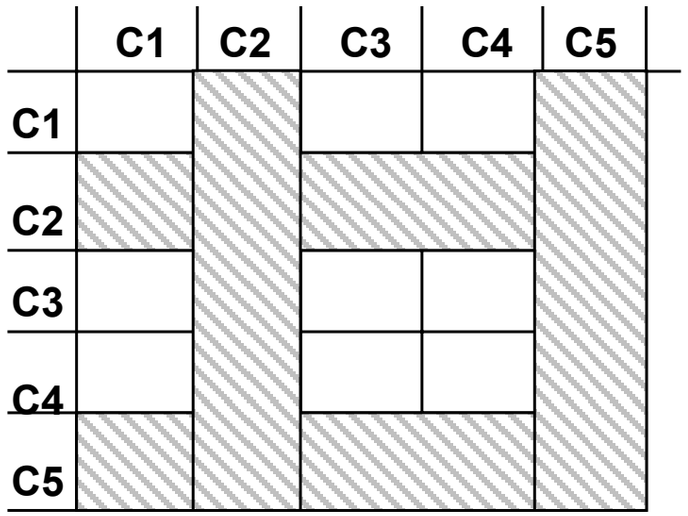
Proximity Matrix



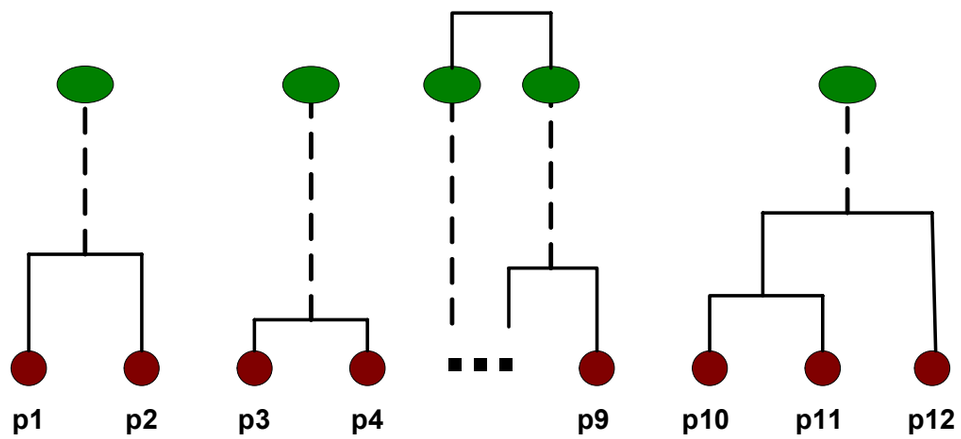
Agglomerative Clustering: Middle

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



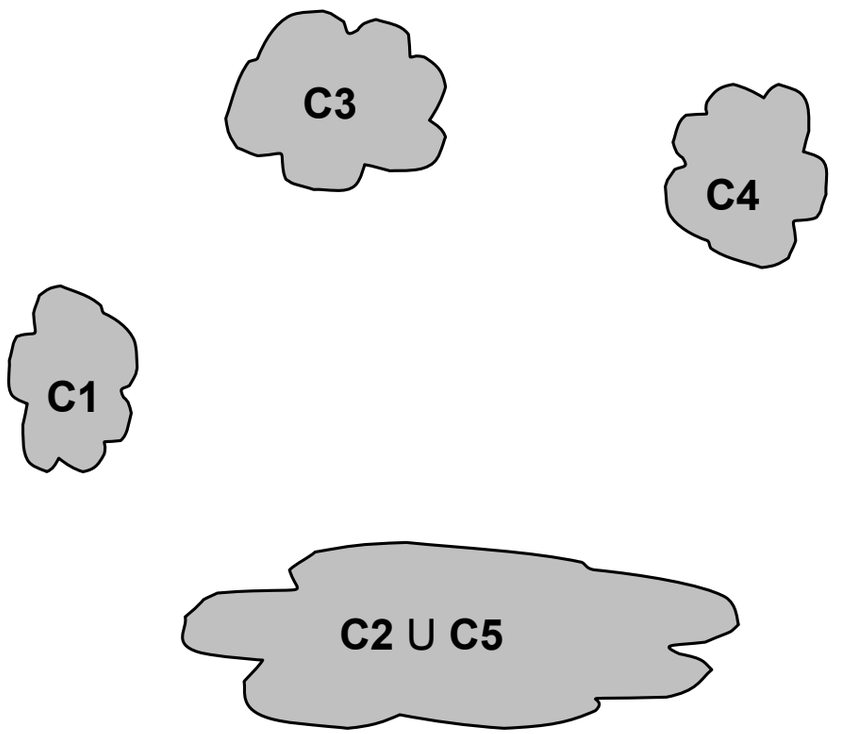


Proximity Matrix



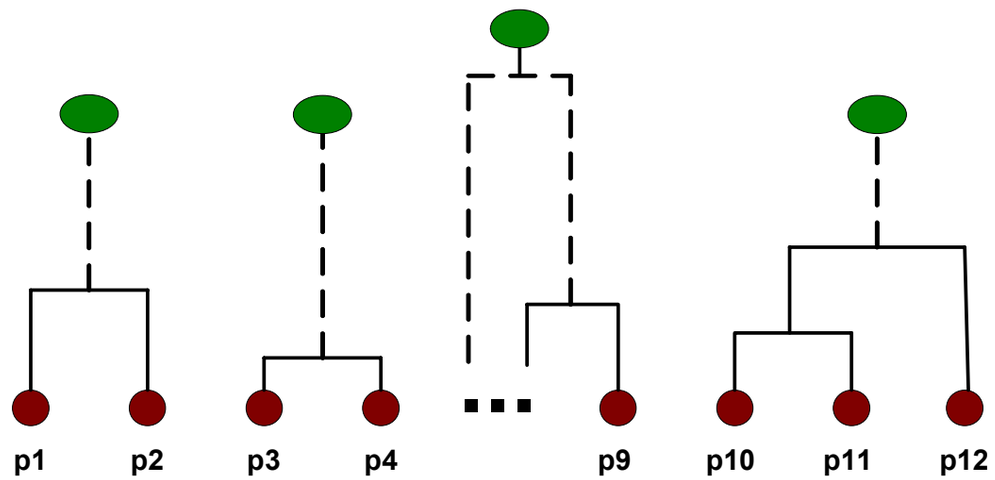
Agglomerative Clustering: After Merging

- The question is “How do we update the proximity matrix?”

