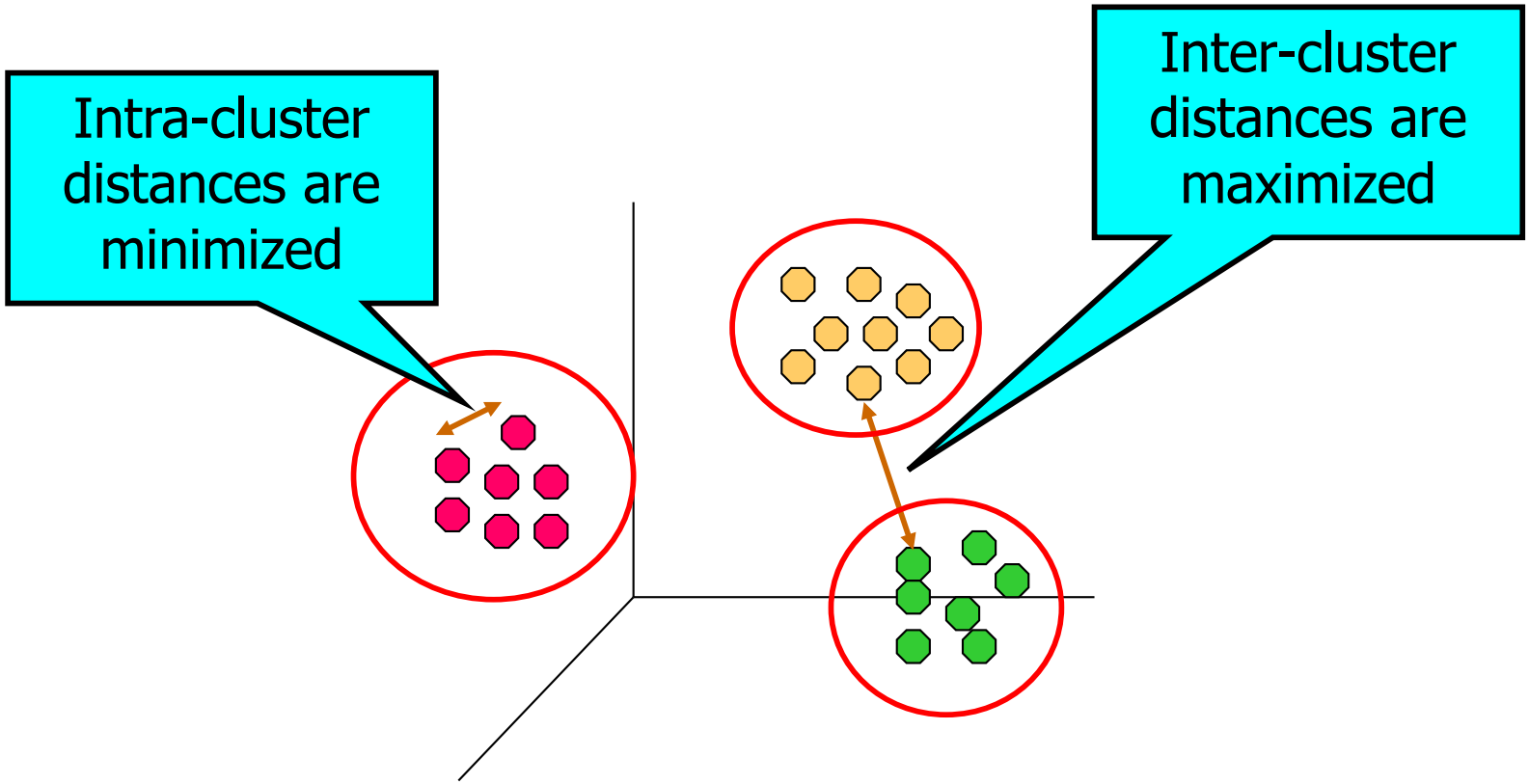
PENNSTATE

Chapter 8-9: Cluster Analysis

Presentation extended from the slides of the textbook, Introduction to Data Mining by Tan et al. and supplementary material

What is Cluster Analysis?

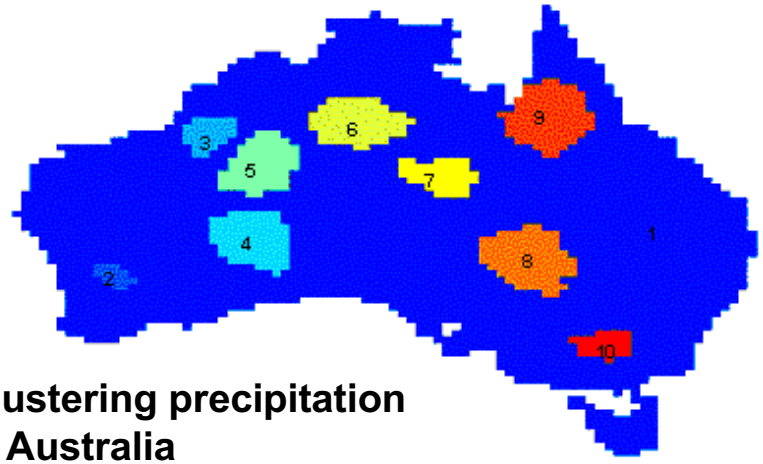
- Finding groups of objects such that the objects in a group will be *similar* (or *related*) to one another and different from (or unrelated to) the objects in other groups



Applications of Cluster Analysis

- Understanding
  - Group related articles for browsing
  - group genes and proteins that have similar functionality
  - group stocks with similar price fluctuations
- Summarization
  - Reduce the size of large data sets

	Discovered Clusters	*Industry Group*
1	Applied-Matl-DOWN,Bay-Network-Down,3-COM-DOWN, Cabletron-Sys-DOWN,CISCO-DOWN,HP-DOWN, DSC-Comm-DOWN,INTEL-DOWN,LSI-Logic-DOWN, Micron-Tech-DOWN,Texas-Inst-Down,Tellabs-Inc-Down, Natl-Semiconduct-DOWN,Oracl-DOWN,SGI-DOWN, Sun-DOWN	Technology1-DOWN
2	Apple-Comp-DOWN,Autodesk-DOWN,DEC-DOWN, ADV-Micro-Device-DOWN,Andrew-Corp-DOWN, Computer-Assoc-DOWN,Circuit-City-DOWN, Compaq-DOWN, EMC-Corp-DOWN, Gen-Inst-DOWN, Motorola-DOWN,Microsoft-DOWN,Scientific-Atl-DOWN	Technology2-DOWN
3	Fannie-Mae-DOWN,Fed-Home-Loan-DOWN, MBNA-Corp-DOWN,Morgan-Stanley-DOWN	Financial-DOWN
4	Baker-Hughes-UP,Dresser-Inds-UP,Halliburton-HLD-UP, Louisiana-Land-UP,Phillips-Petro-UP,Unocal-UP, Schlumberger-UP	Oil-UP



Clustering precipitation in Australia

What is NOT Cluster Analysis?

- Supervised classification
  - Have class label information
- Simple segmentation
  - Dividing students into different registration groups alphabetically, by last name
- Graph partitioning
  - Some mutual relevance and synergy, but areas are not identical
- Group query
  - Groupings are a result of an external specification (e.g., Group by in SQL)

Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

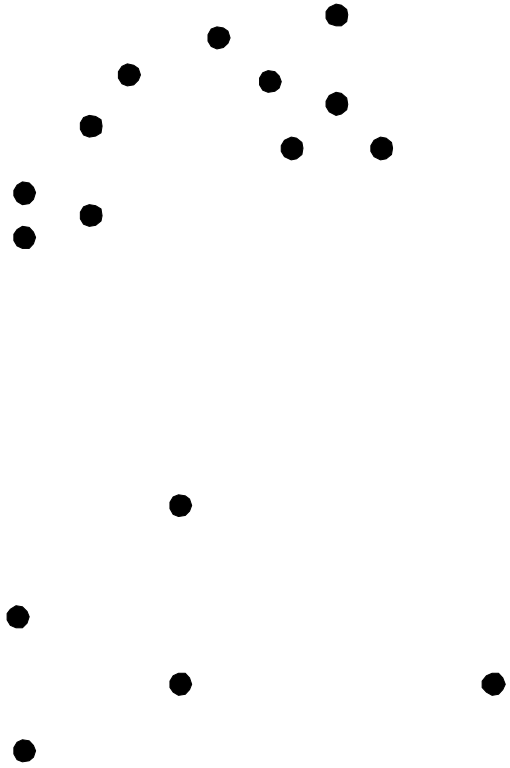
Two Clusters

Four Clusters

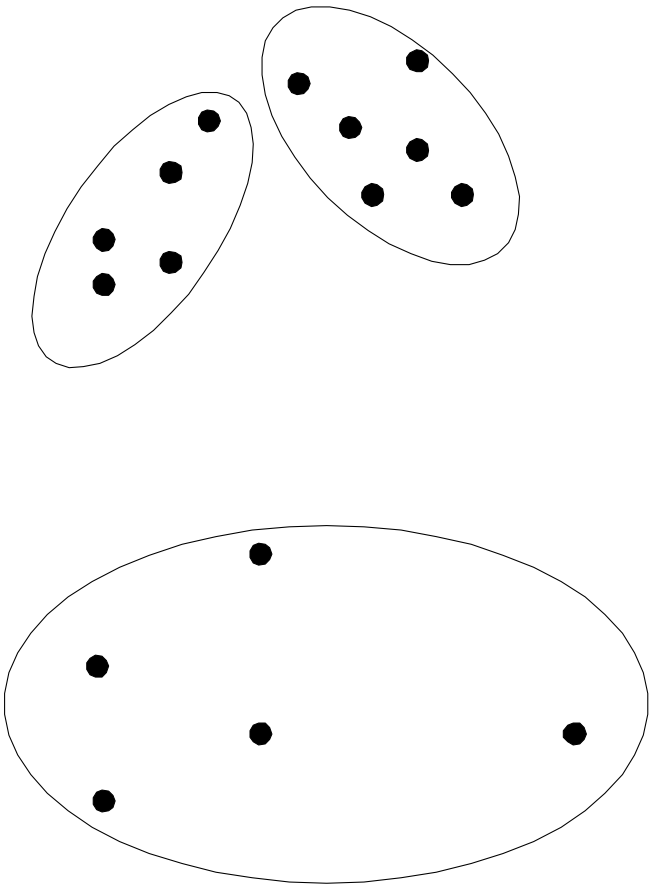
Types of Clusterings

- A *clustering* is a set of clusters
- Important distinction between *hierarchical (nested)* and *(flat) partitional* sets of clusters
- Partitional Clustering
  - A division of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- Hierarchical clustering
  - A set of nested clusters organized as a hierarchical tree

Partitional Clustering

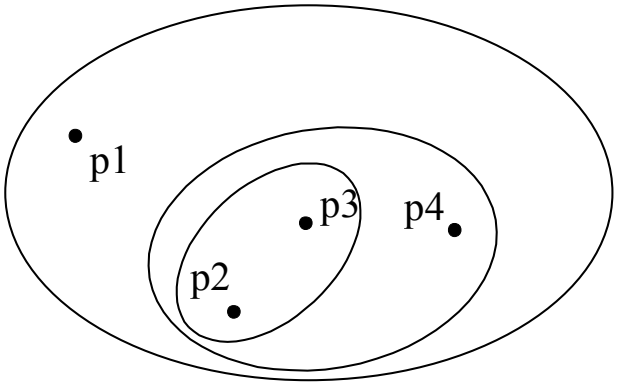


Original Points

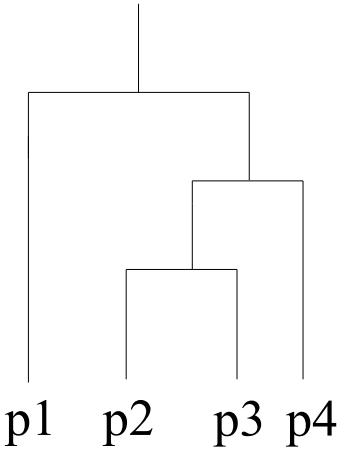


A Partitional Clustering

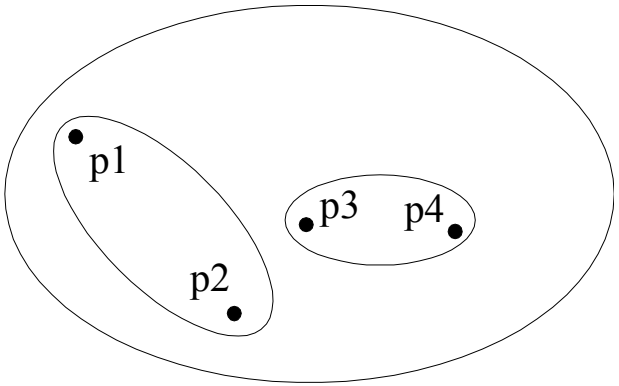
Hierarchical Clustering



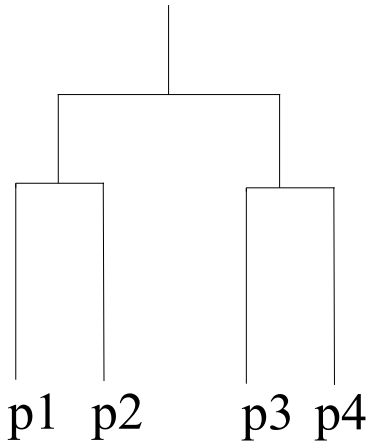
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

Other Distinctions Between Clusterings

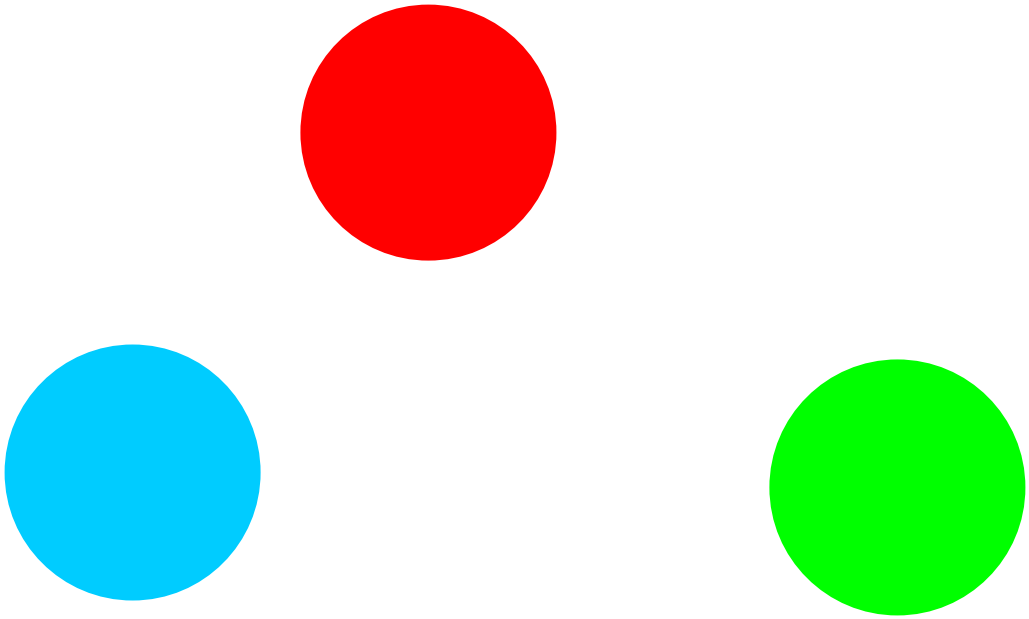
- Exclusive versus Non-exclusive
  - In non-exclusive clusterings, points may belong to multiple clusters.
  - Points may present in multiple classes or as border points
- Fuzzy versus Non-fuzzy
  - In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
  - Weights must sum to 1
  - Probabilistic clustering has similar characteristics
- Partial versus Complete
  - In some cases, we only want to cluster some of the data
- Heterogeneous versus Homogeneous
  - Cluster of widely different sizes, shapes, and densities

Types of Clusters

- Well-separated clusters
- Center-based clusters
- Contiguous clusters
- Density-based clusters
- Property or Conceptual
- Described by an Objective Function

Types of Clusters: Well-Separated

- Well-Separated Clusters:
  - A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.



3 well-separated clusters

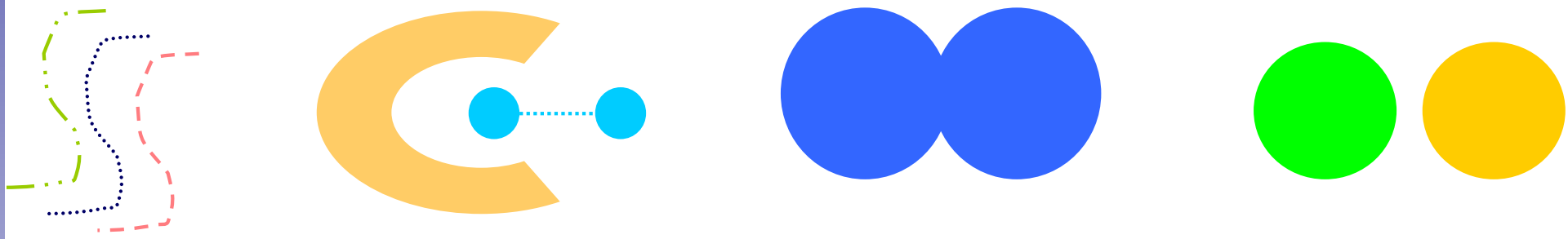
Types of Clusters: Center-Based

- Center-based
  - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the "center" of a cluster, than to the center of any other cluster
  - The center of a cluster is often a *centroid*, the average of all the points in the cluster, or
  - a *medoid*, the most "representative" point of a cluster

4 center-based clusters

Types of Clusters: Contiguity-Based

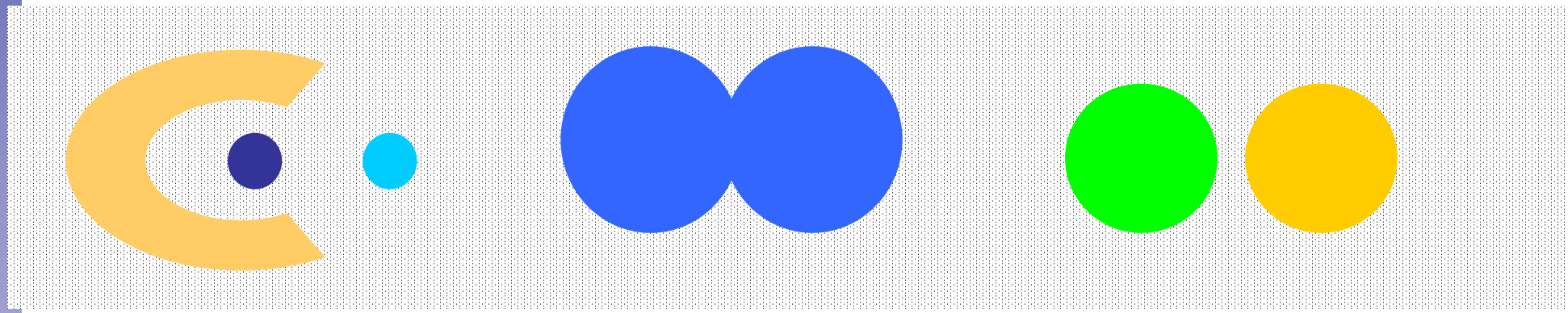
- Contiguous Cluster (Nearest neighbor or Transitive)
  - A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.