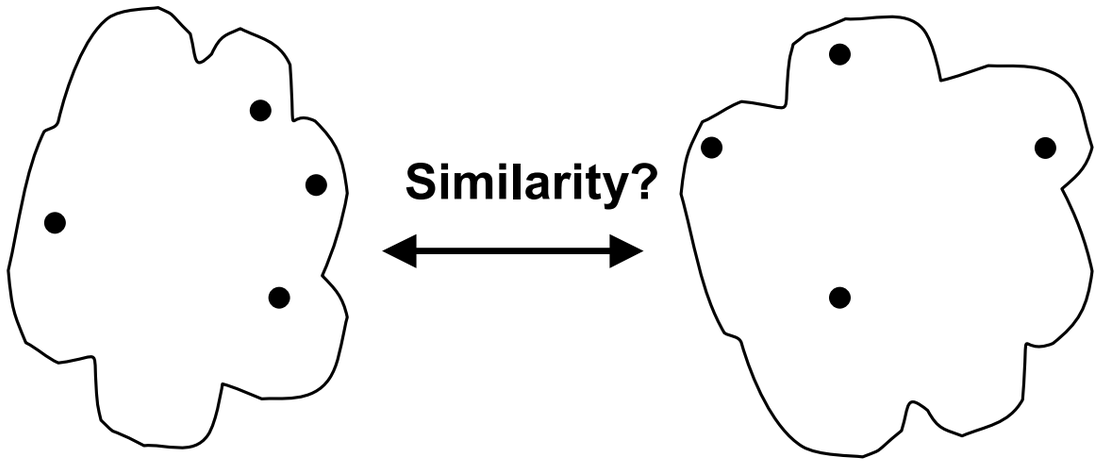


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



How to define Inter-cluster similarity

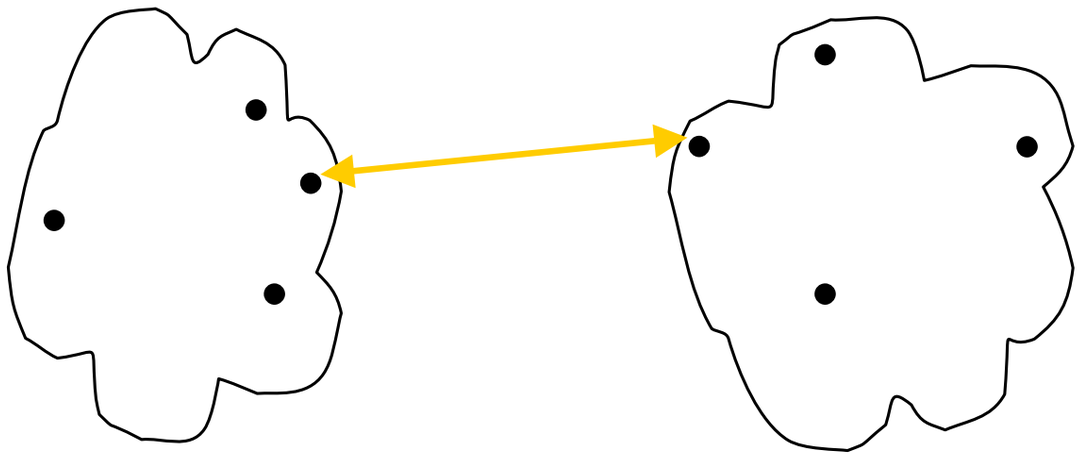


- MIN
- MAX
- Group Average
- Distance Between Centroids

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

Proximity Matrix

How to define Inter-cluster similarity

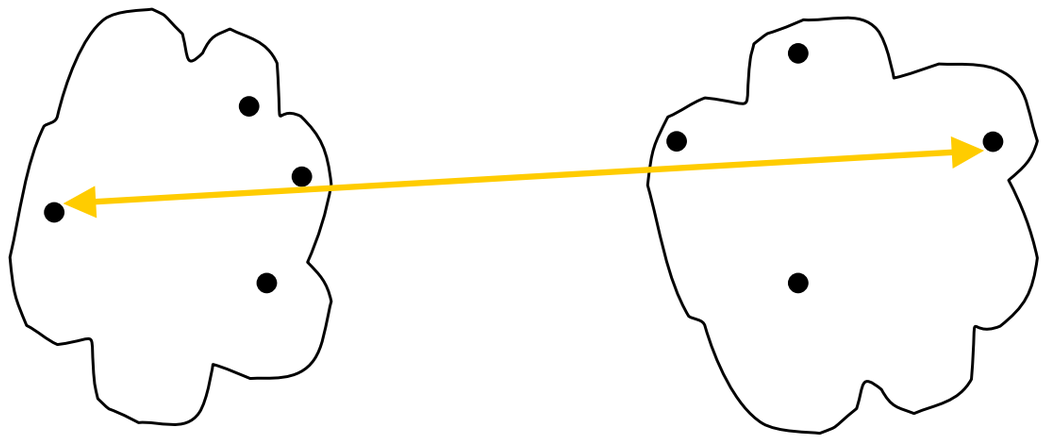


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids

How to define Inter-cluster similarity

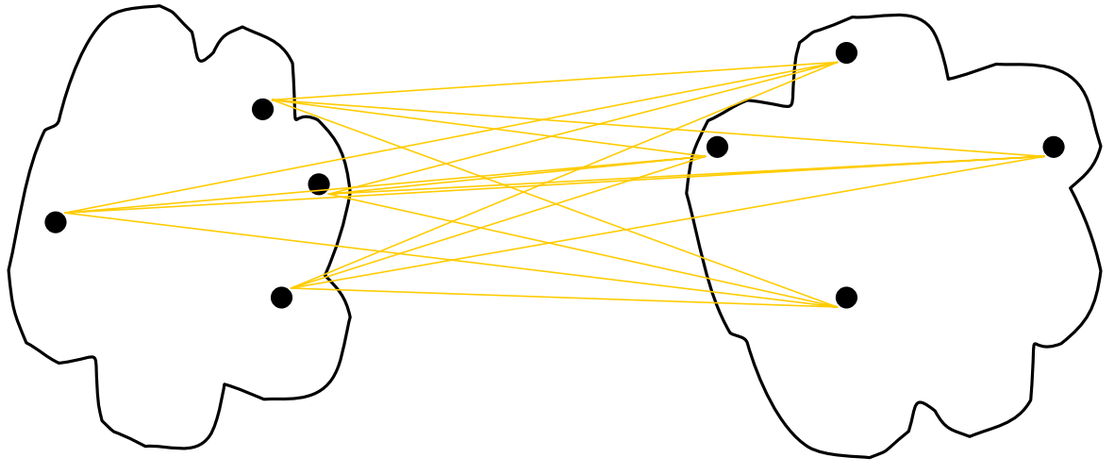


- MIN
- MAX
- Group Average
- Distance Between Centroids

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

Proximity Matrix

How to define Inter-cluster similarity

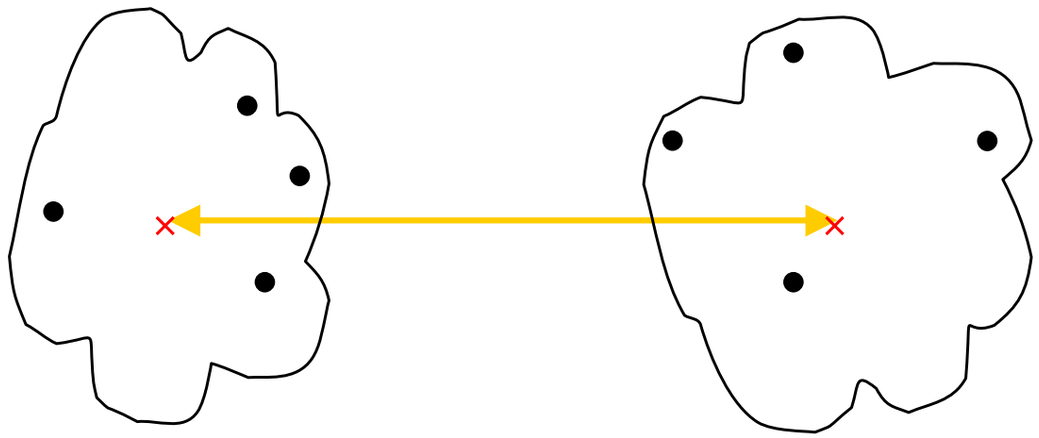


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids

How to define Inter-cluster similarity



- MIN
- MAX
- Group Average
- Distance Between Centroids

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

Proximity Matrix

Cluster Similarity: MIN

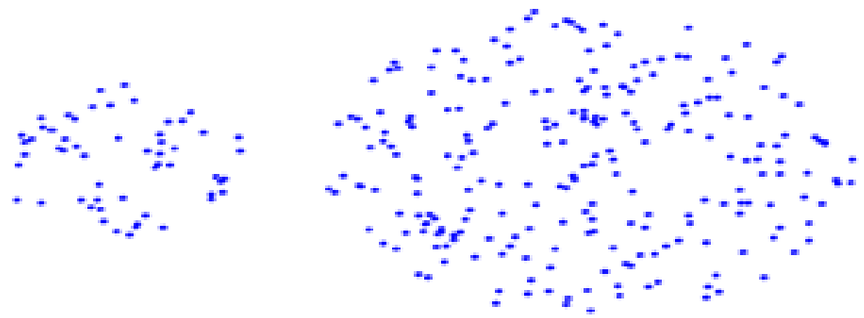
- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.

Cluster Similarity: MIN

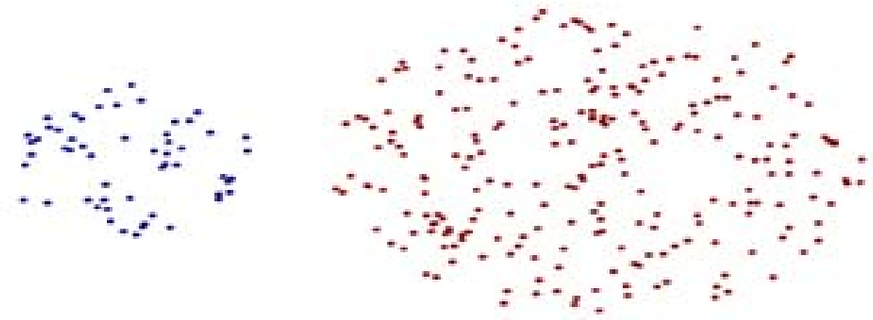
- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.