8 contiguous clusters

Types of Clusters: Density-Based

- Density-based
  - A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
  - Used when the clusters are irregular or intertwined, and when noise and outliers are present.

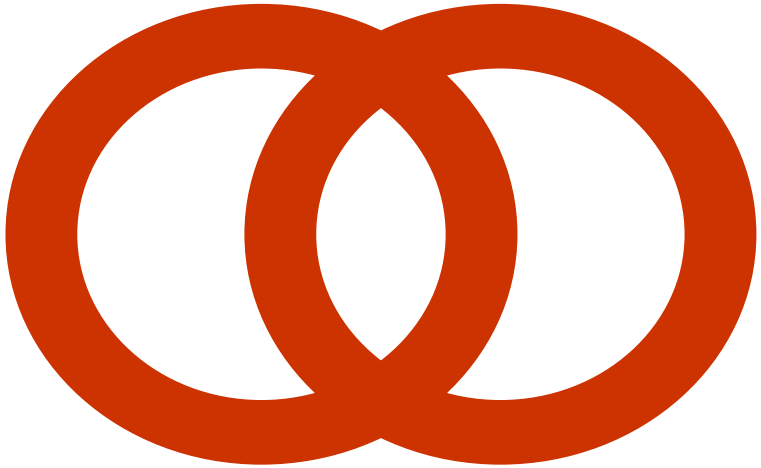


6 density-based clusters

Types of Clusters: Conceptual Clusters

- Shared Property or Conceptual Clusters
  - Finds clusters that share some common property or represent a particular concept.

.



2 Overlapping Circles

Types of Clusters: Objective Function

- Clusters defined by an objective function
  - Finds clusters that minimize/maximize an obj. function.
  - Enumerate all possible ways of dividing the points into clusters and evaluate the *goodness* of each potential clustering using the given objective function. (NP Hard)
  - Can have global or local objectives.
    - Hierarchical clustering algorithms typically have *local objectives*
    - Partitional algorithms typically have *global objectives*
  - A variation is to fit the data to a parameterized model.
    - Parameters for the model are determined from the data.
    - Mixture models assume that the data is a ‘mixture' of a number of statistical distributions.

Types of Clusters: Objective Function…

- Map the clustering problem to a different domain and solve a related problem in that domain
  - Proximity matrix defines a *weighted graph*, where the nodes are the points being clustered, and the weighted edges represent the proximities between points
  - Clustering is equivalent to breaking the graph into connected components, one for each cluster.
  - Want to minimize the edge weight between clusters and maximize the edge weight within clusters

Characteristics of the Input Data

- Type of proximity or density measure
  - This is a derived measure, but central to clustering
- Sparseness
  - Dictates type of similarity
  - Adds to efficiency
- Attribute type
  - Dictates type of similarity
- Type of Data
  - Dictates type of similarity
  - Other characteristics, e.g., autocorrelation
- Dimensionality
- Noise and Outliers
- Type of Distribution

Clustering Algorithms

- K-means and its variants
- Hierarchical clustering
- Density-based clustering
- Graph-based clustering

K-means Clustering

- Partitional clustering approach
- Each cluster is associated with a *centroid* (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified
- The basic algorithm is very simple

1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

K-means Clustering – Details

- Initial centroids are often chosen *randomly*.
  - Clusters produced vary from one run to another.
- The centroid is (typically) the *mean* of the points in the cluster.
- Closeness is measured by Euclidean distance, cosine similarity, correlation, etc.
- K-means converge for above similarity measures
  - Usually converging in the first few iterations.
  - Often the stopping condition is loosen to '*Until relatively few points change clusters*'
- Complexity is O(n * K * I * d)
  - n = number of points, K = number of clusters, I = number of iterations, d = number of attributes

Evaluating K-means Clusters

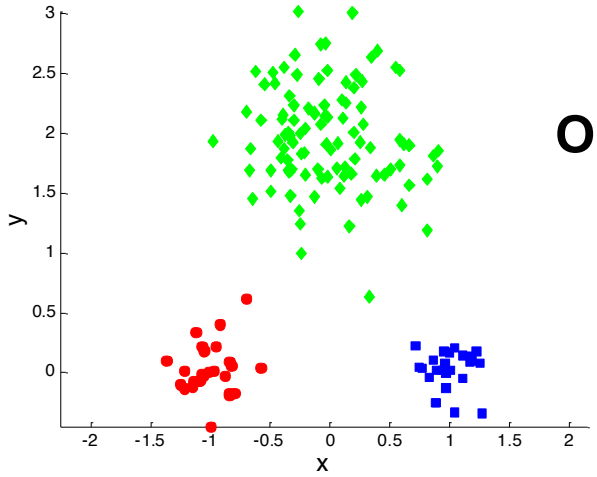
- Common measure: *Sum of Squared Error (SSE)*
  - Error of a point is its distance to the nearest cluster
  - To get SSE, we square these errors and sum them.

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

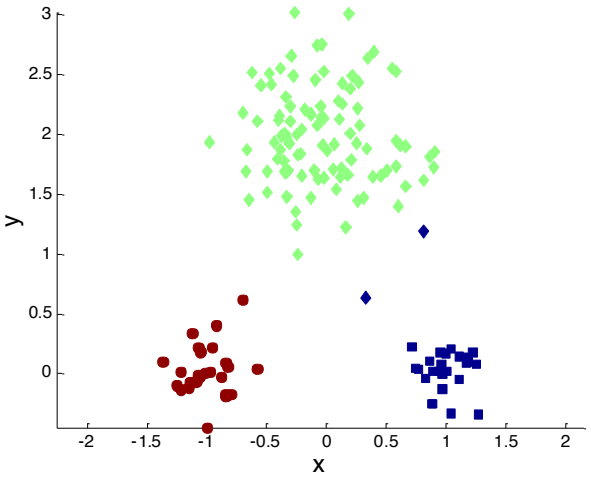
where x is a data point in cluster C_i and m_i is the centroid for cluster C_i

    - can show that m_i corresponds to the center (mean) of C_i
  - Among clusterings, choose the one with smallest SSE
    - One way is to reduce SSE is to increase K, the # of clusters
    - Don't use a very large K -- SSE=0 when each point is a cluster
  - *A good clustering with smaller K could have a lower SSE than a poor clustering with higher K*

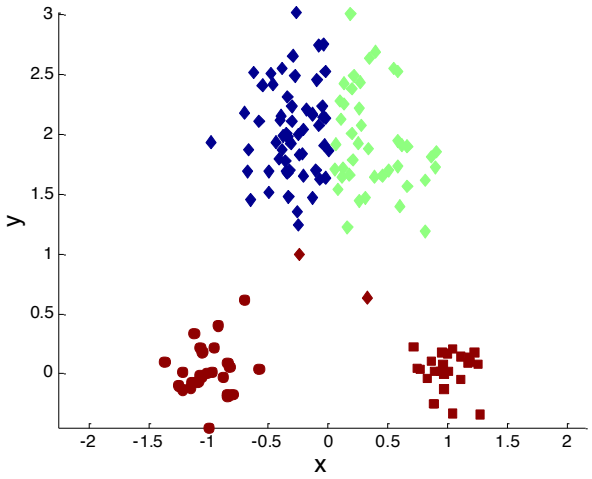
Two different K-means Clusterings



Original Points

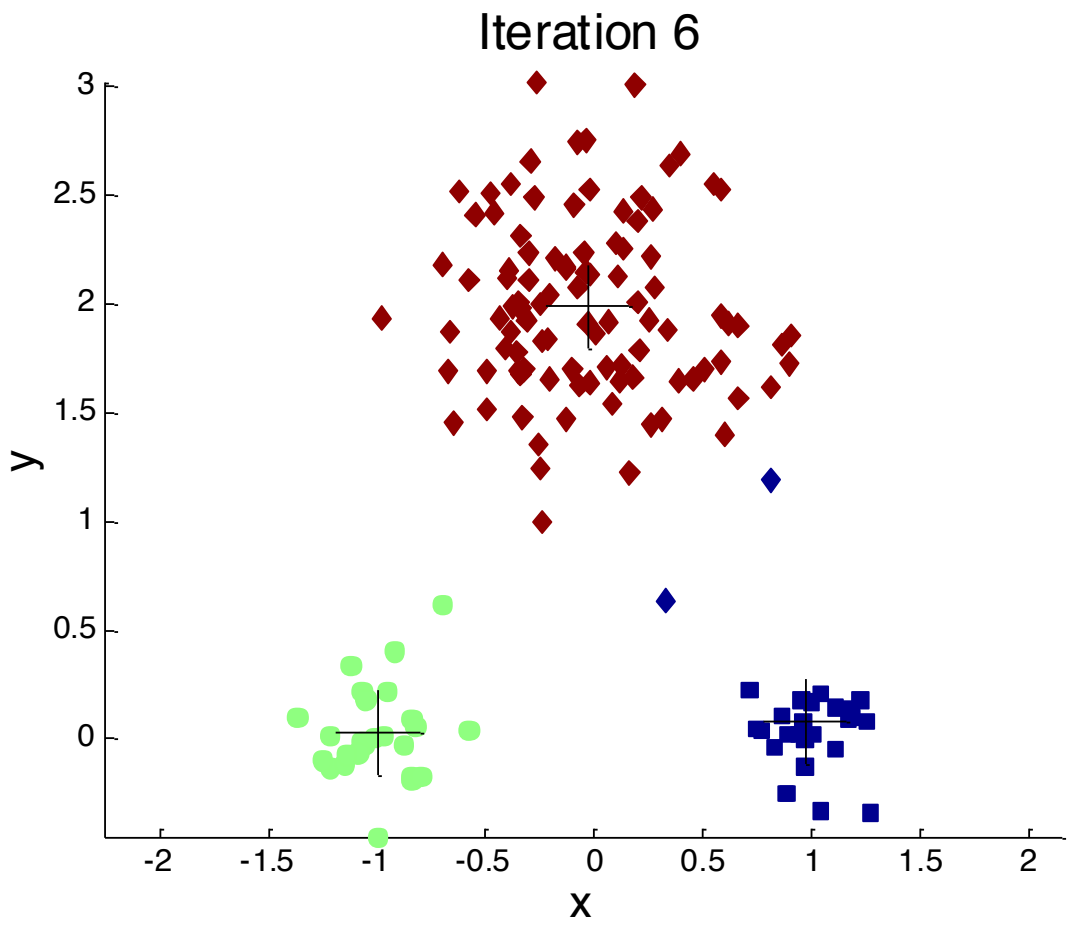


Optimal Clustering

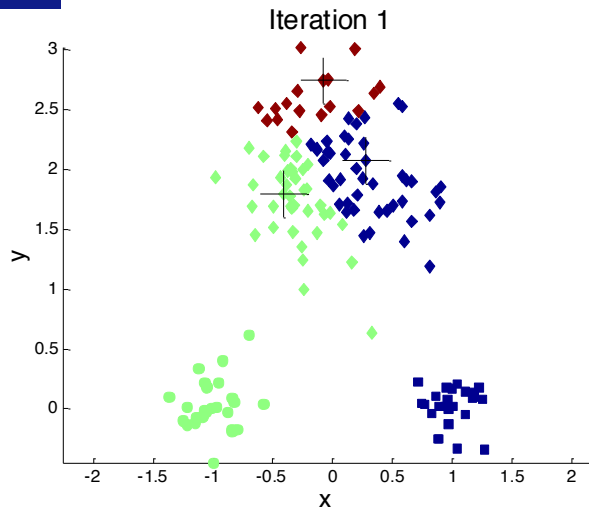


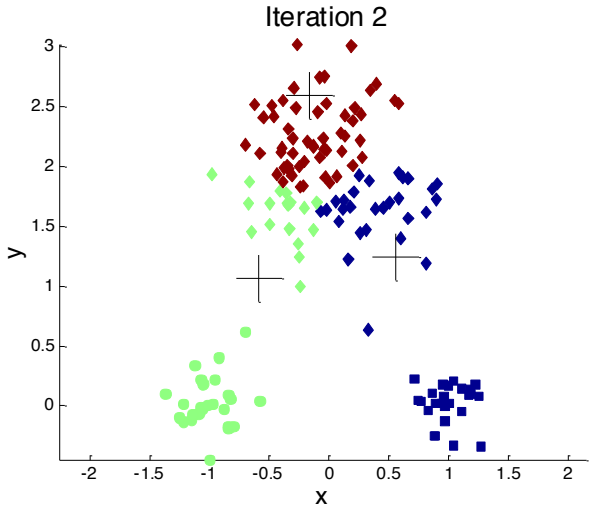
Sub-optimal Clustering

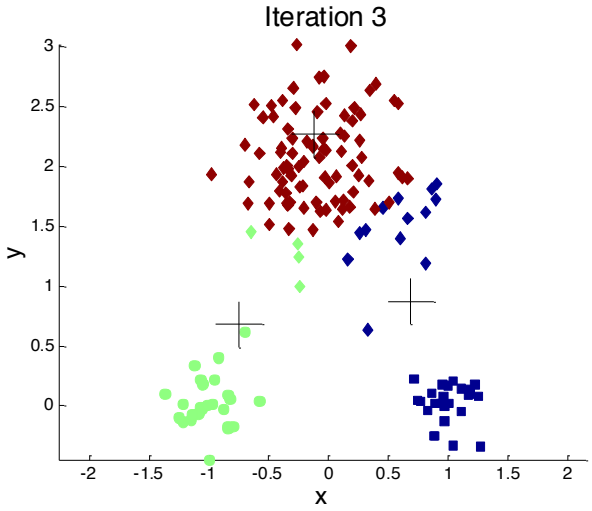
Importance of Initial Centroids

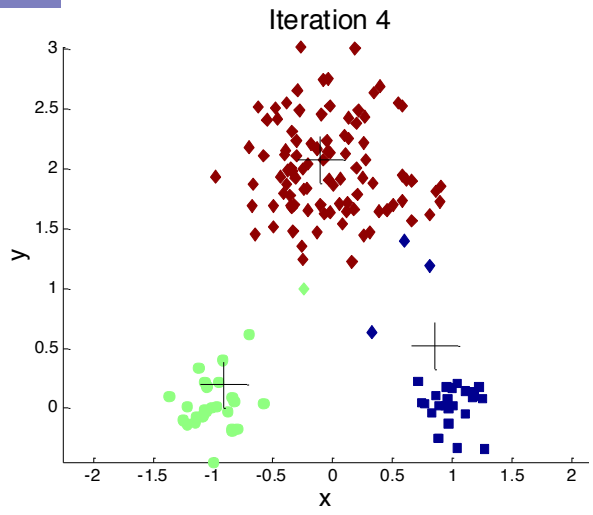


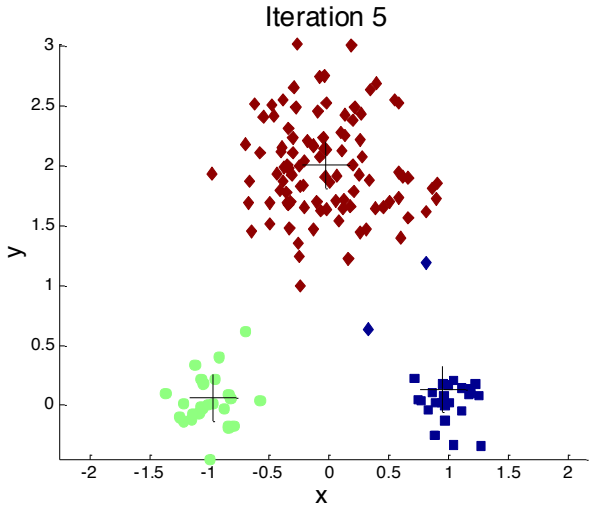
Importance of Initial Centroids

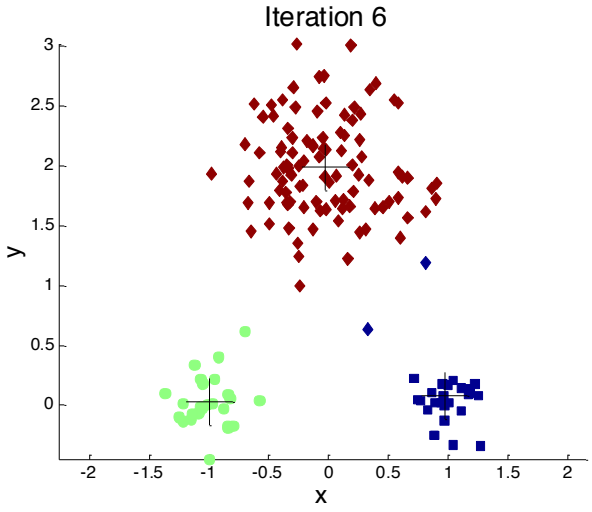




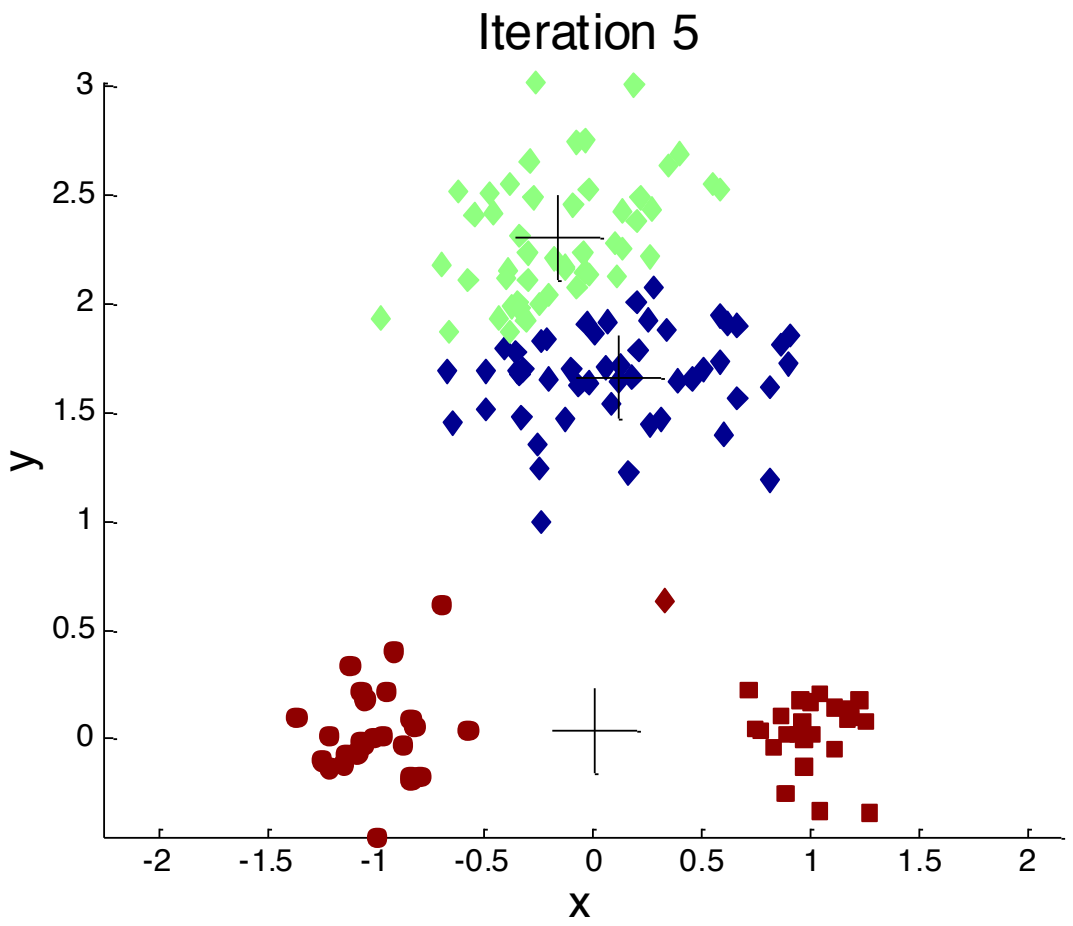




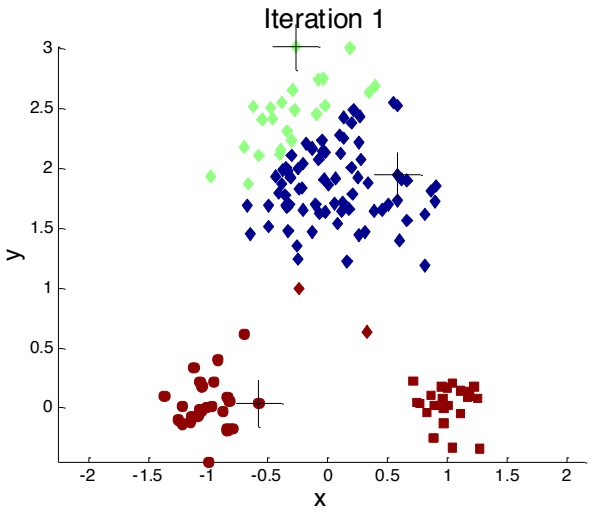


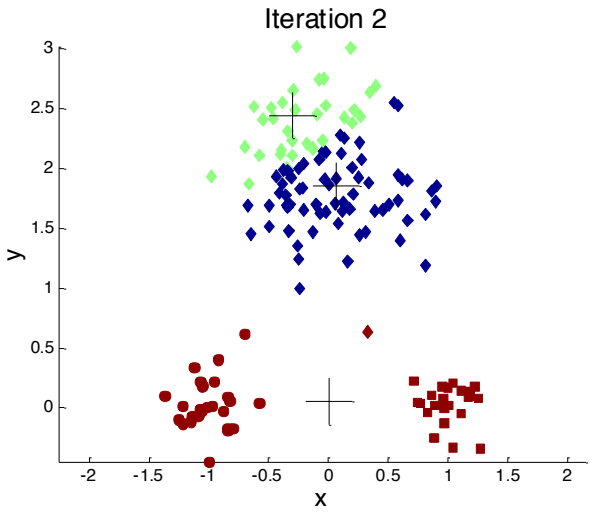


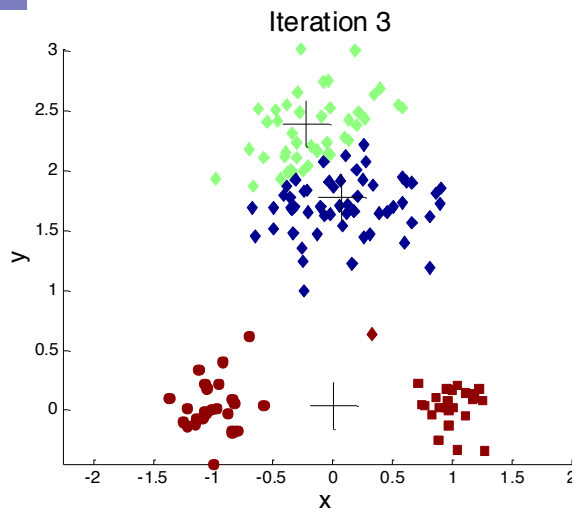
Different Initial Centroids

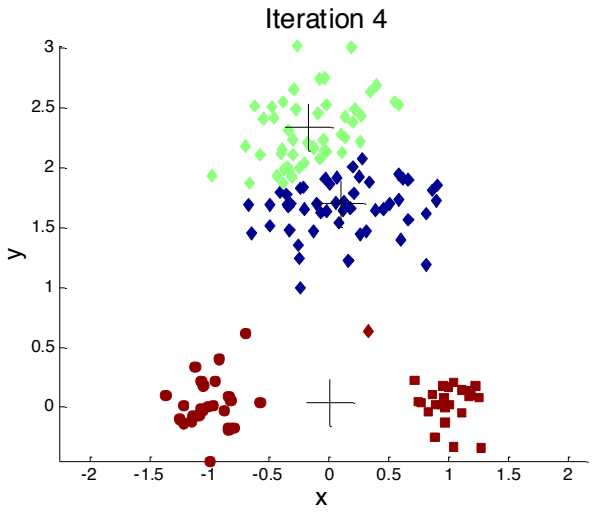


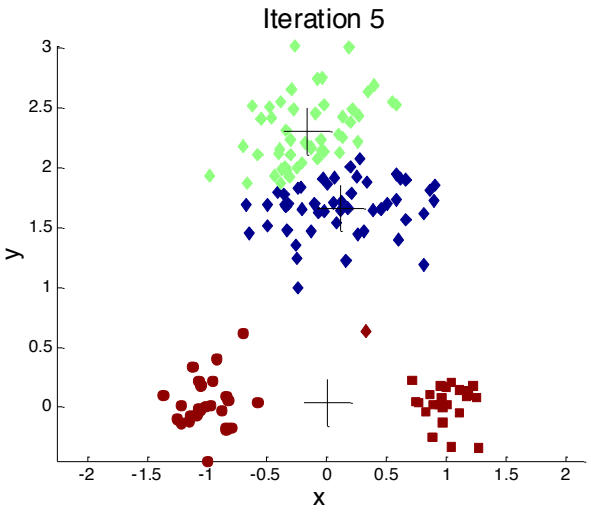
Different Initial Centroids











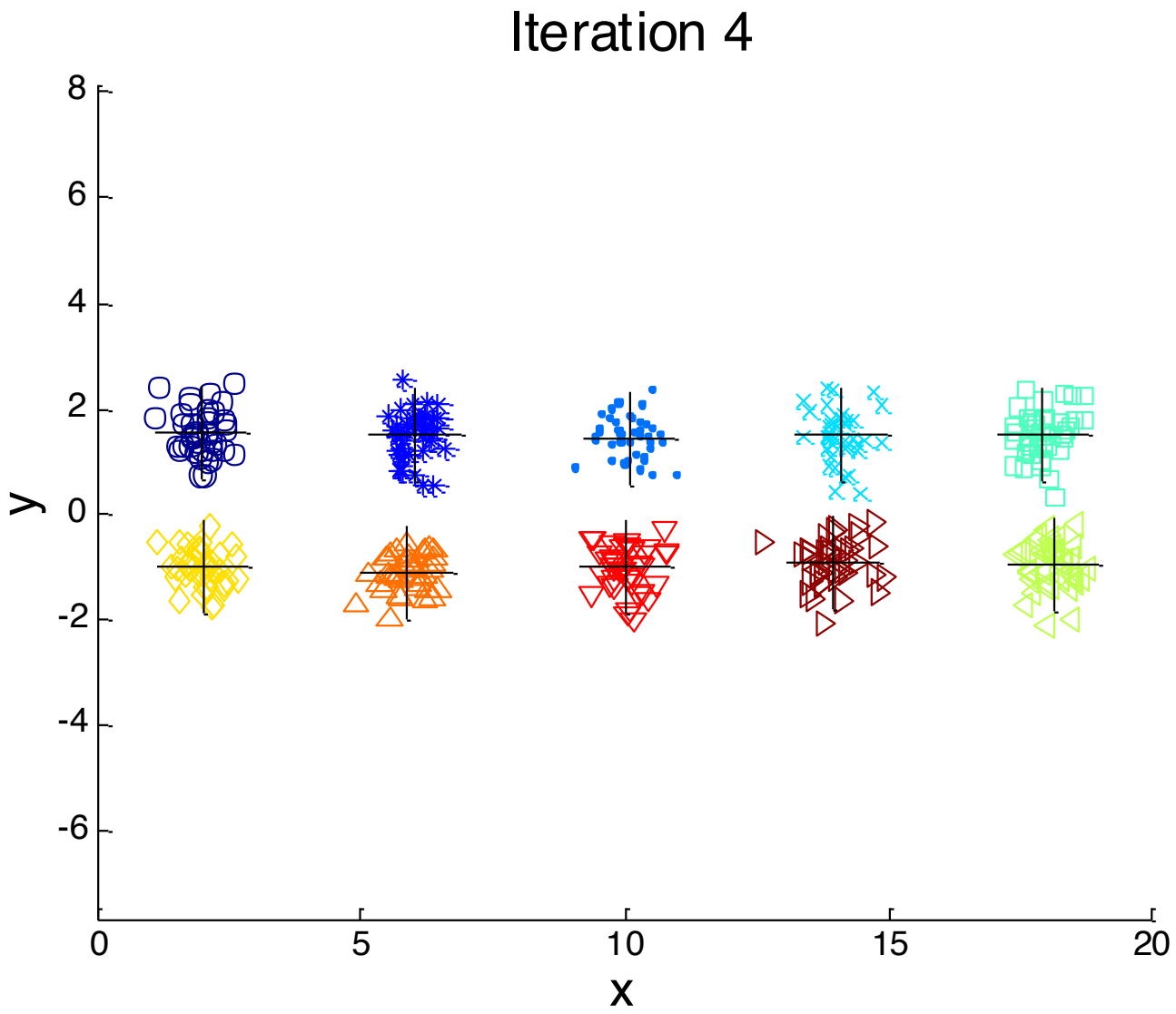
Problems with Selecting Initial Points

- If there are K 'real' clusters then the chance of selecting one centroid from each cluster is small.
  - Chance is relatively small when K is large
  - If clusters are the same size, n, then

$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K}$$

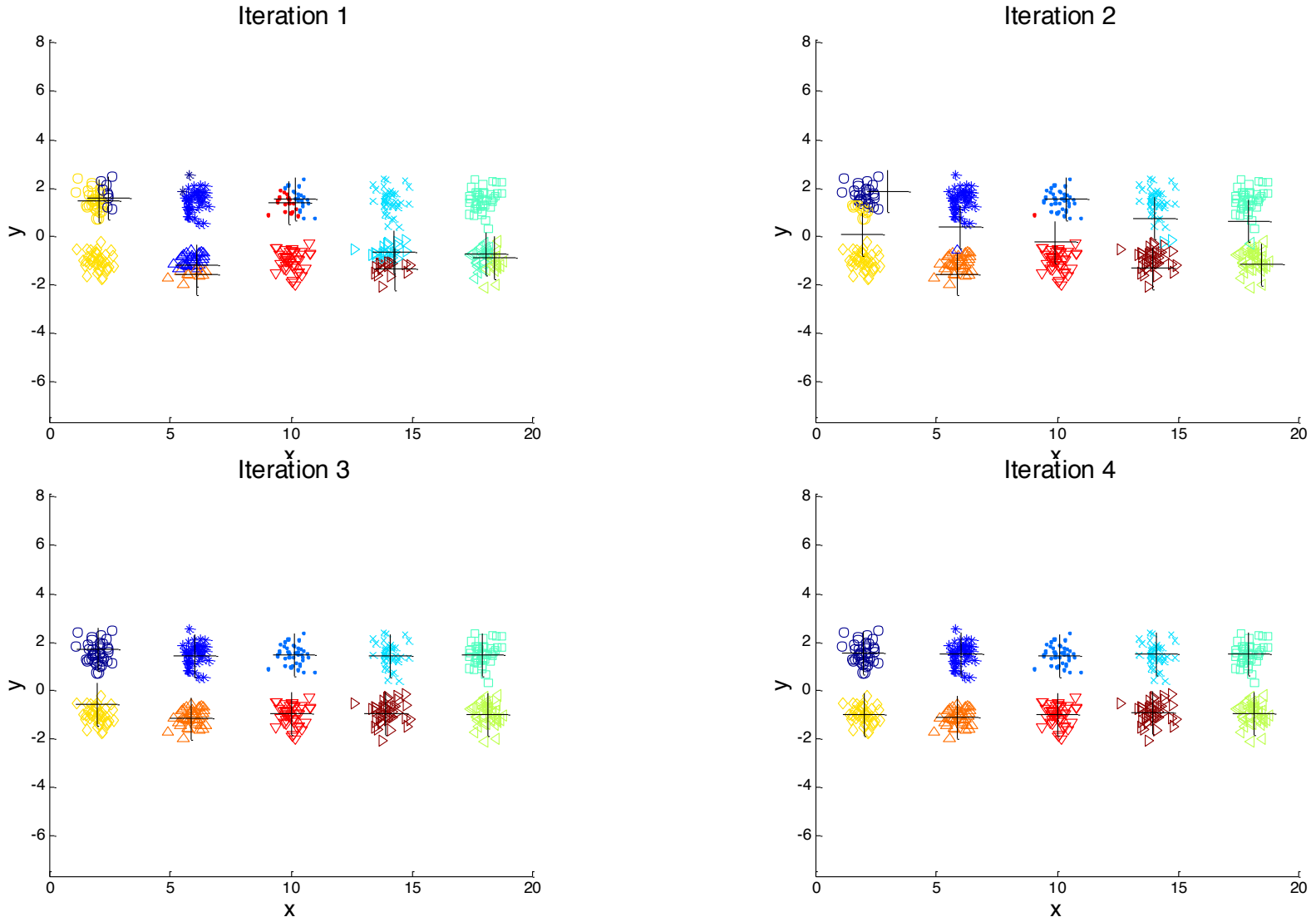
  - For example, if K = 10, then probability = $10!/10^{10} = 0.00036$
  - Sometimes the initial centroids will read just themselves in 'right' way, and sometimes they don't
  - Consider an example of five pairs of clusters

10 Cluster Example 1



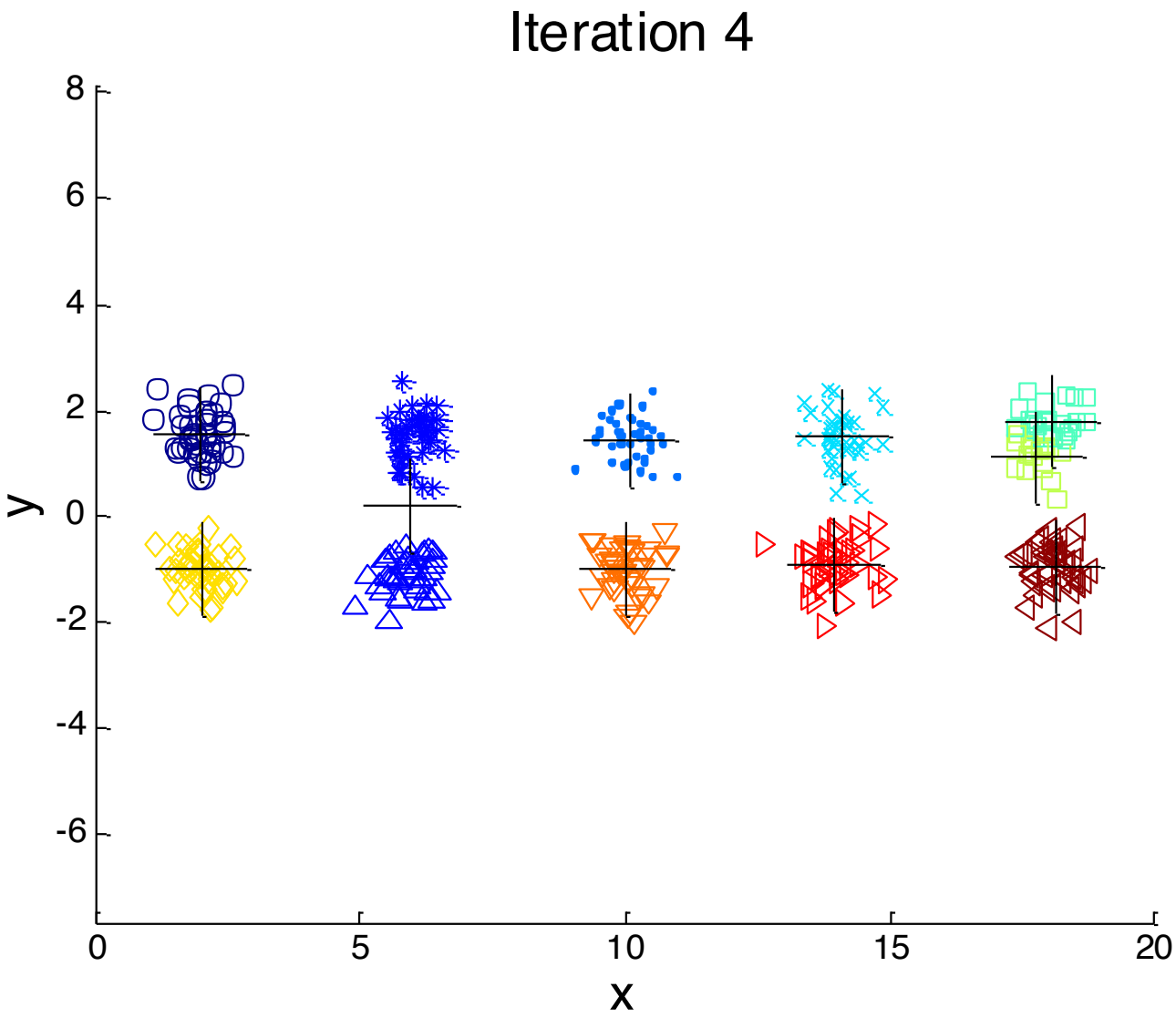
Starting with two initial centroids in one cluster of each pair of clusters

10 Cluster Example 1



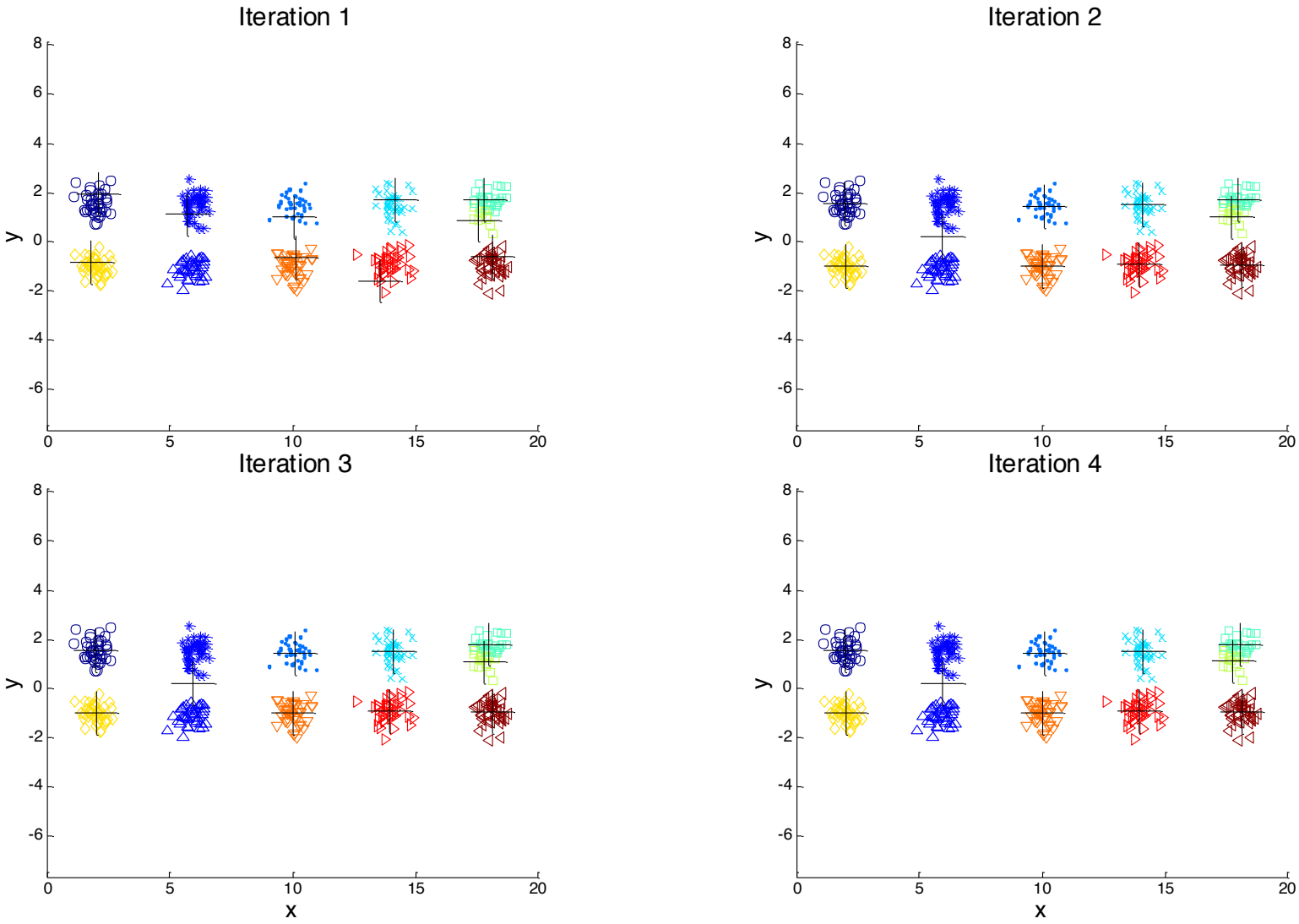
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example 2



Starting with some pairs of clusters having three initial centroids, while other have only one.