Also known as **Single Link**

Hierarchical Clustering, MIN: Strengths



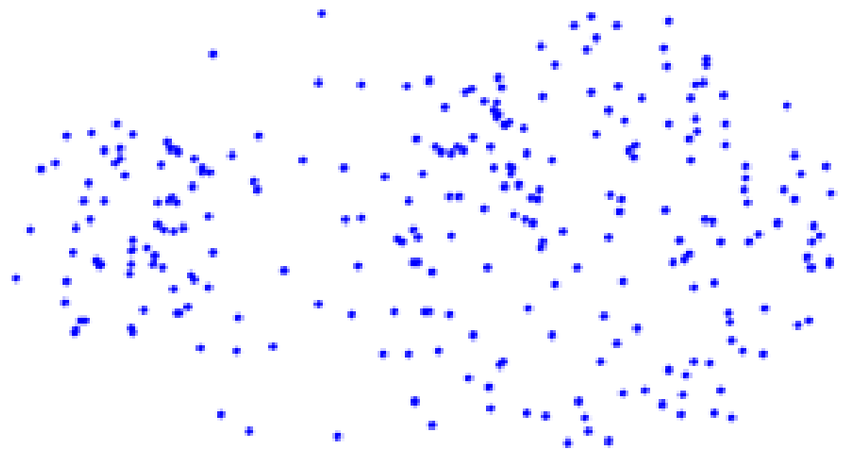
Original Points



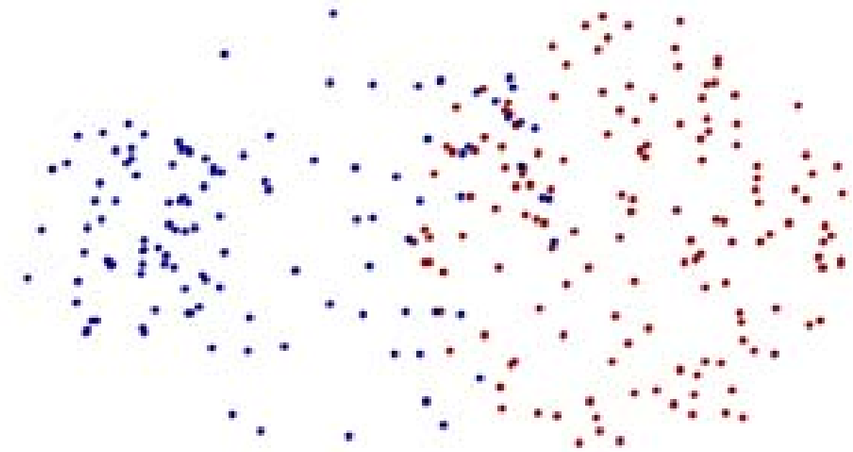
Two Clusters

- **Can handle non-elliptical shapes**

Hierarchical Clustering, MIN: Limitations



Original Points



Two Clusters

- **Sensitive to noise and outliers**

Cluster Similarity: MAX

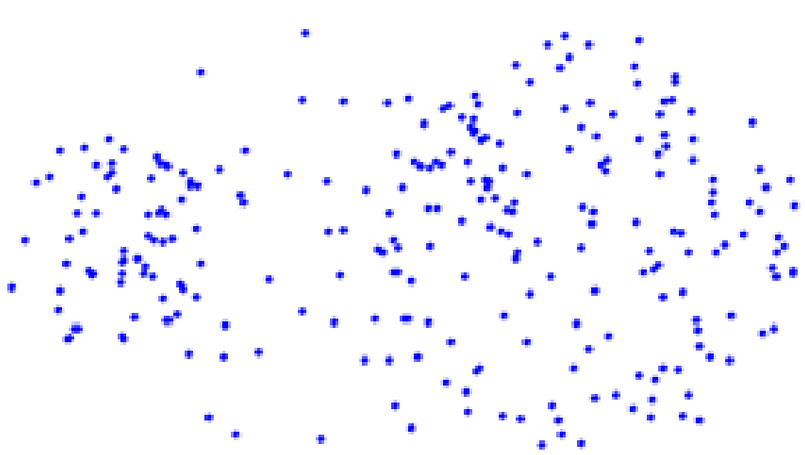
- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters
  - Determined by all pairs of points in the two clusters

Cluster Similarity: MAX

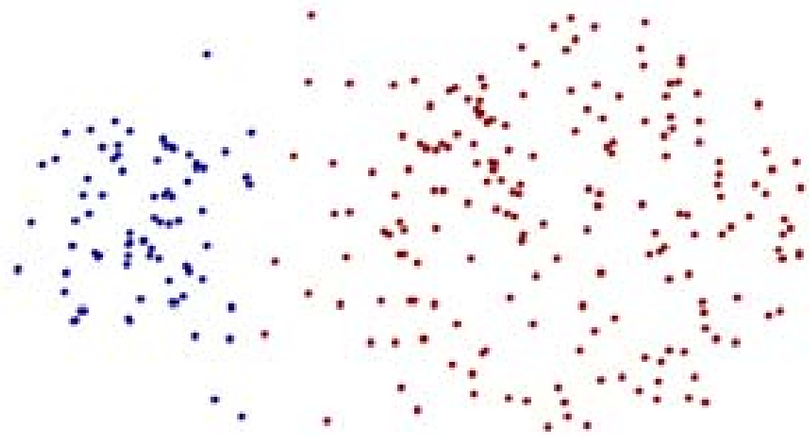
- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters
  - Determined by all pairs of points in the two clusters

Also known as **Complete Link**

Hierarchical Clustering, MAX: Strengths



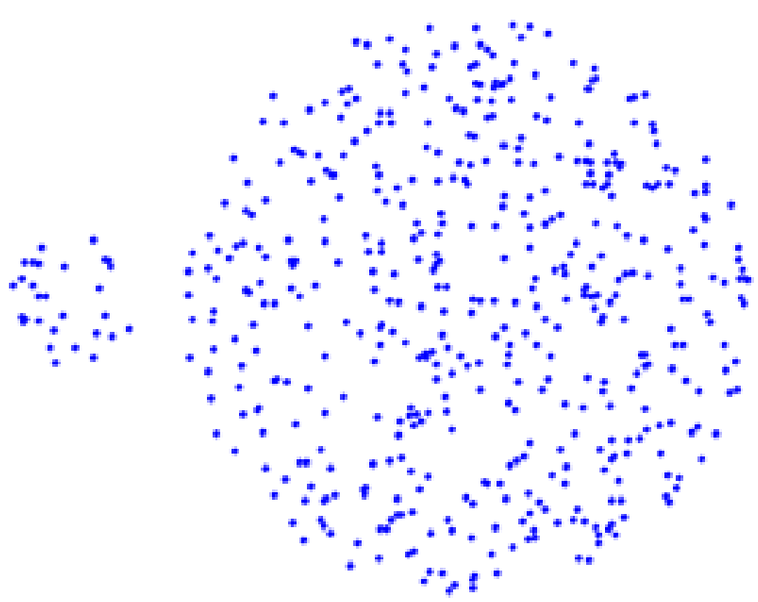
Original Points



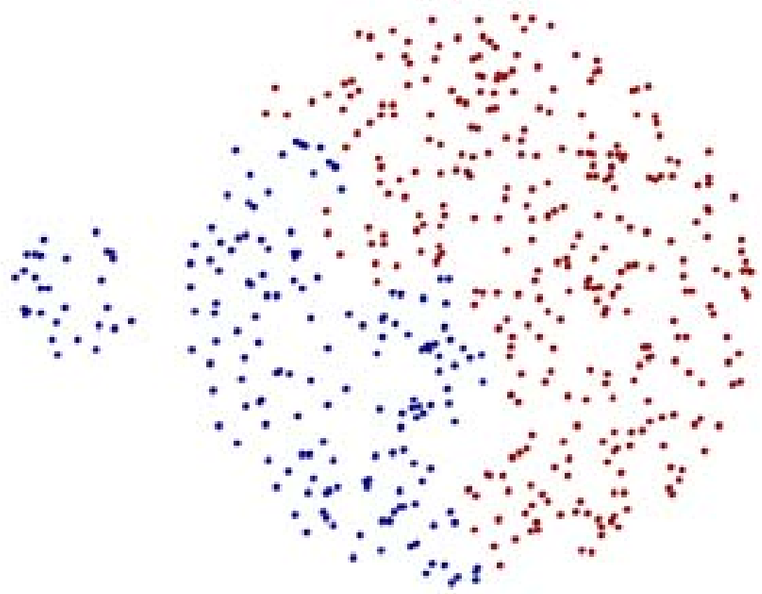
Two Clusters

- **Less susceptible to noise and outliers**

Hierarchical Clustering, MAX: Limitations



Original Points



Two Clusters

- Tends to break large clusters

Cluster Similarity: Group Average

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

$$\textbf{proximity}(\textbf{Cluster}_i, \textbf{Cluster}_j) = \frac{\sum_{\substack{\mathbf{p}_i \in \textbf{Cluster}_i \\ \mathbf{p}_j \in \textbf{Cluster}_j}} \textbf{proximity}(\mathbf{p}_i, \mathbf{p}_j)}{|\textbf{Cluster}_i| * |\textbf{Cluster}_j|}$$