10 Clusters Example 2



Starting with some pairs of clusters having three initial centroids, while other have only one.

Solutions to Initial Centroids Problem

- Multiple runs (with random initial centroids)
  - Helps, but probability is not on your side
- Sample and use hierarchical clustering to determine initial centroids
- Sample (i.e., select more than k) initial centroids and then select among these initial centroids
  - Select most widely separated
- Postprocessing
- Bisecting K-means
  - Not as susceptible to initialization issues

Handling Empty Clusters

- Basic K-means algorithm can yield empty clusters
  - No point is assigned to a cluster (i.e., all points in the dataset is far away from its centroid)
- Strategies to find a new centroid
  - Choose the point that contributes most to SSE Choose a point farthest to centroid of the empty cluster, i.e., reduce its error to 0
  - Choose a point from the cluster with the highest SSE
  - If there are several empty clusters, the above can be repeated several times.

Updating Centroids Incrementally

- In the basic K-means algorithm, centroids are updated after all points are assigned to a centroid

- Incremental approach: *update the centroids after each assignment*
  - Each assignment updates zero or one centroids
  - More expensive
  - Introduces an order dependency
  - Never get an empty cluster

Reducing the SSE

- K-Means does not always produce minimal SSE. Strategies have been proposed to improve it.
- Pre-processing
  - Normalize the data
  - Eliminate outliers
- Post-processing
  - Eliminate small clusters that may represent outliers
  - Split ‘loose’ clusters with relatively high SSE
  - Merge ‘close’ clusters that have relatively low SSE
  - Can use these steps during the clustering process
    - ISODATA Clustering Algorithm

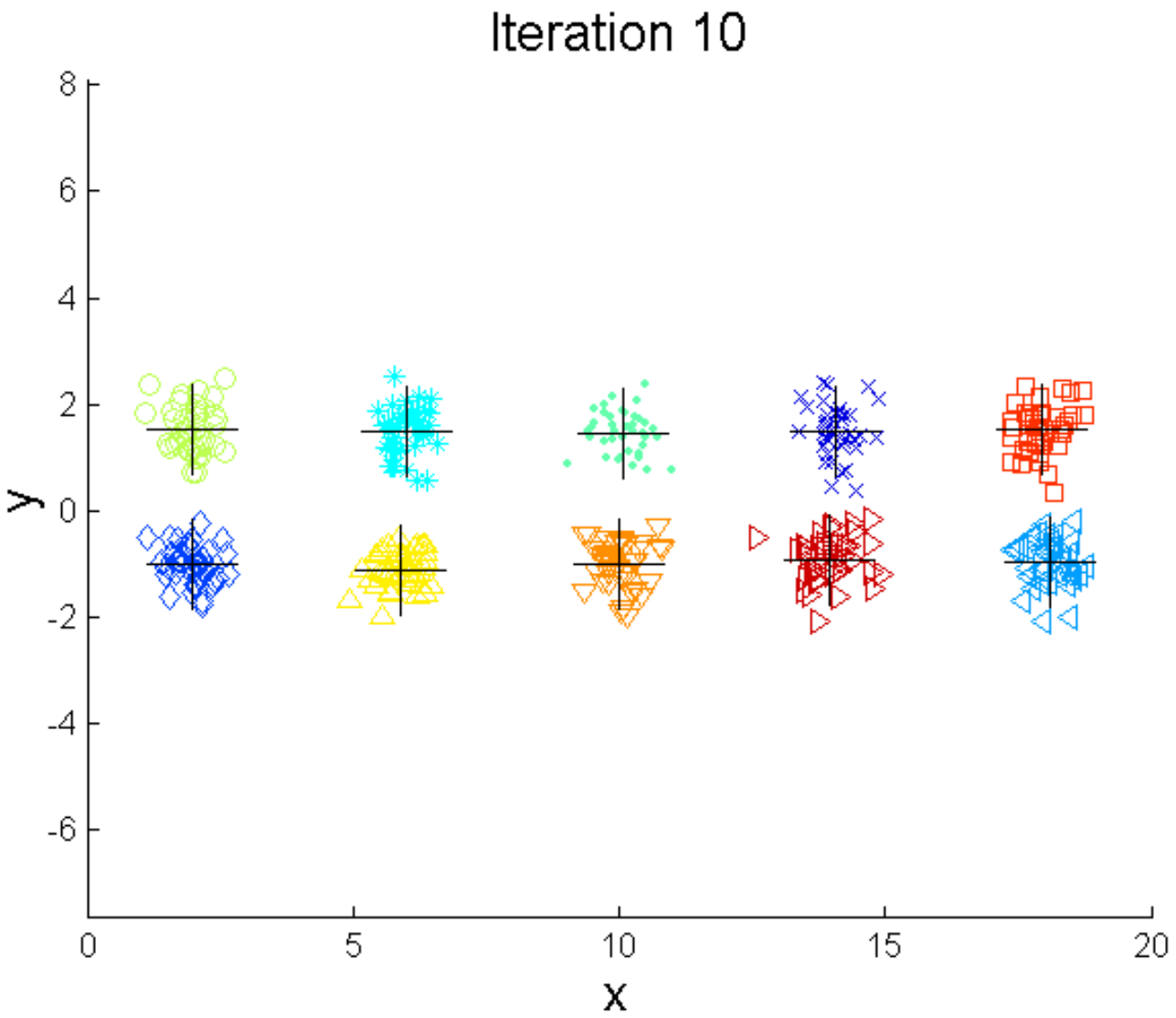
Bisecting K-means

- Bisecting K-means algorithm
  - Variant of K-means that can produce a partitional or hierarchical clustering
  - Using k-Means locally to bisecting individual clusters!

1: Initialize the list of clusters to contain the cluster containing all points.
2: **repeat**
3: Select a cluster from the list of clusters
4: **for** $i = 1$ to *number_of_iterations* **do**
5: Bisect the selected cluster using basic K-means
6: **end for**
7: Add the two clusters from the bisection with the lowest SSE to the list of clusters.
8: **until** Until the list of clusters contains K clusters

  - By recording the sequence of bisecting, we can use bisecting K-Means to produce a hierarchical clustering.

Bisecting K-means Example

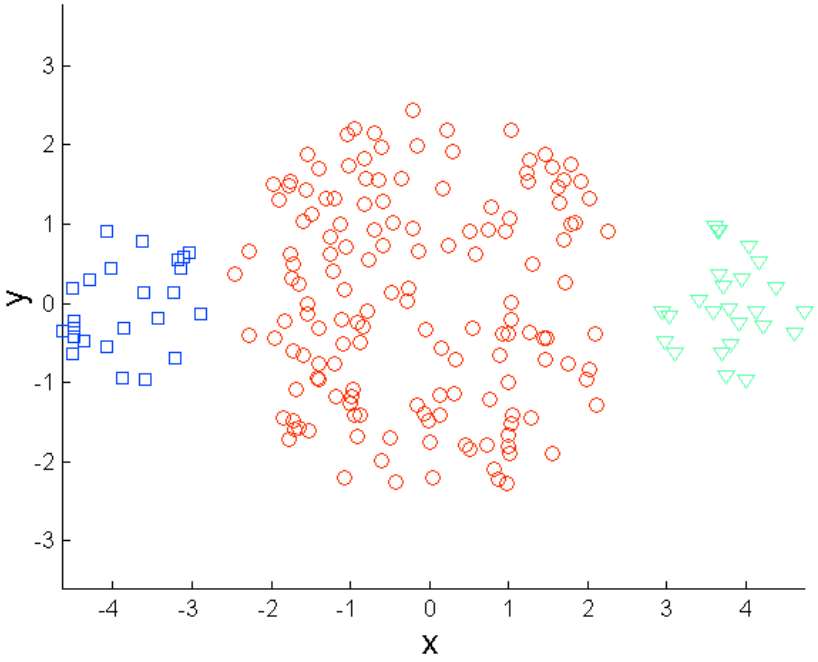


Limitations of K-means

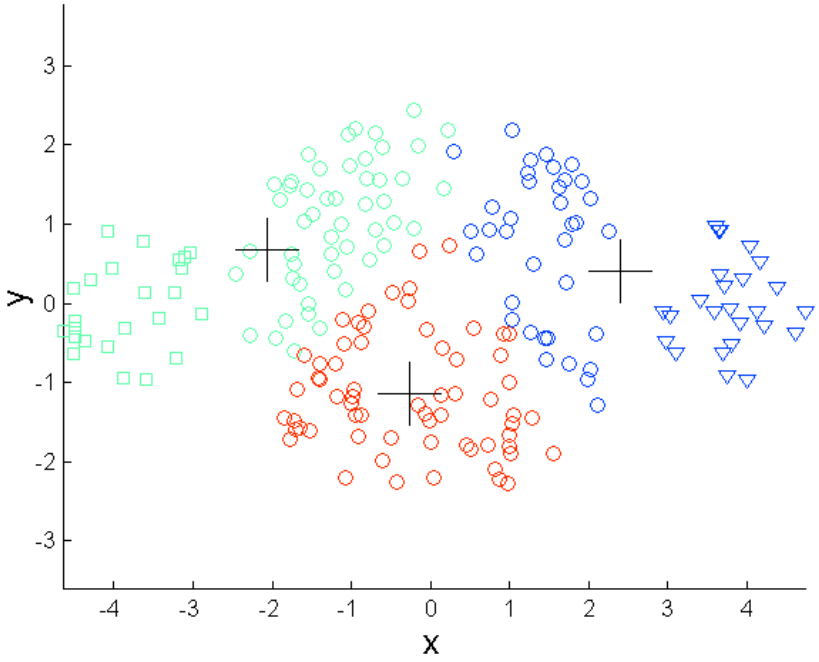
- K-means has problems when clusters are of different
  - Sizes
  - Densities
  - Non-globular shapes

- K-means has problems when the data contains outliers.

Limitations of K-means: Different Sizes

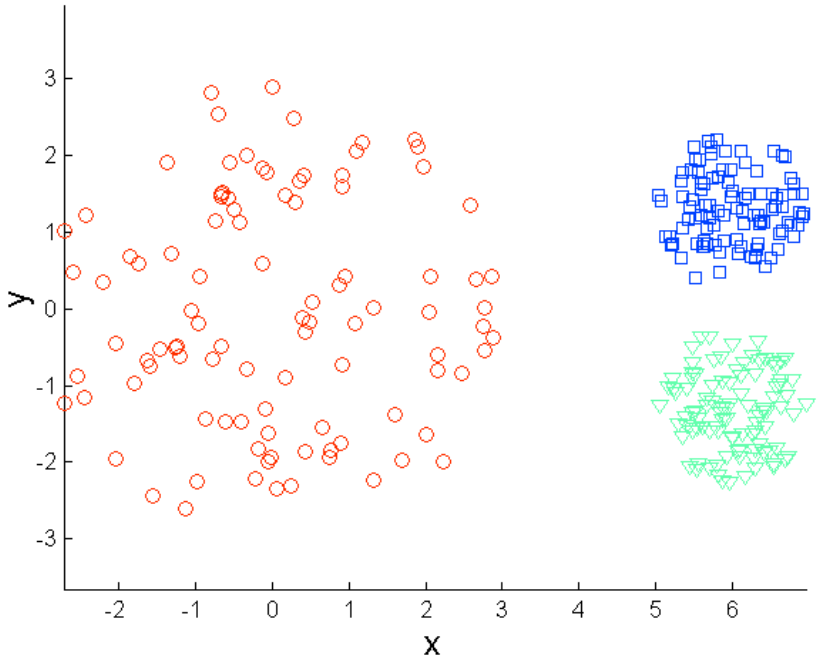


Original Points

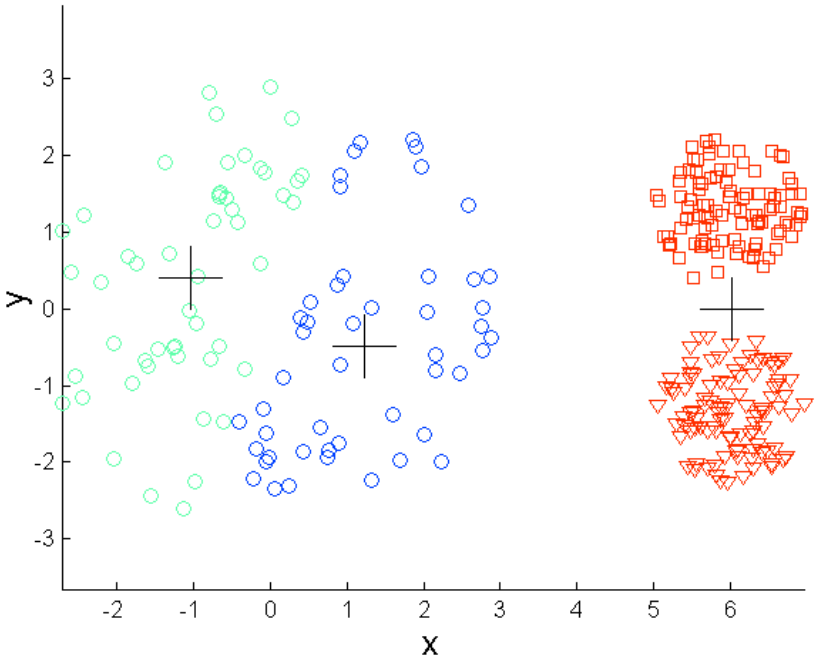


K-means (3 Clusters)