- Compromise between Single and Complete Link

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

Hierarchical Clustering: Problems and Limitations

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

- No objective function is directly minimized

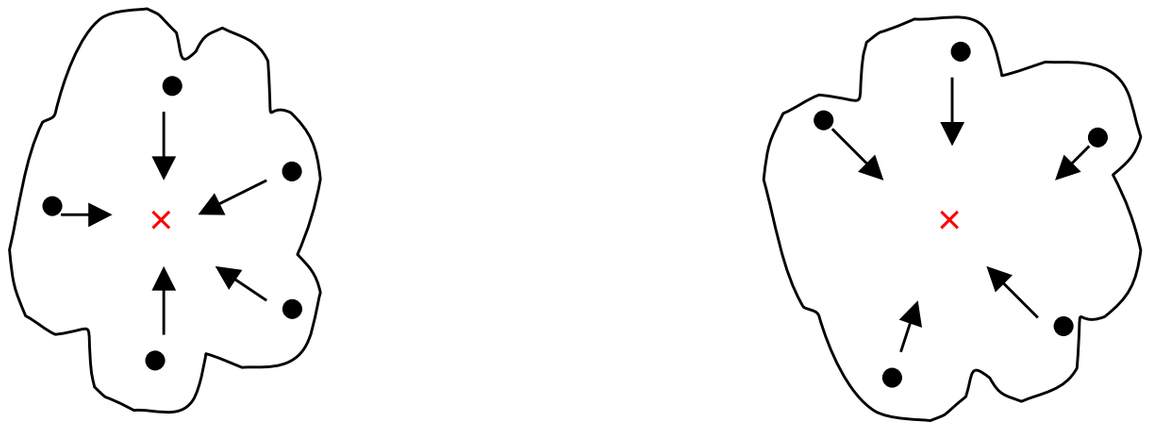
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Difficulty handling different sized clusters and convex shapes
  - Breaking large clusters

Hierarchical Clustering, CURE

- 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: A Hierarchical Clustering Variant

- Uses a number of points to represent a cluster



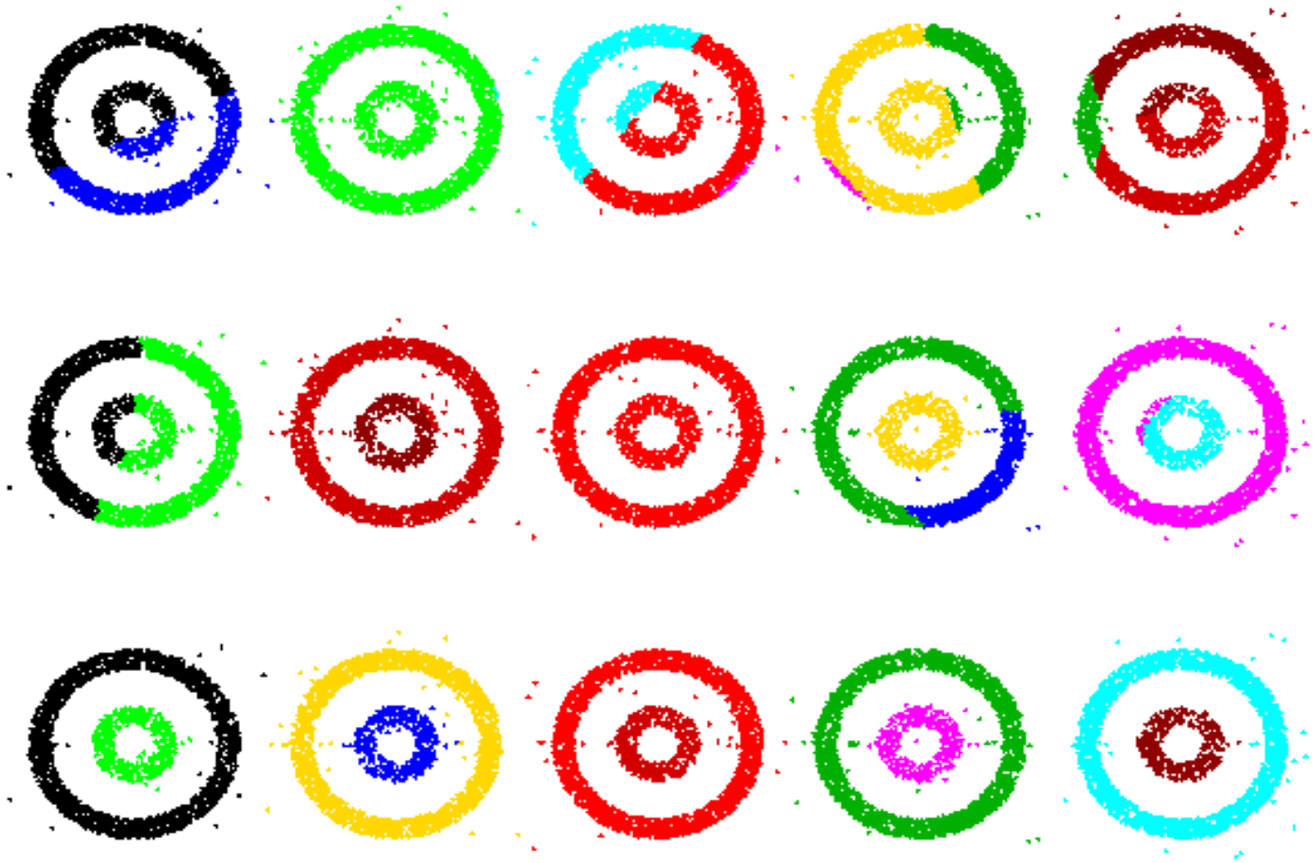
- Representative points are found by selecting a constant number of points from a cluster and then “shrinking” them toward the center of the cluster
- Cluster similarity is the similarity of the closest pair of representative points from different clusters

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

(centroid)