Limitations of K-means: Different Density

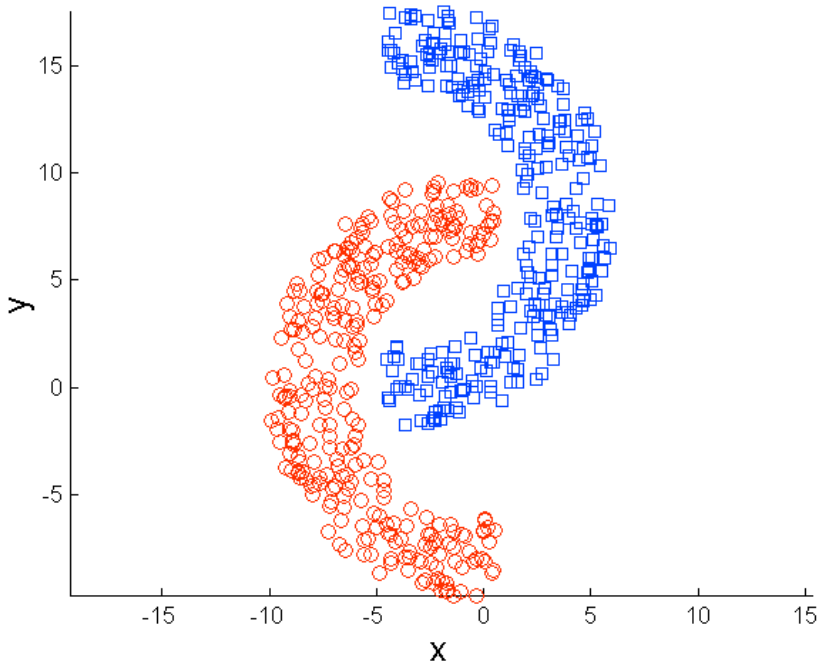


Original Points

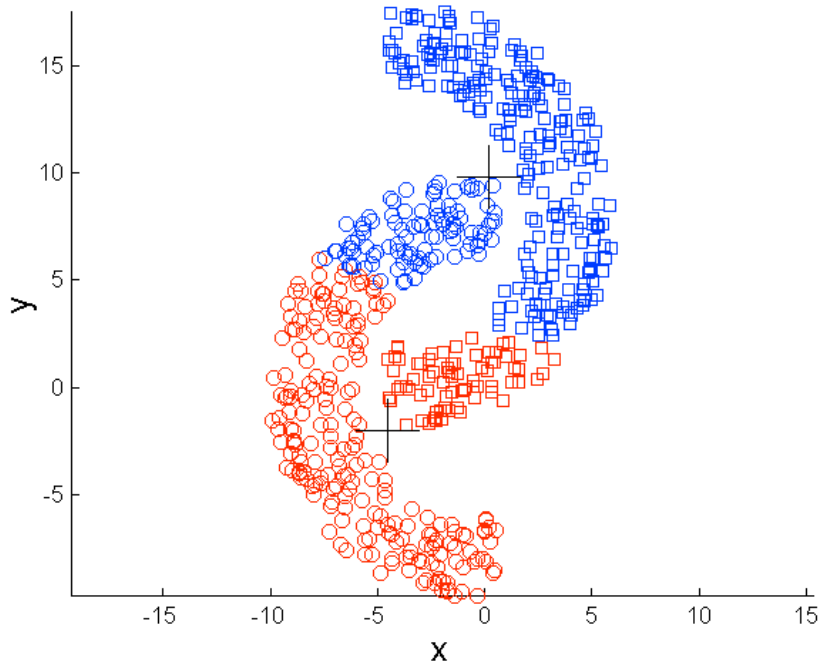


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes

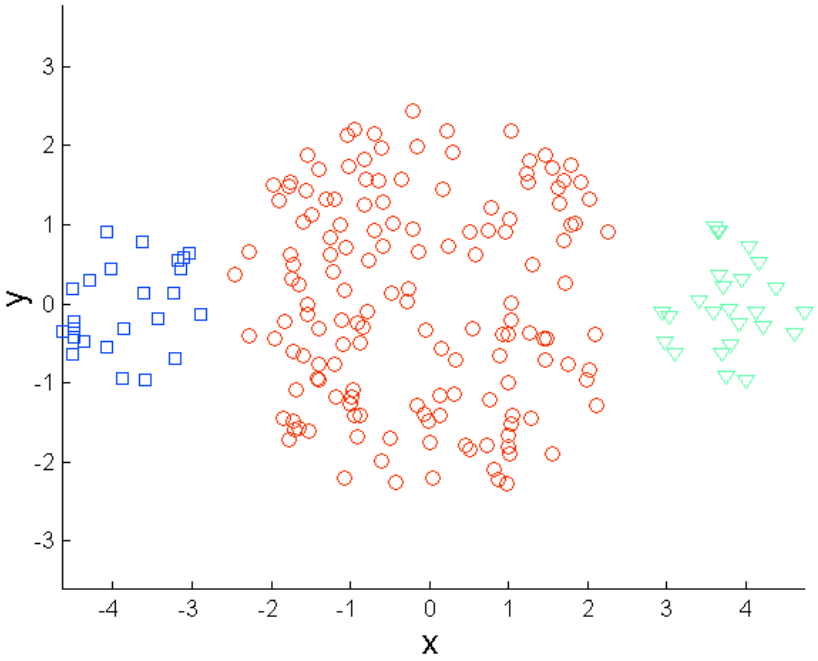


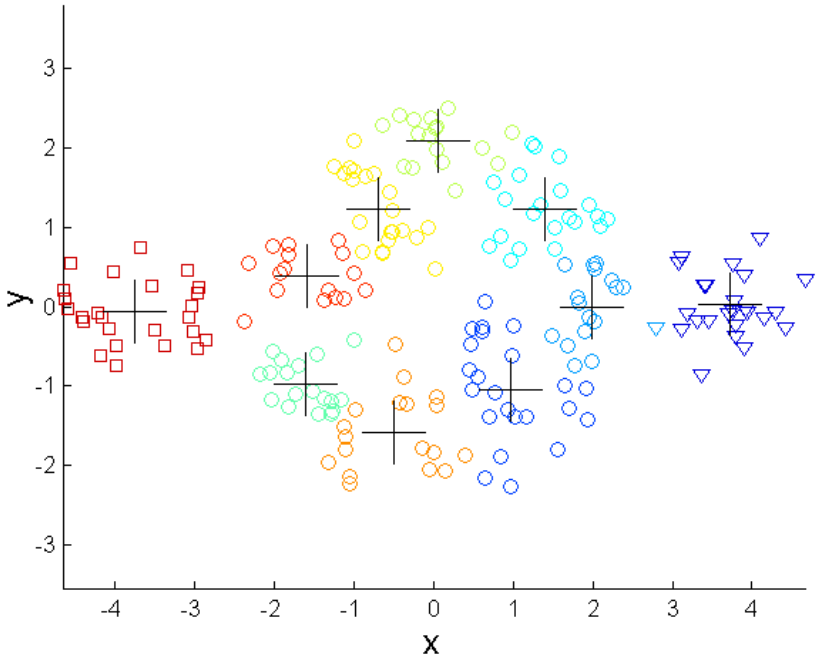
Original Points



K-means (2 Clusters)

Overcoming K-means Limitations

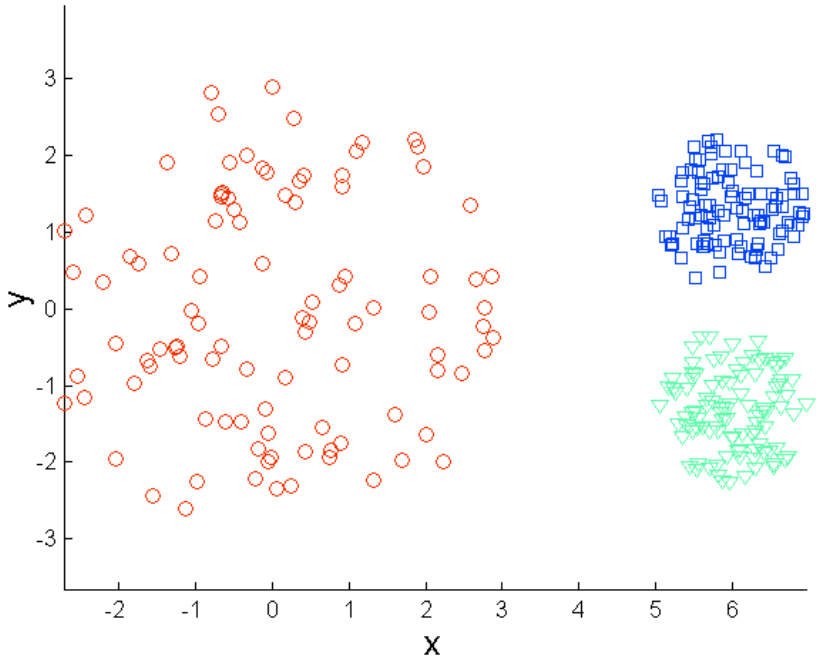




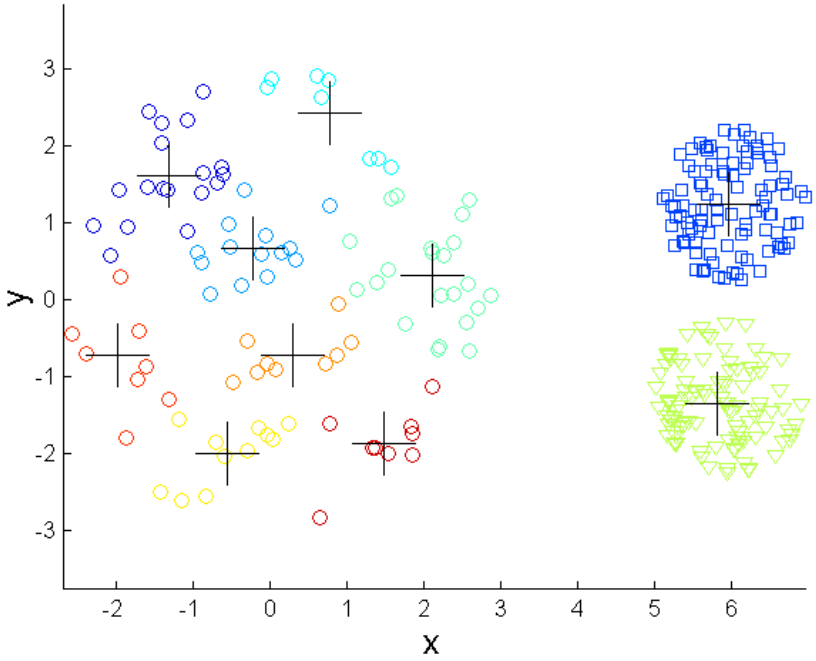
Original Points **K-means Clusters**

- One solution is to use many clusters.
  - Find parts of clusters, but need to put together.

Overcoming K-means Limitations

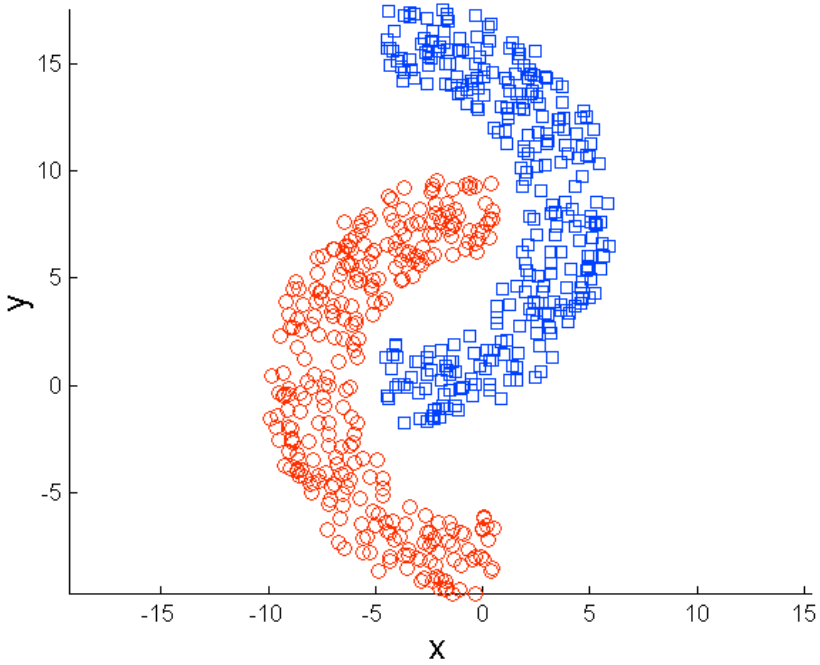


Original Points

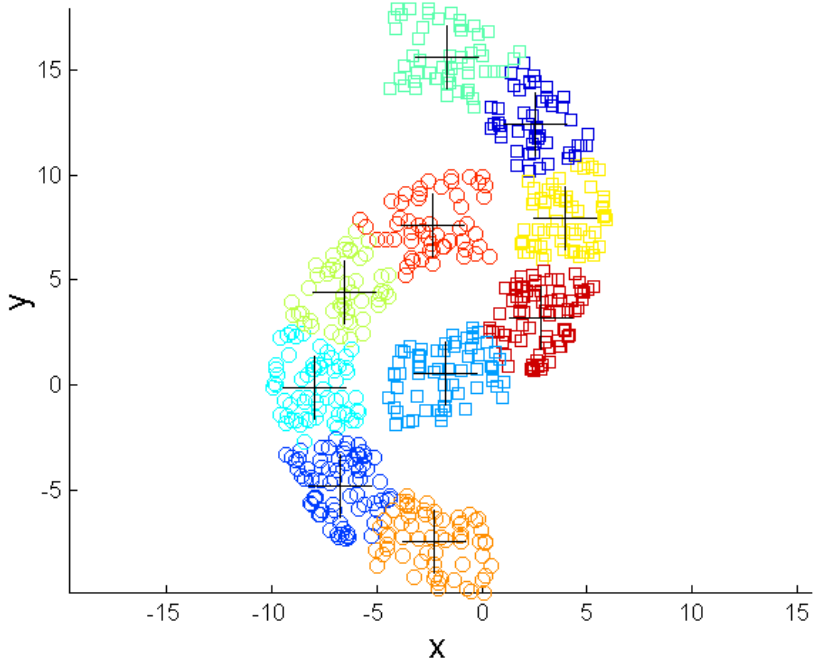


K-means Clusters

Overcoming K-means Limitations



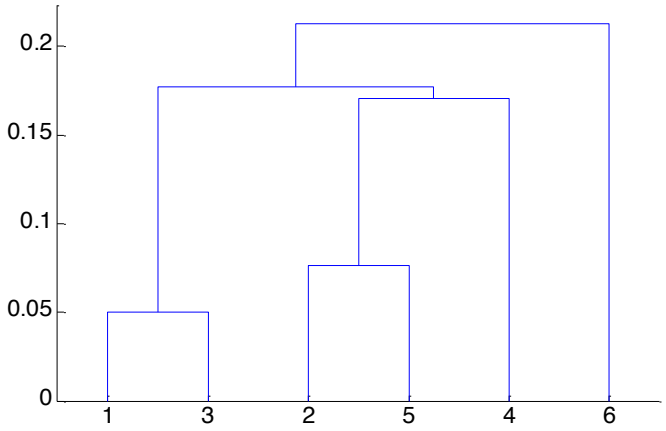
Original Points

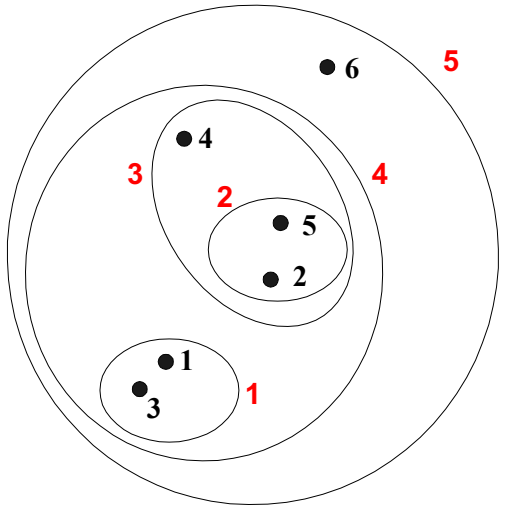


K-means Clusters

Hierarchical Clustering

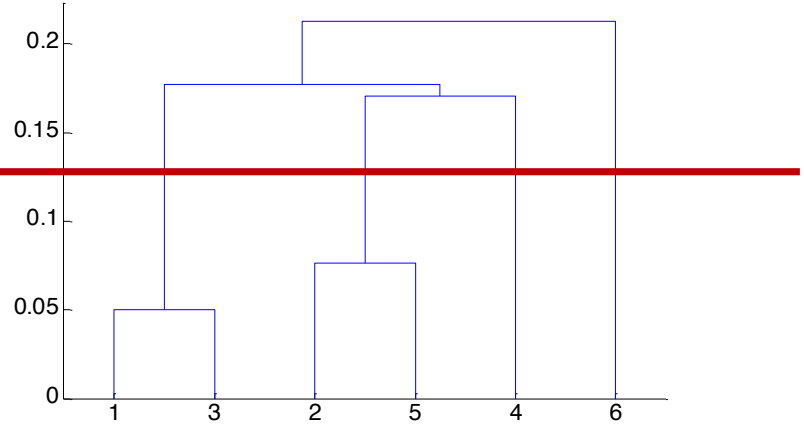
- Produces a set of nested clusters organized as a *hierarchical tree*
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits





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

- *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 (proximity) matrix
  - Merge or split one cluster at a time

Divisive Clustering Algorithms

- Bisecting K-Means is an example of divisive clustering
- Minimum Spanning Tree (MST) Clustering
  - Compute a MST based on a dissimilarity graph
  - Repeat
  - Create a new cluster by breaking the link with the largest dissimilarity
  - Until only singleton clusters left

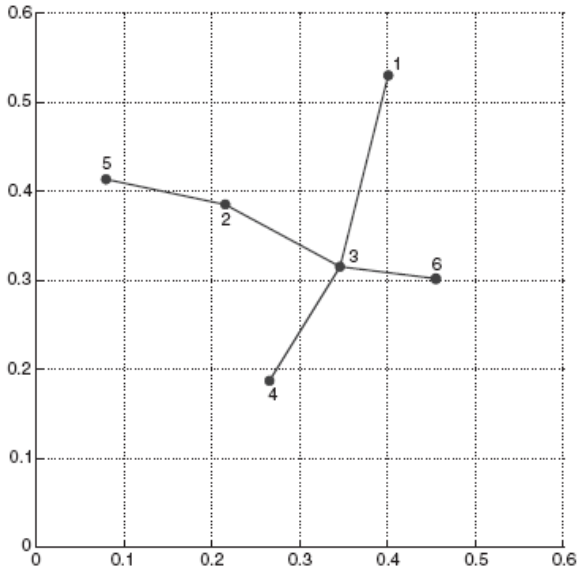


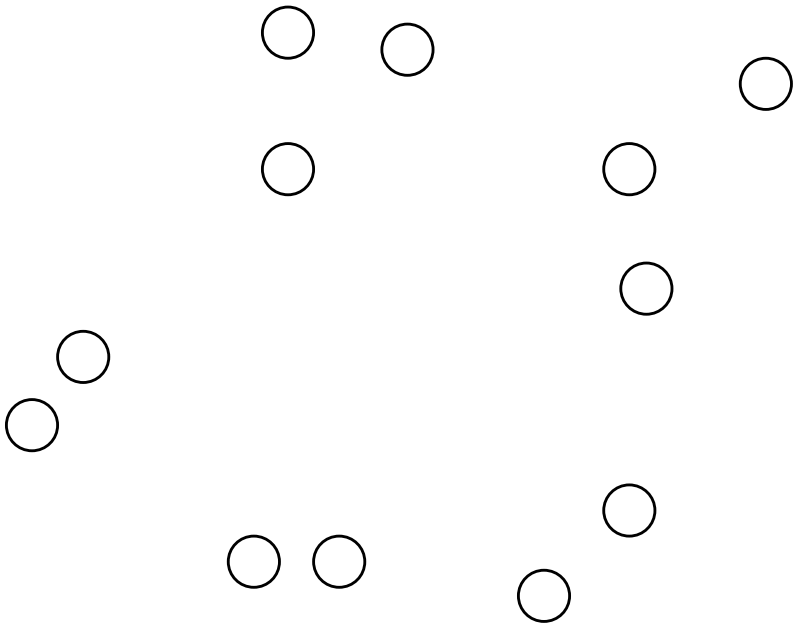
Figure 9.16. Minimum spanning tree for a set of six two-dimensional points.