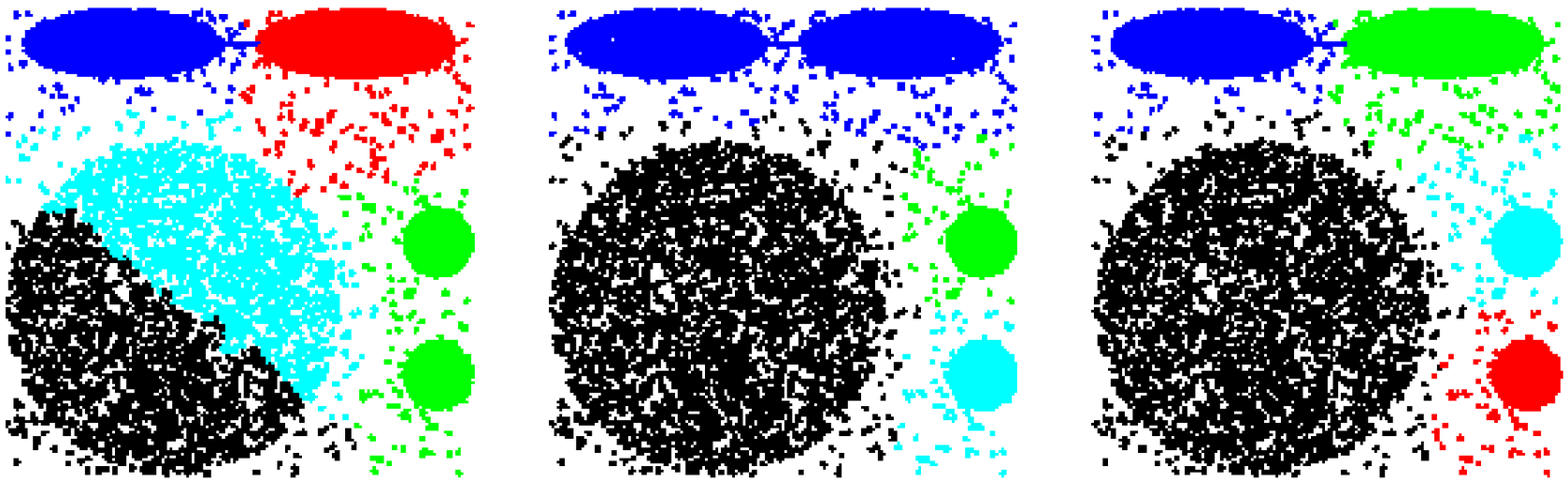
b) MST METHOD

(single link)