Agglomerative Clustering Algorithm

- 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 define the distance between clusters distinguish the different algorithms

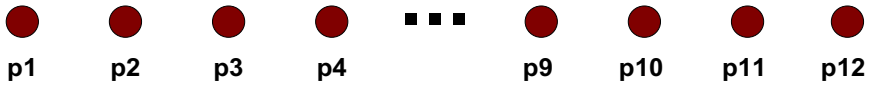
Starting Situation

- Start with clusters of individual points and a proximity matrix



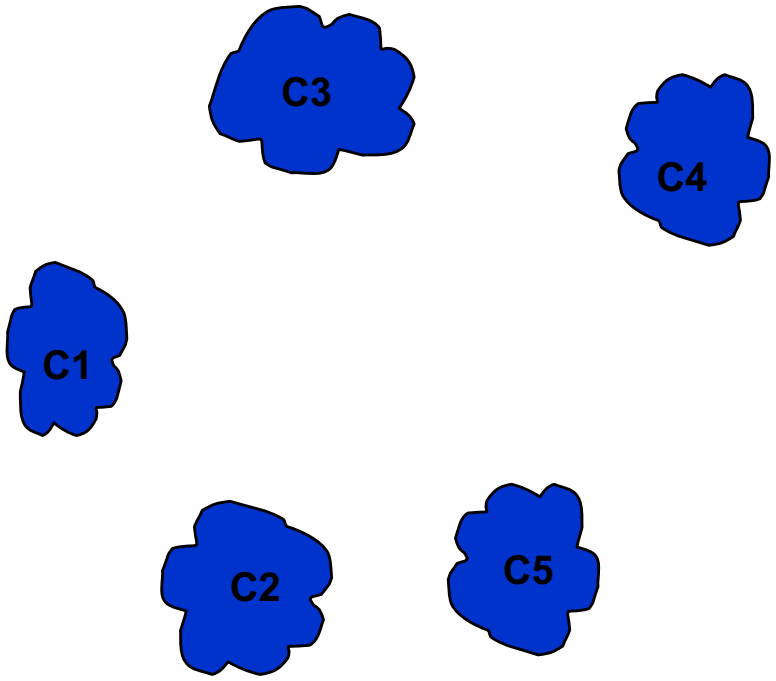
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix



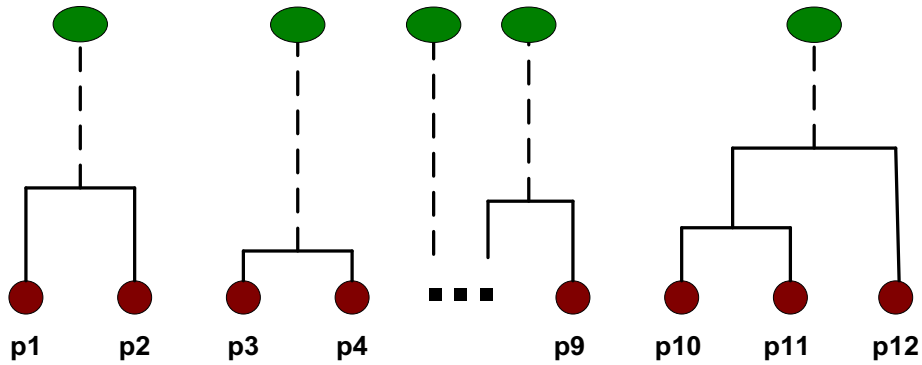
Intermediate Situation

- After some merging steps, we have some clusters



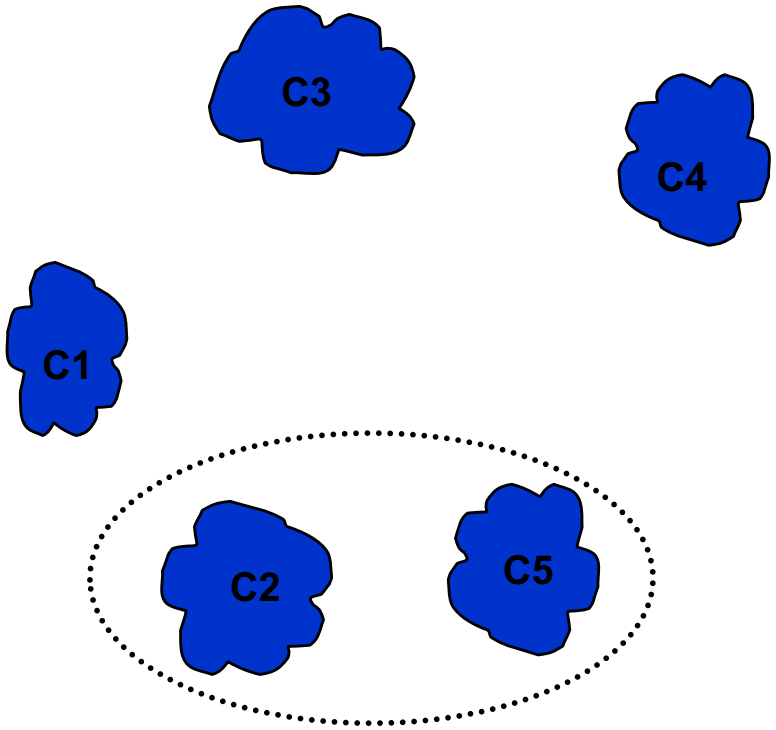
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

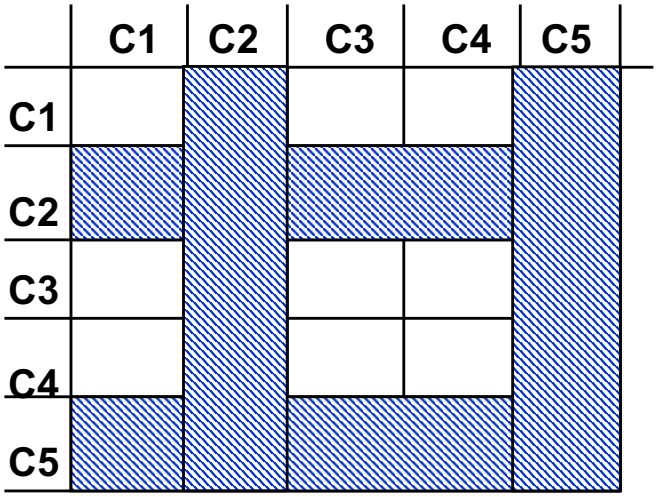
Proximity Matrix



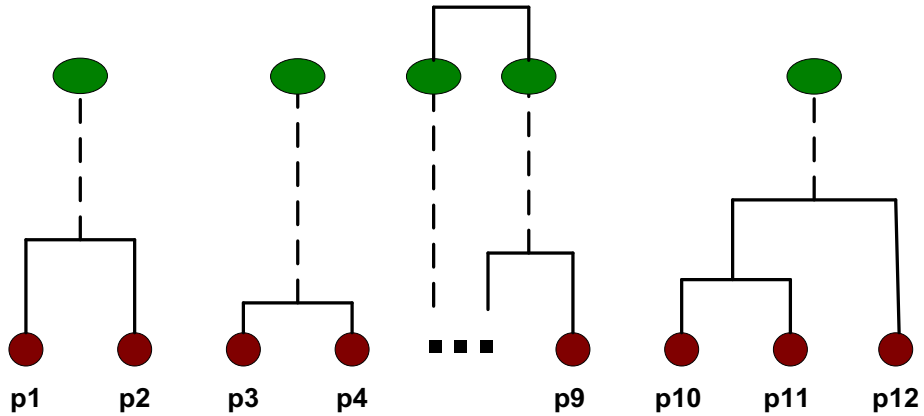
Intermediate Situation

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



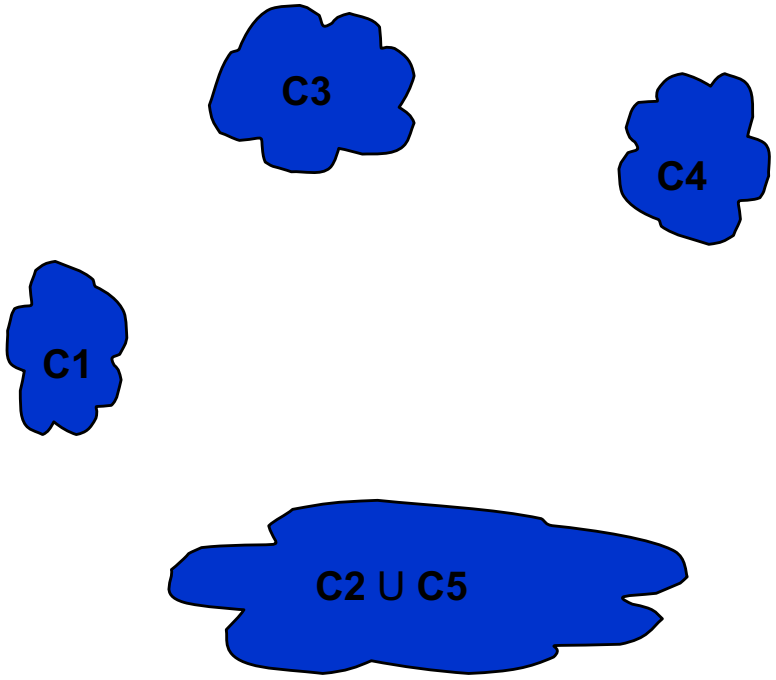


Proximity Matrix



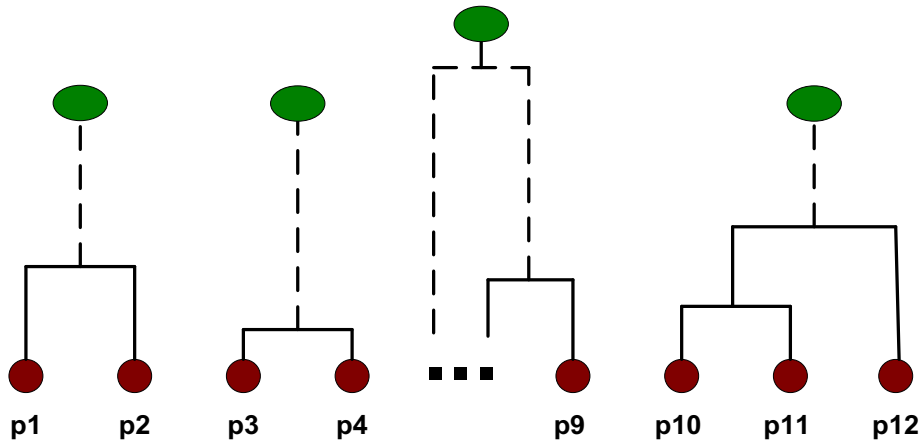
After Merging

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

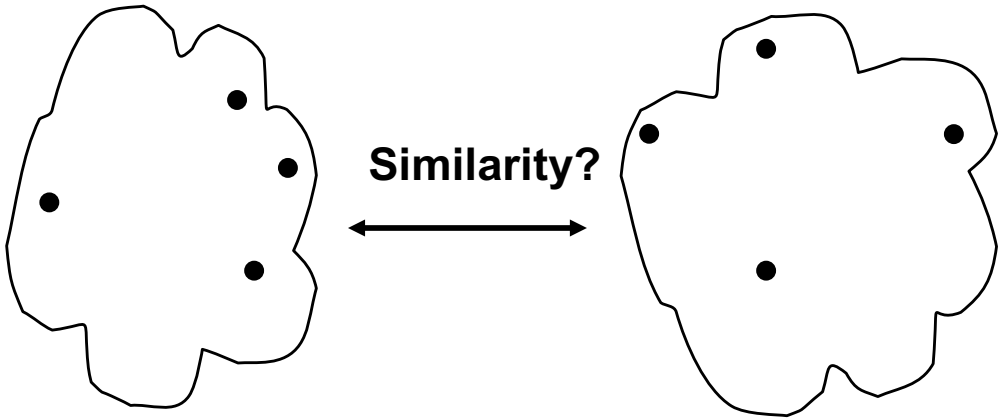


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



How to Define Inter-Cluster Similarity

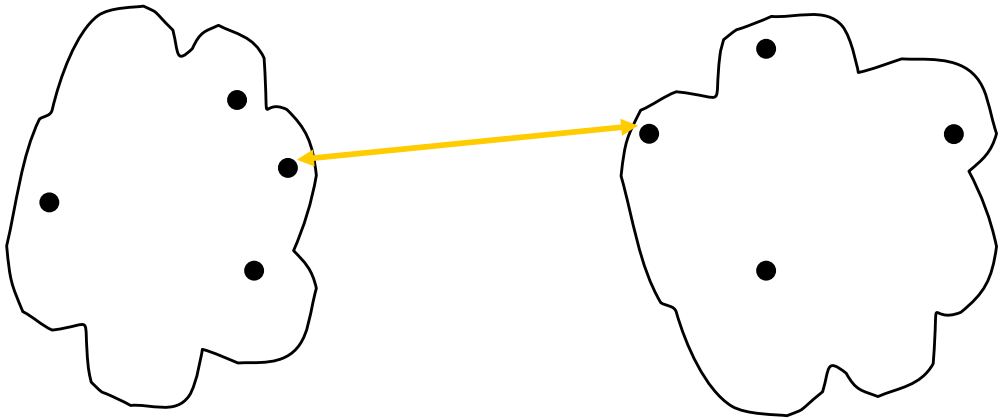


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

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

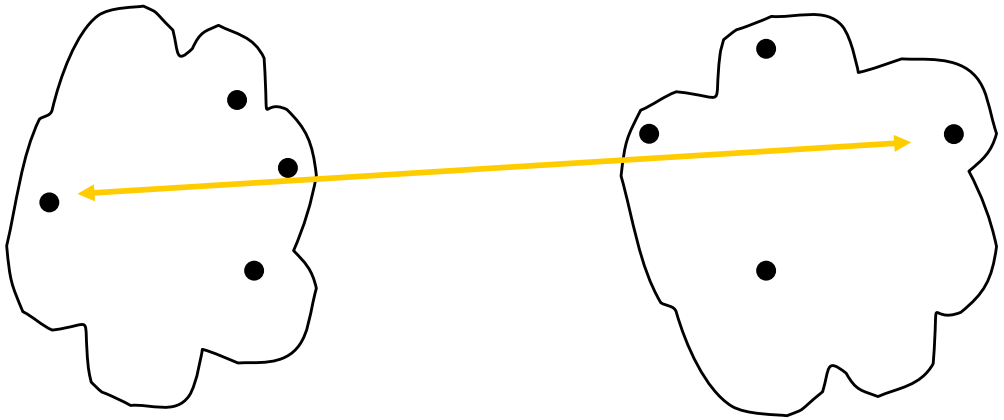


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

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

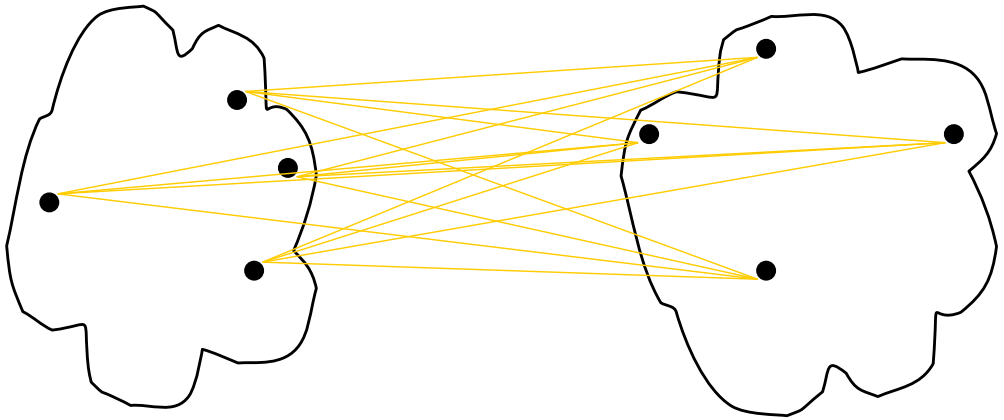


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

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

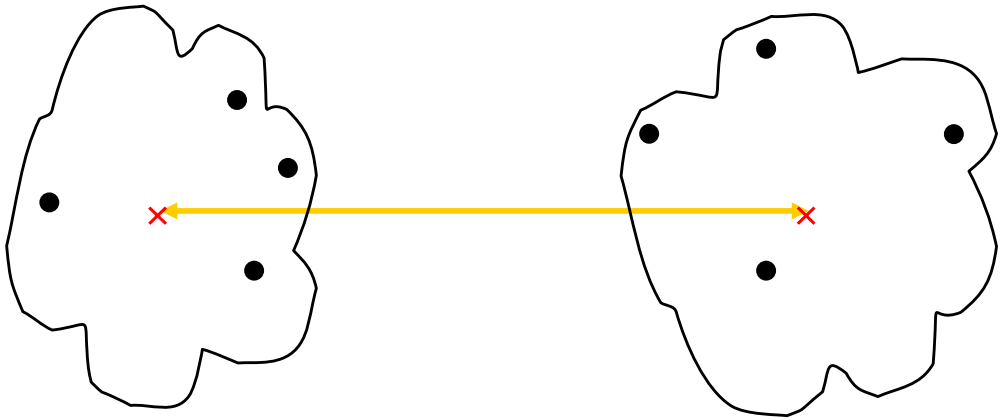


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

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						
.						
.						
.						

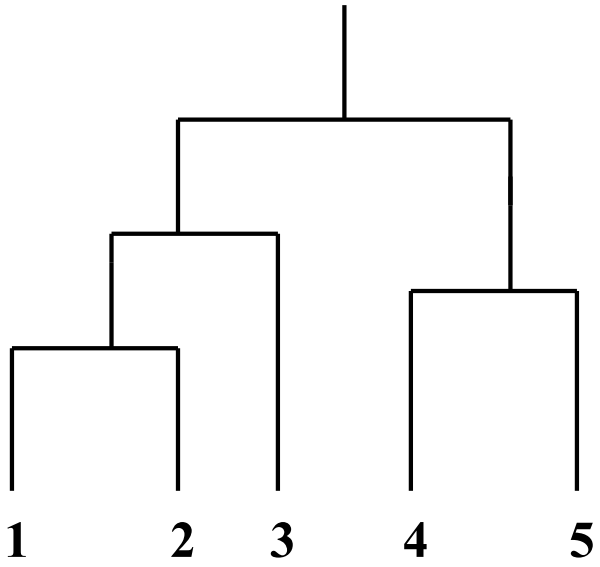
Proximity Matrix

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

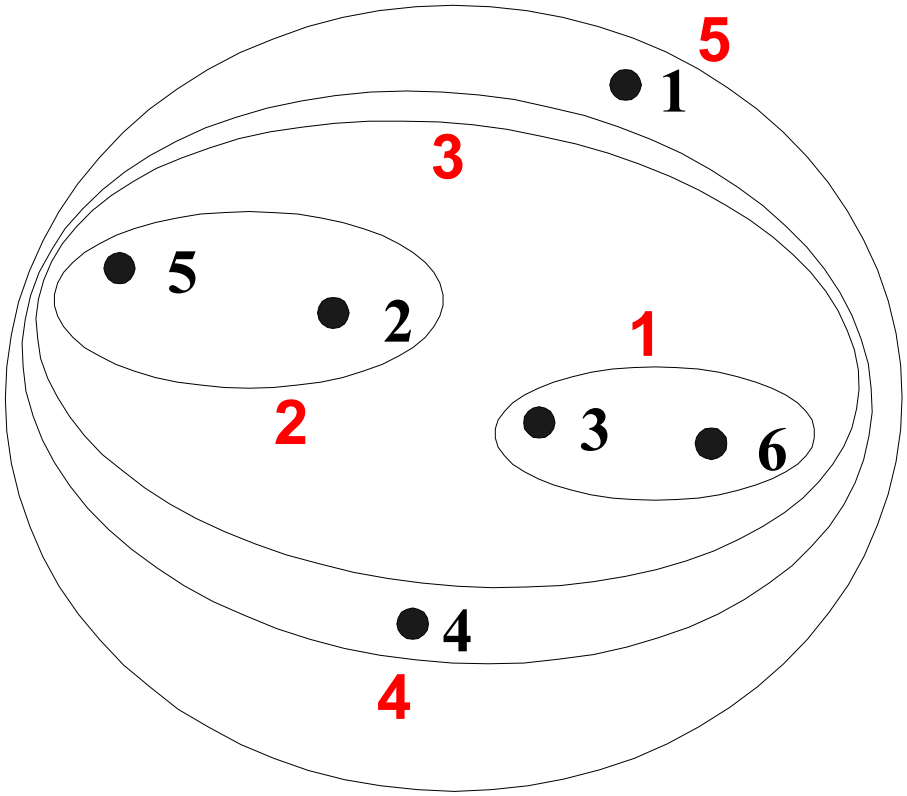
Cluster Similarity: MIN or Single Link

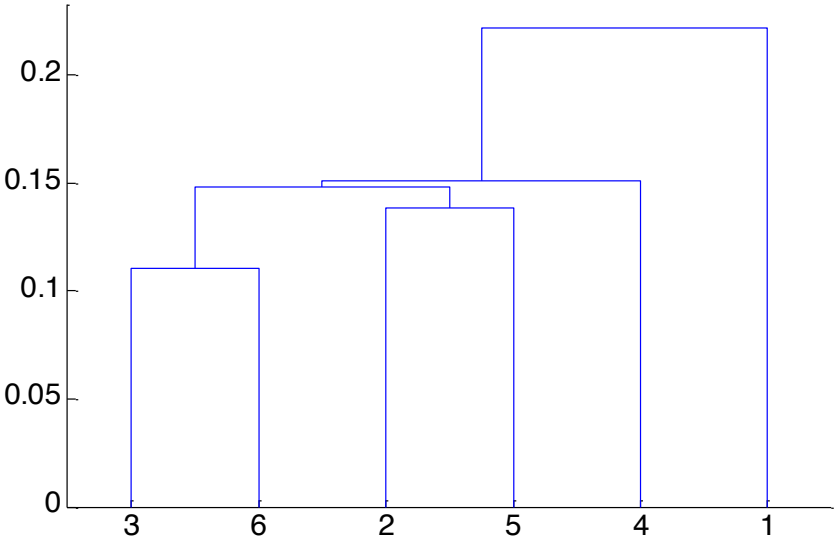
- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.

	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





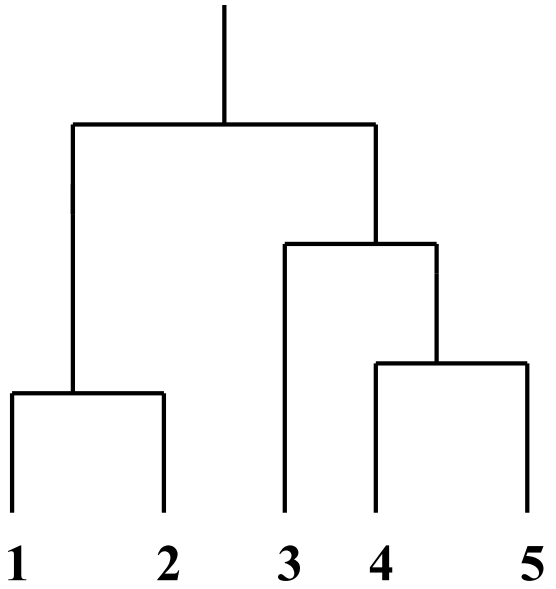
Nested Clusters

Dendrogram

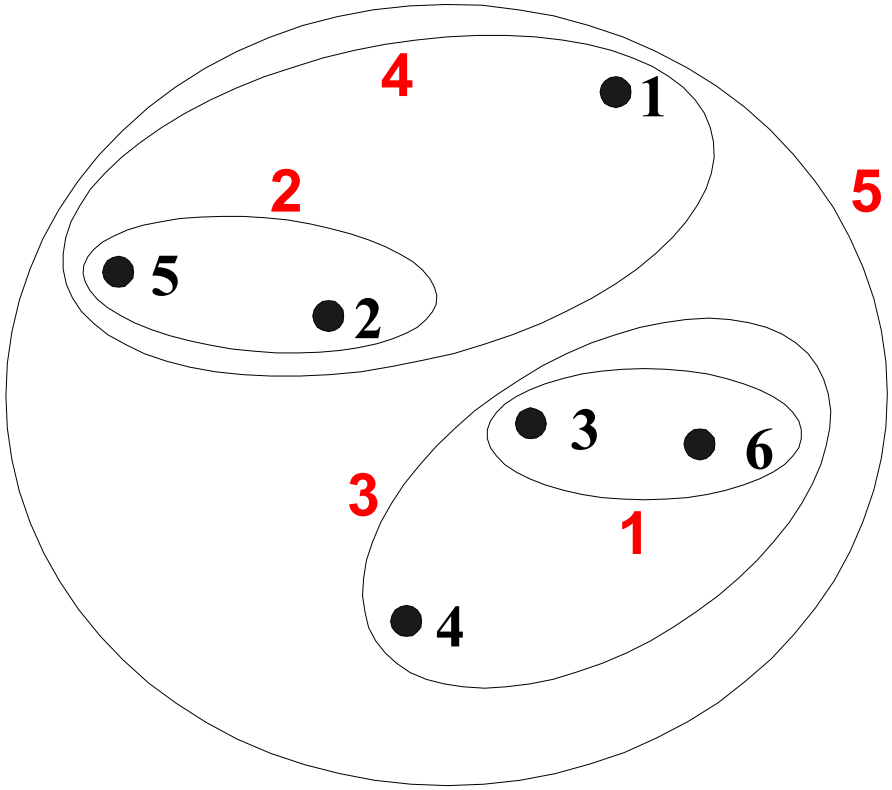
Cluster Similarity: MAX or Complete Link

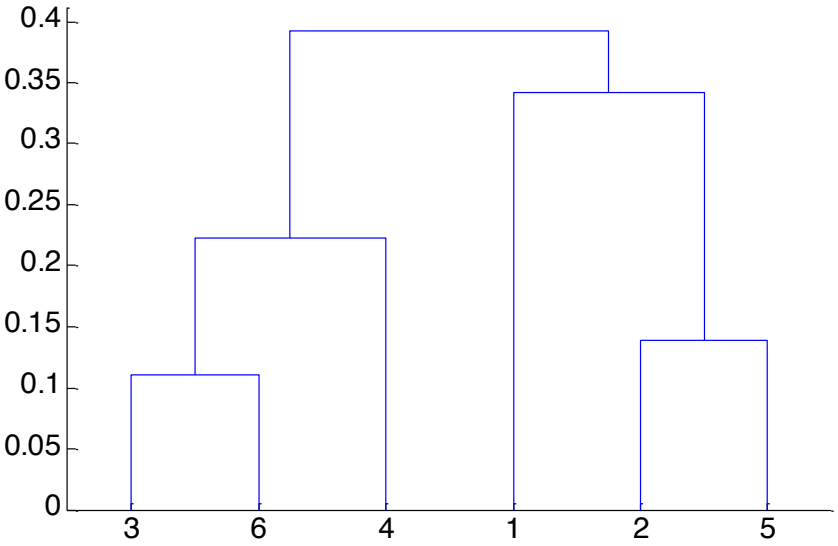
- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters
  - Determined by all pairs of points in the two clusters

	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

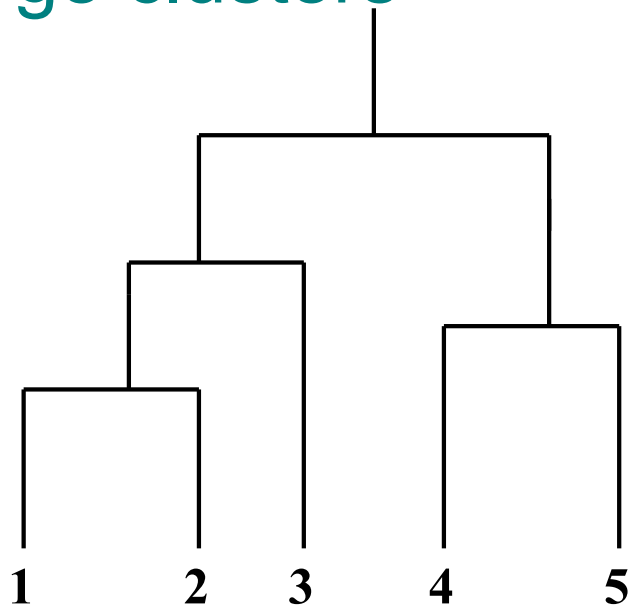
Cluster Similarity: Group Average

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

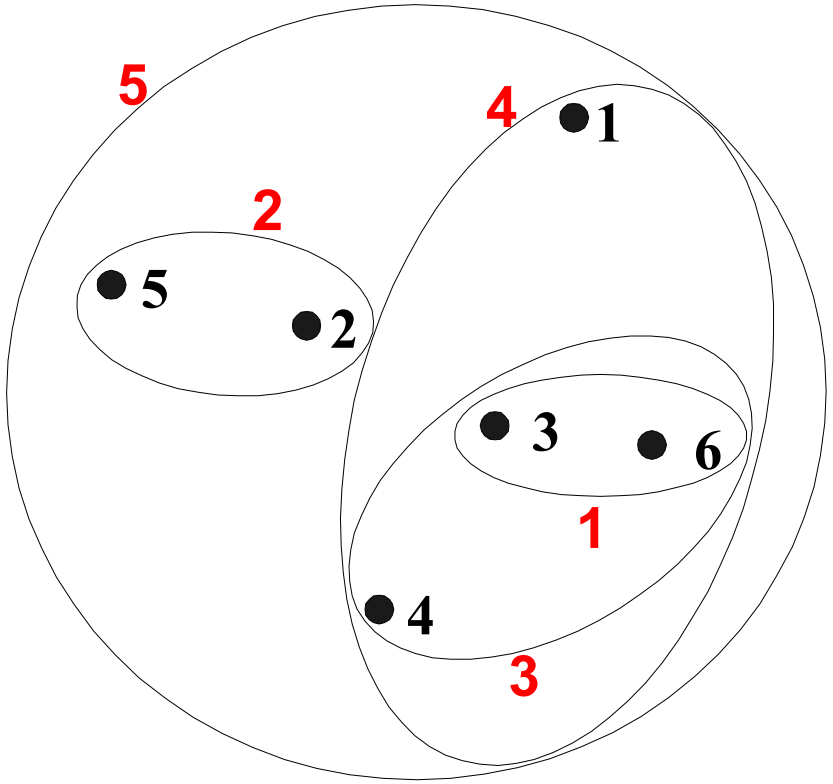
$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| * |\text{Cluster}_j|}$$

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

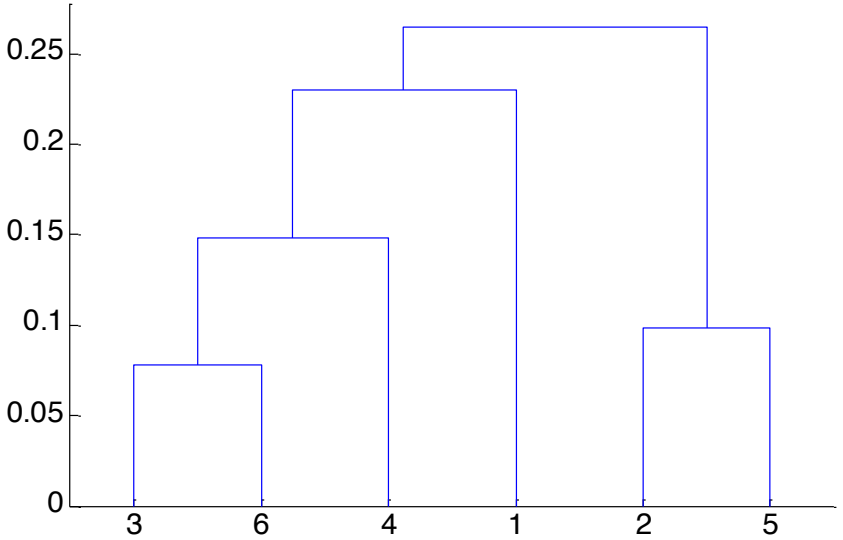
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: Group Average



Nested Clusters

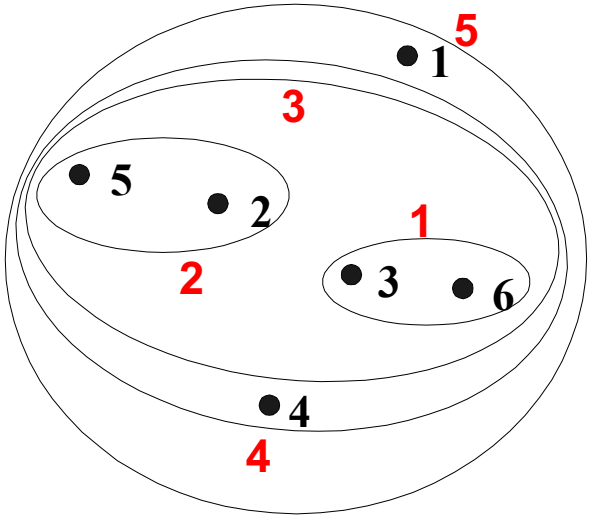


Dendrogram

Cluster Similarity: Ward's Method

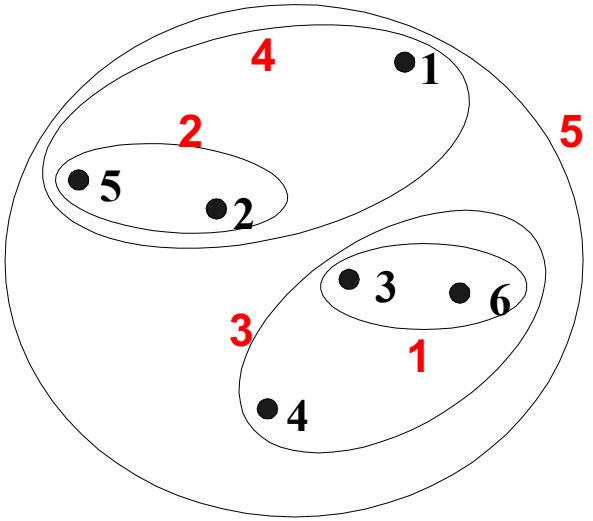
- Similarity of two clusters is based on the *increase in squared error* when two clusters are merged
  - Similar to group average if distance between points is distance squared
- Less susceptible to noise and outliers
- Biased towards globular clusters
- Hierarchical analogue of K-means
  - Can be used to initialize K-means

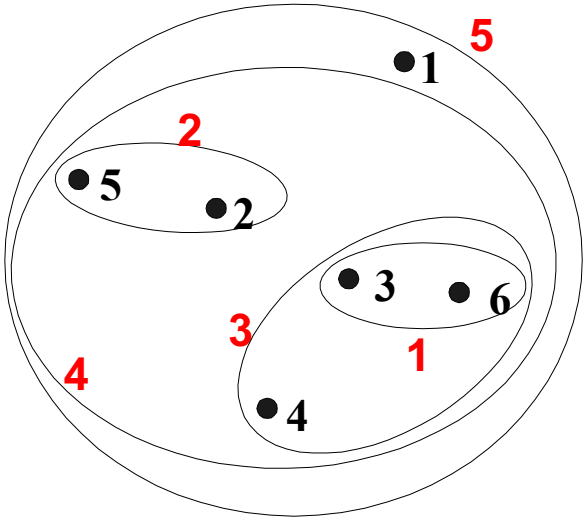
Hierarchical Clustering: Comparison



MIN

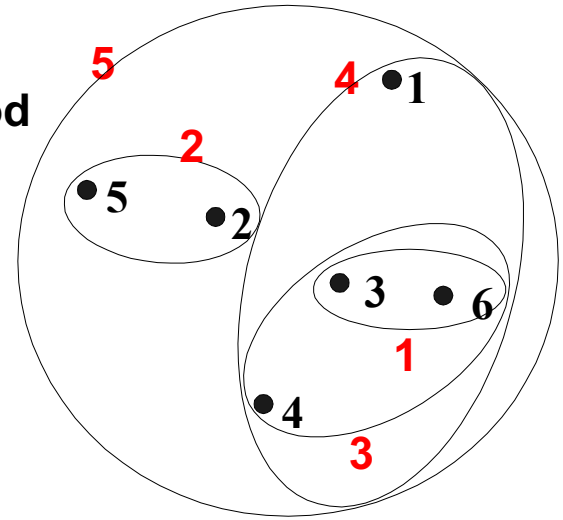
MAX





Group Average

Ward's Method



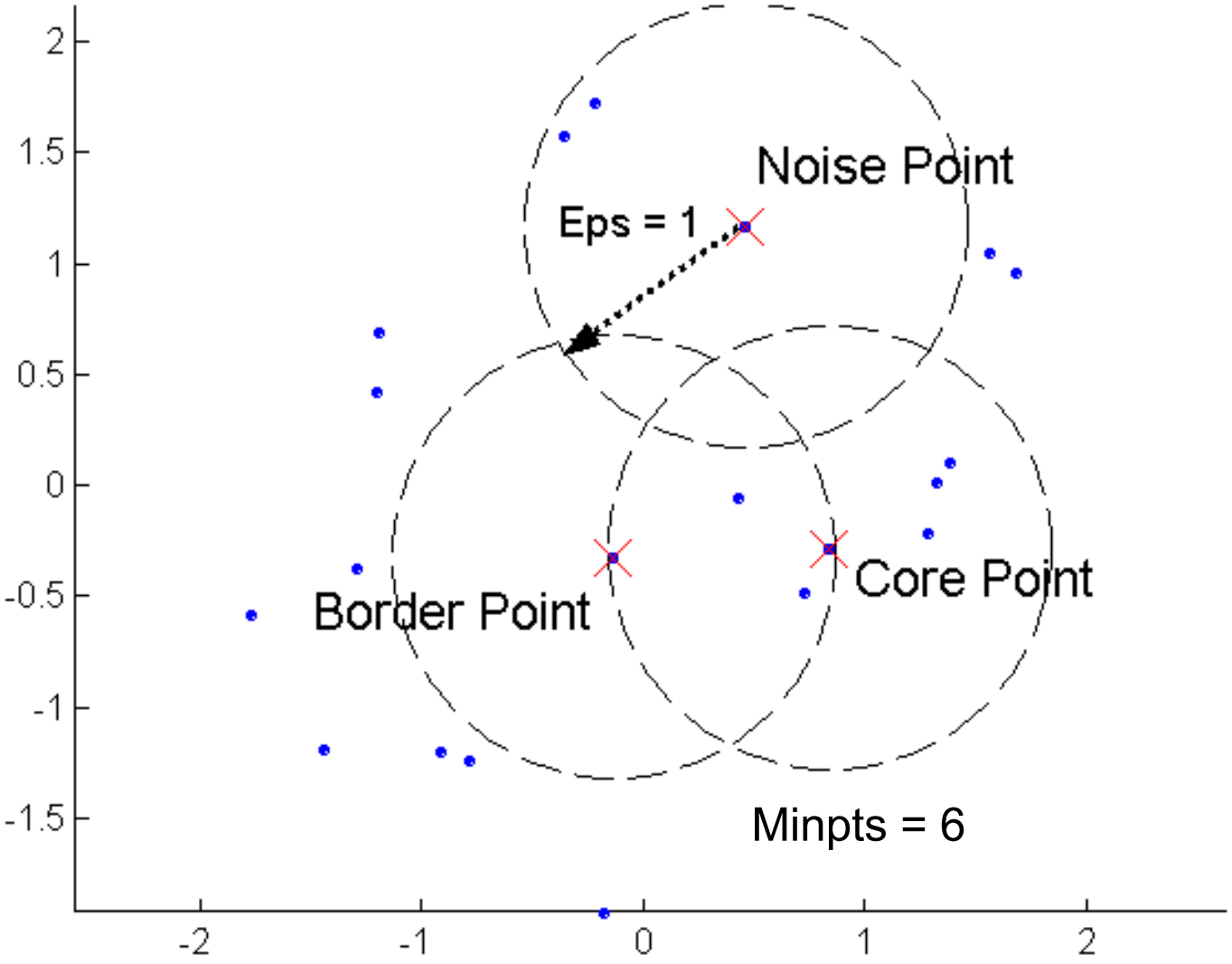
HC: Problems and Limitations

- 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

DBSCAN

- DBSCAN is a density-based algorithm.
  - Density = number of points within a specified radius
  - Density threshold expressed in two parameters: Eps (radius) and MinPts (minimum points)
- A point is a core point if it meets the density threshold, i.e., has more than a specified number of points (MinPts) within Eps
  - These are points that are at the interior of a cluster
- A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point
- A noise point is any point that is neither a core point nor a border point.