c) CURE

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

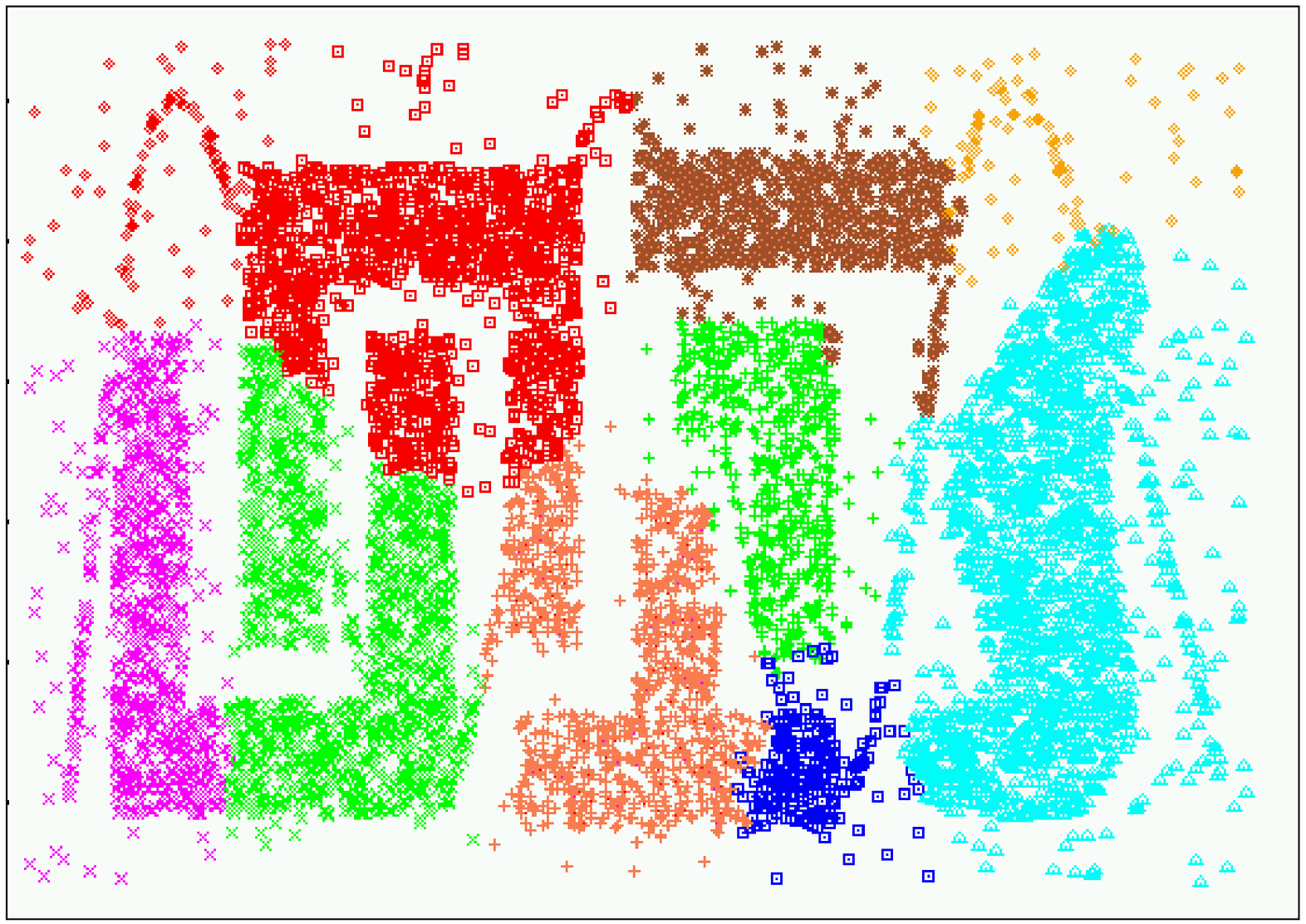
Experimental Results: CURE



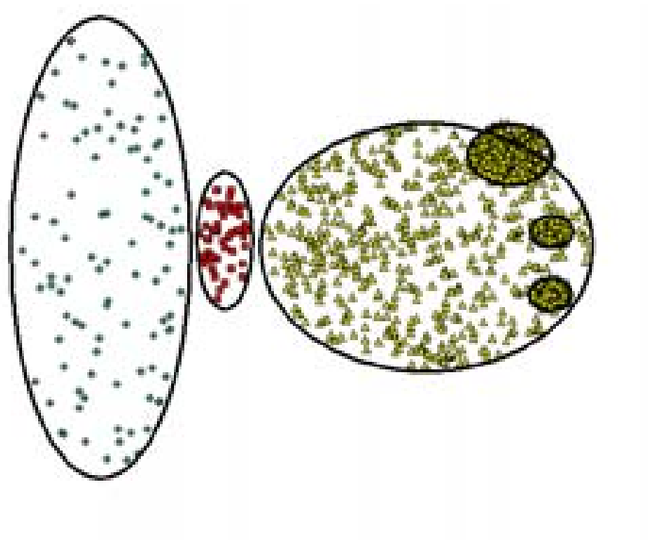
a) BIRCH b) MST METHOD c) CURE

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

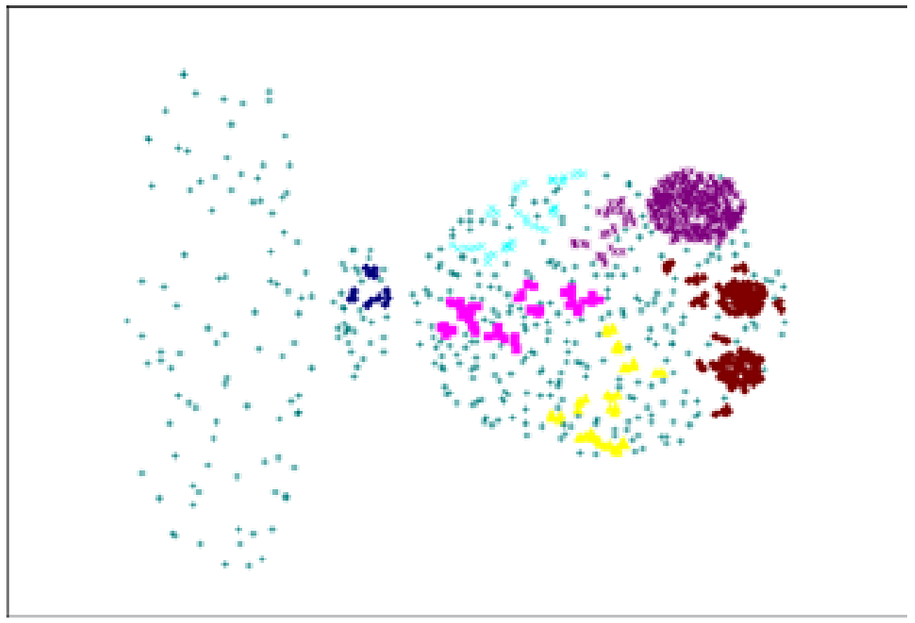
Experimental Results: CURE, 10 Clusters



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

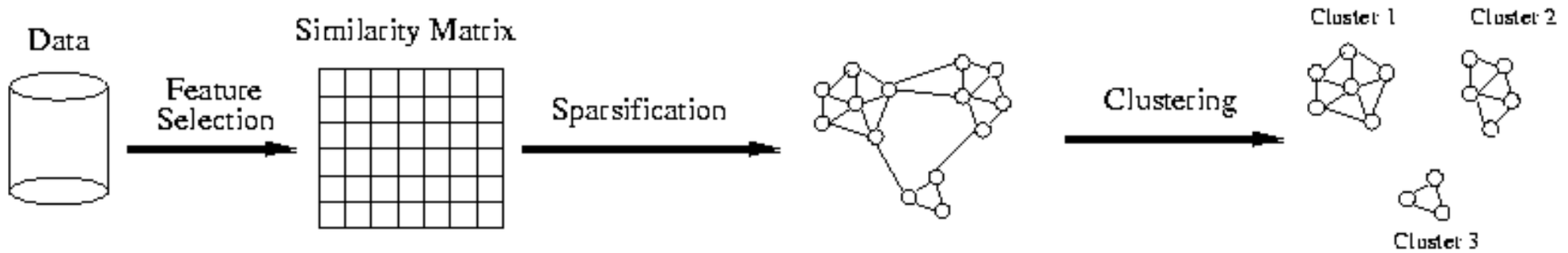
- The amount of data that needs to be processed is drastically reduced
  - Sparsification can eliminate more than 99% of the entries in a proximity matrix
  - The amount of time required to cluster the data is drastically reduced
  - The size of the problems that can be handled is increased

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.

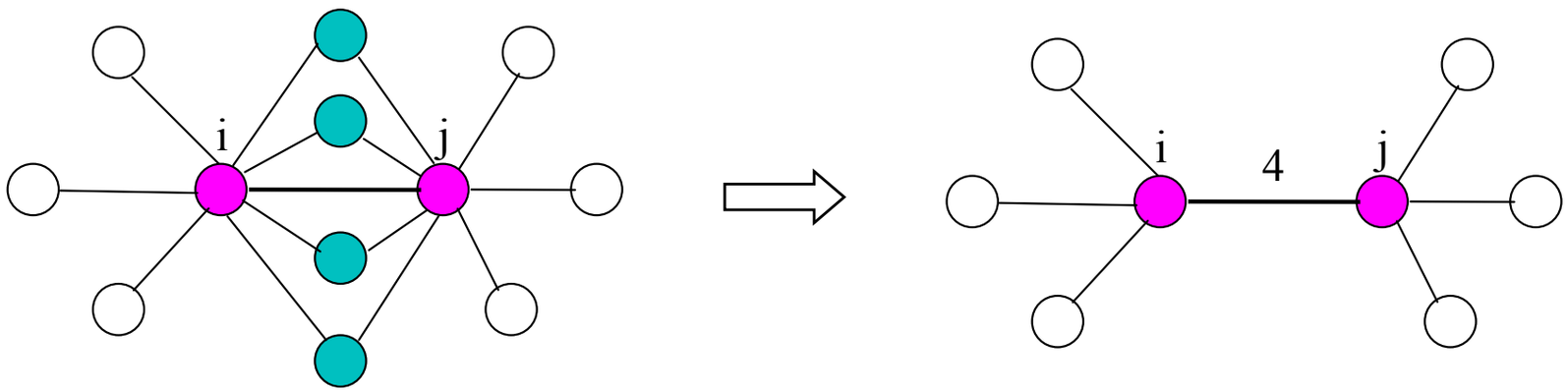
- Sparsification facilitates the use of graph partitioning algorithms (or algorithms based on graph partitioning algorithms.

Graph-based Clustering: Sparsification ...

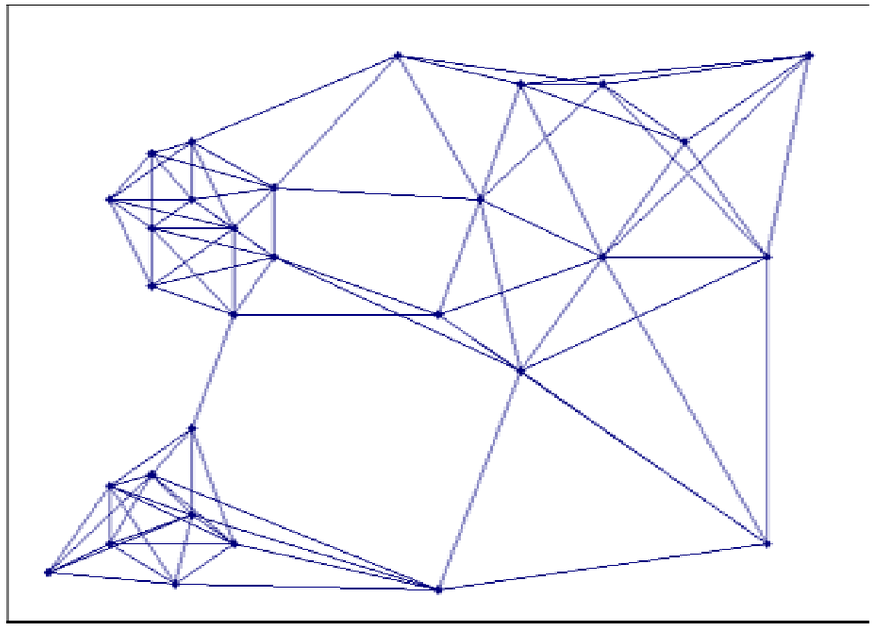


Shared Nearest Neighbor (SNN) Clustering

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

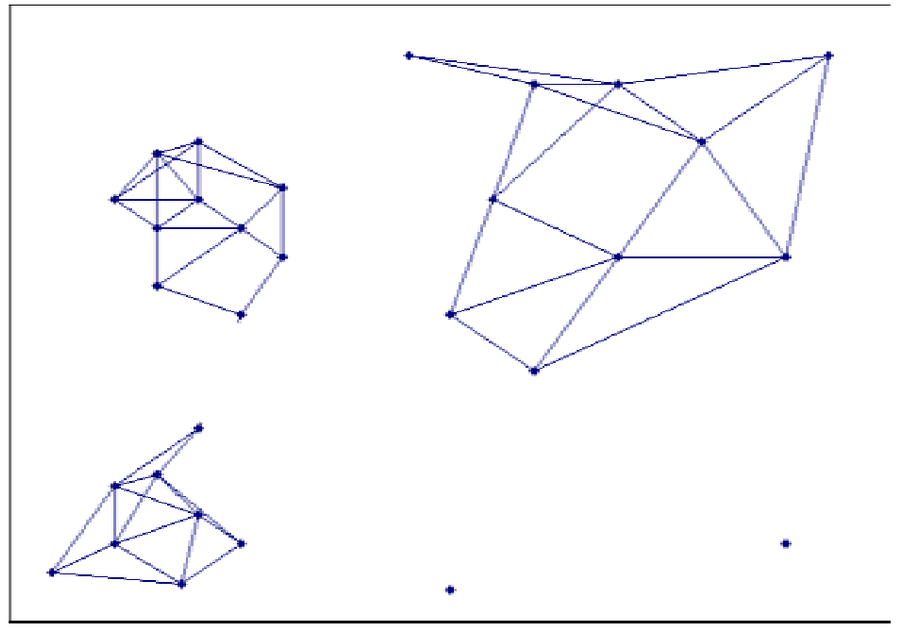


Creating the SNN Graph



Sparse Graph

Link weights are similarities between neighboring points



Shared Near Neighbor Graph

Link weights are number of Shared Nearest Neighbors

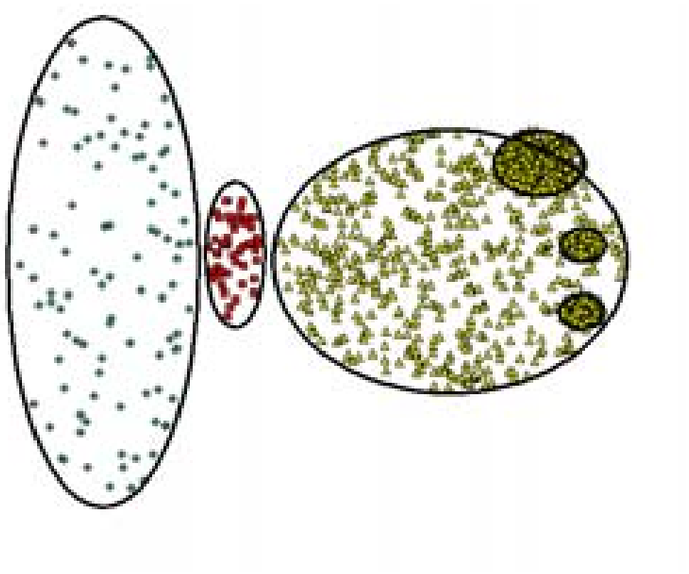
SNN Clustering

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

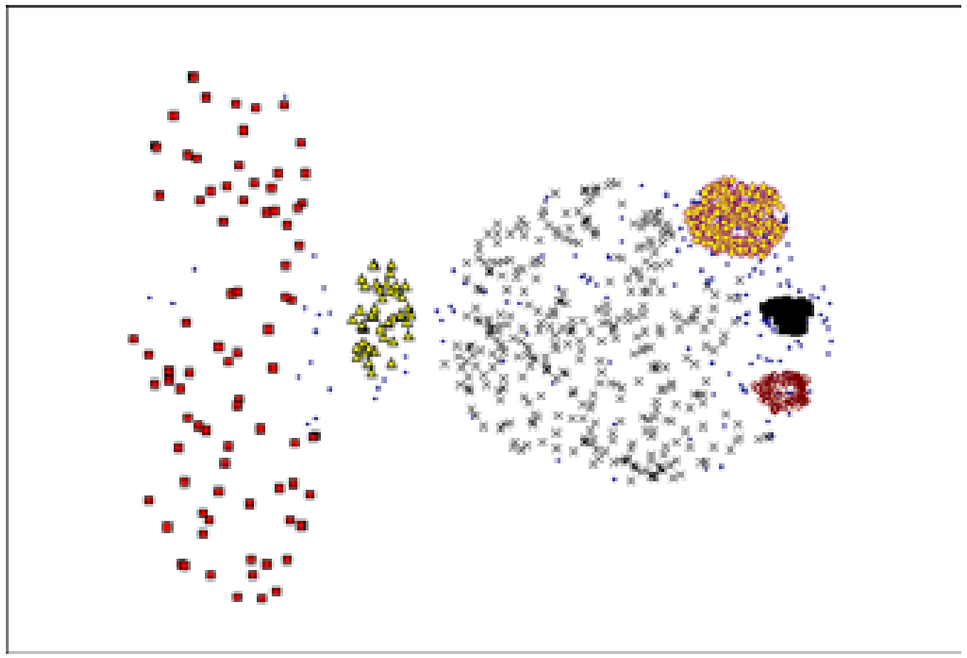
2. **Sparsify the similarity matrix by keeping only the *k* most similar neighbors**
   This corresponds to only keeping the *k* strongest links of the similarity graph

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

SNN Clustering: Differing Densities



Original Points



SNN Clustering

SNN Clustering: Other difficult scenarios