DBSCAN: Core, Border, and Noise Points

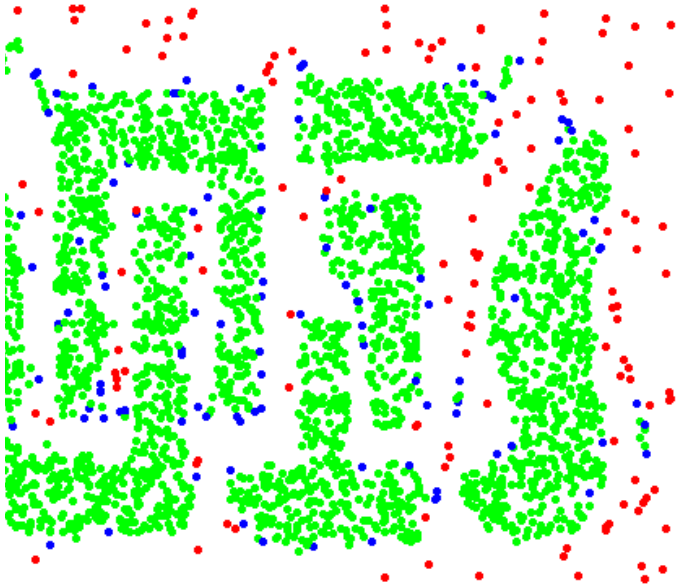


DBSCAN Algorithm

1. Scan all points to label all points as core, border and noise points
2. Eliminate noise points
3. Perform clustering on the remaining points
   a) Put an edge between all core points within Eps of each other
   b) Make each group of connected core points as a separate cluster
   c) Assign each border points to one of the clusters o its associated core points

DBSCAN: Core, Border and Noise Points

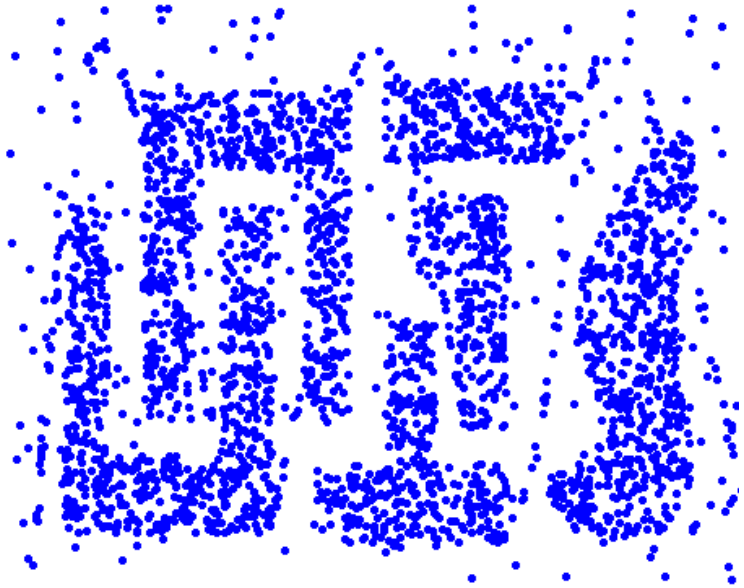
Original Points



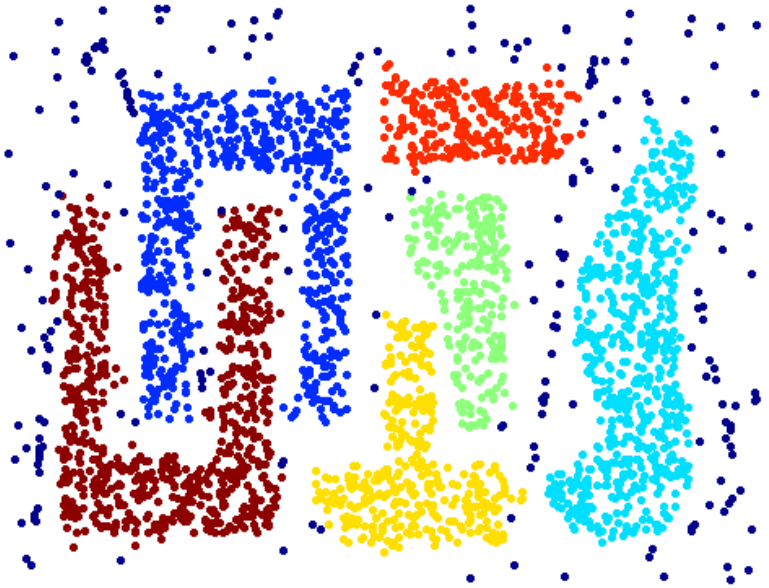
Point types: core, border and noise

Eps = 10, MinPts = 4

When DBSCAN Works Well



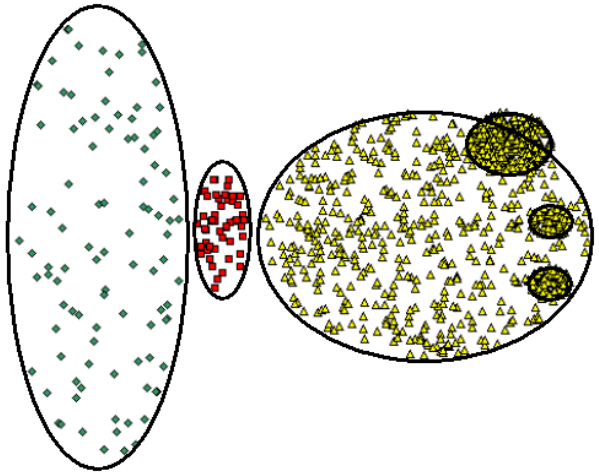
Original Points



Clusters

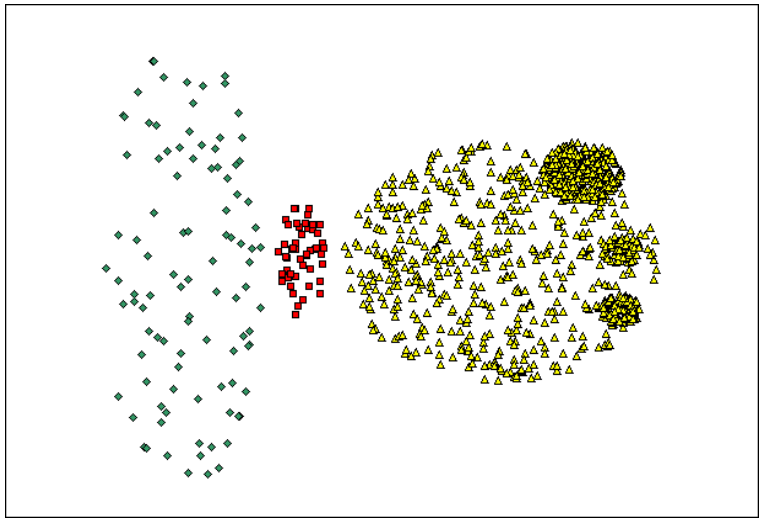
- Resistant to Noise
- Can handle clusters of different shapes and sizes

When DBSCAN Does NOT Work Well

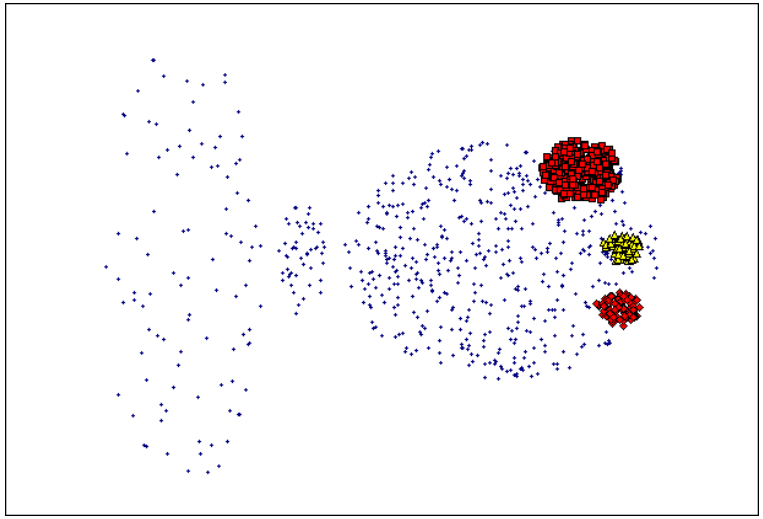


Original Points

- Varying densities
- High-dimensional data



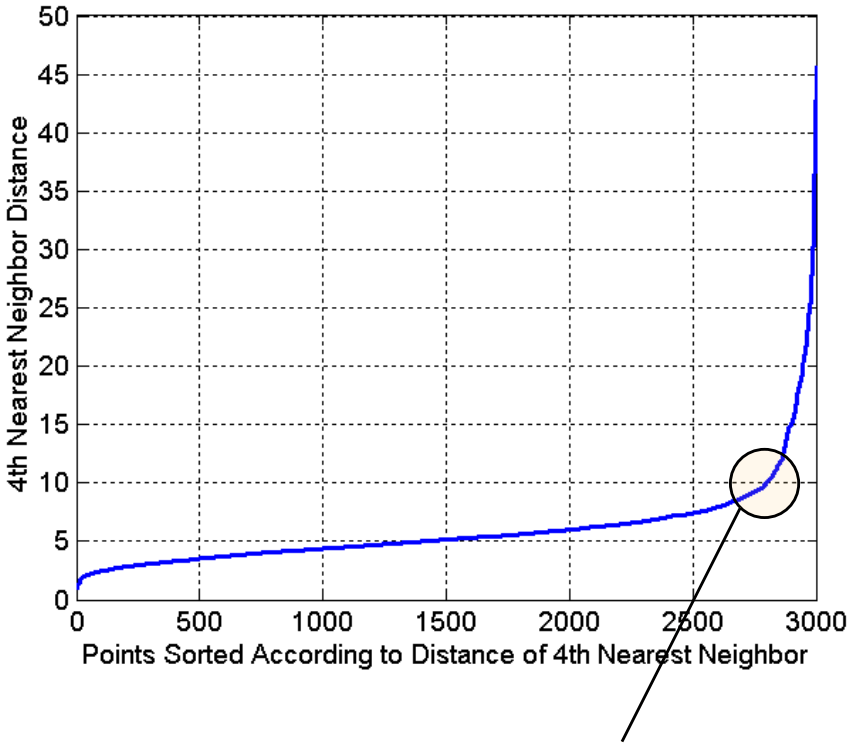
(MinPts=4, Eps=9.92).



(MinPts=4, Eps=9.75)

DBSCAN: Determining EPS and MinPts

- Idea is to observe the behavior of k-dist
- For points in a cluster, their k^{th} nearest neighbors are at roughly the same distance
- Noise points have the k^{th} nearest neighbor at farther distance
- So, plot sorted distance of every point to its k^{th} nearest neighbor



Eps=10
Minpts=4

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
- In the simplest case, clusters are connected components in the graph.
  - Initially there is only one cluster…

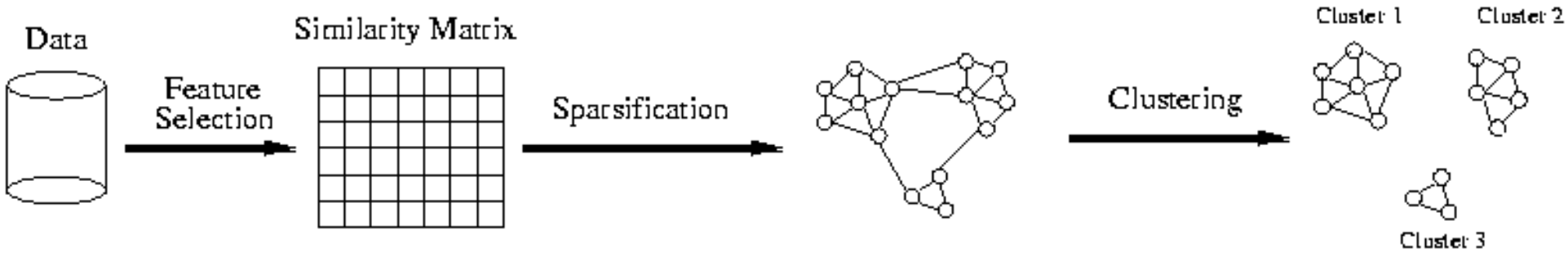
Graph-Based Clustering: Sparsification

- Sparsification is a process to eliminate those edges/links with no/low relevance
  - Filter using a given threshold, or
  - Keep k nearest (most relevant) neighbors for each node
  - How about DBSCAN? MST?
- Advantage: the amount of data that needs to be processed is drastically reduced
  - Sparsification could 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



Chameleon: Clustering Using Dynamic Modeling

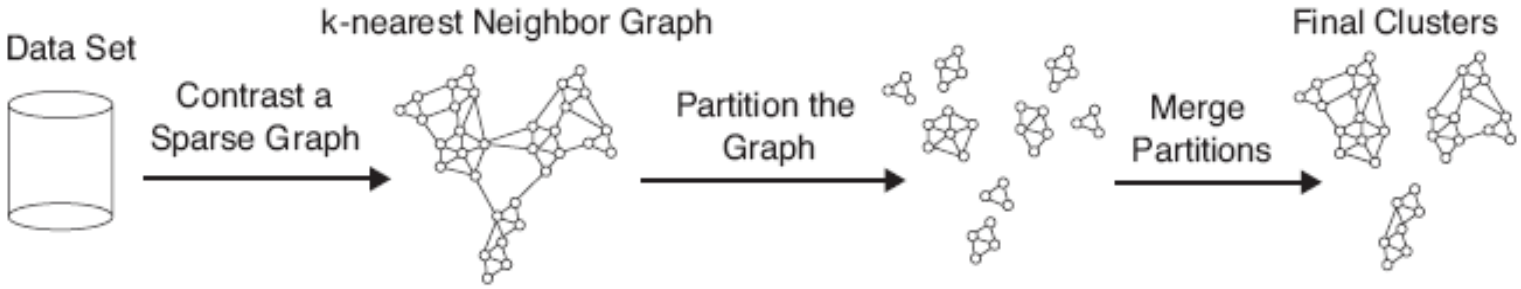
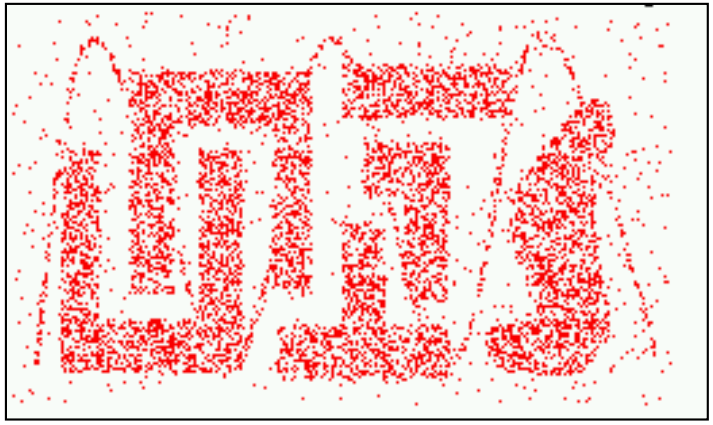


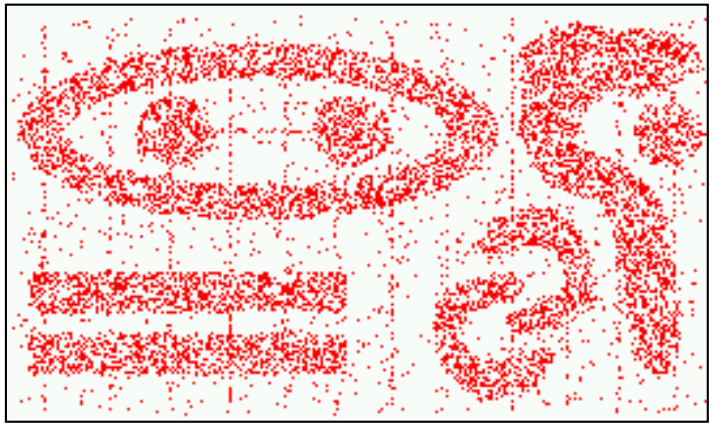
Figure 9.20. Overall process by which Chameleon performs clustering. ©1999, IEEE

- Adapt to the characteristics of the data set to find the natural clusters
- Use a dynamic model to measure the similarity between clusters
- 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
- Densities across clusters may be different
- Density within clusters may vary
- Existence of special artifacts and noise





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

Chameleon: Motivation

- 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* (pair-wise similarity)

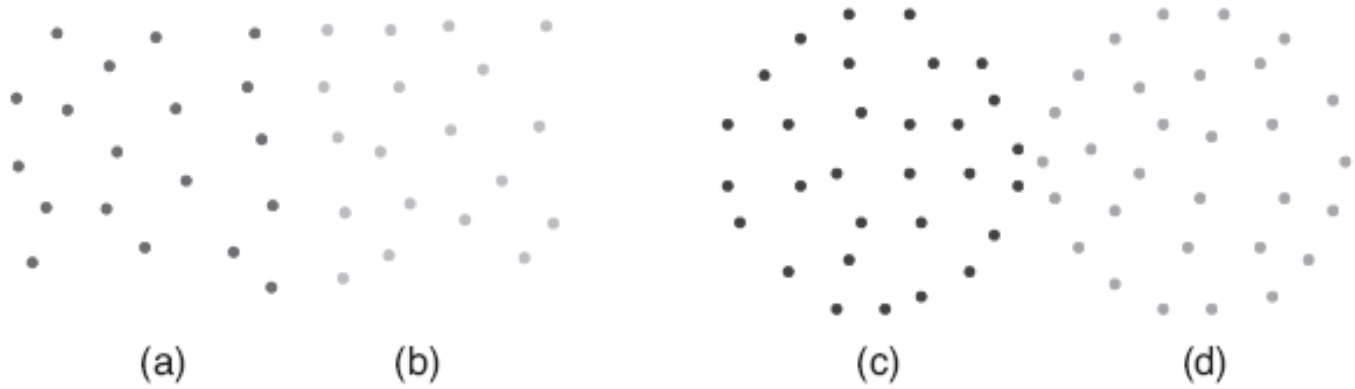
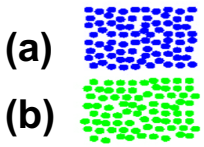
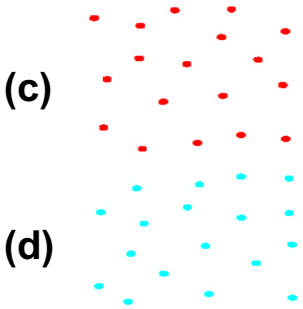


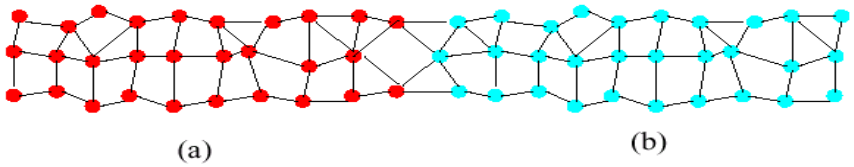
Figure 9.17. Situation in which closeness is not the appropriate merging criterion. ©1999, IEEE

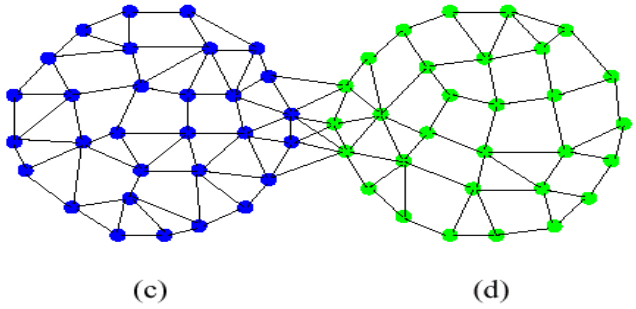
Chameleon: Clustering Using Dynamic Modeling

- Two clusters are combined if the resulting cluster shares certain *properties* with the constituent clusters
  - Main property is the *relative closeness* and *relative inter-connectivity* of the cluster
  - The merging scheme preserves *self-similarity*









Typical closeness schemes will merge (a) and (b)

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

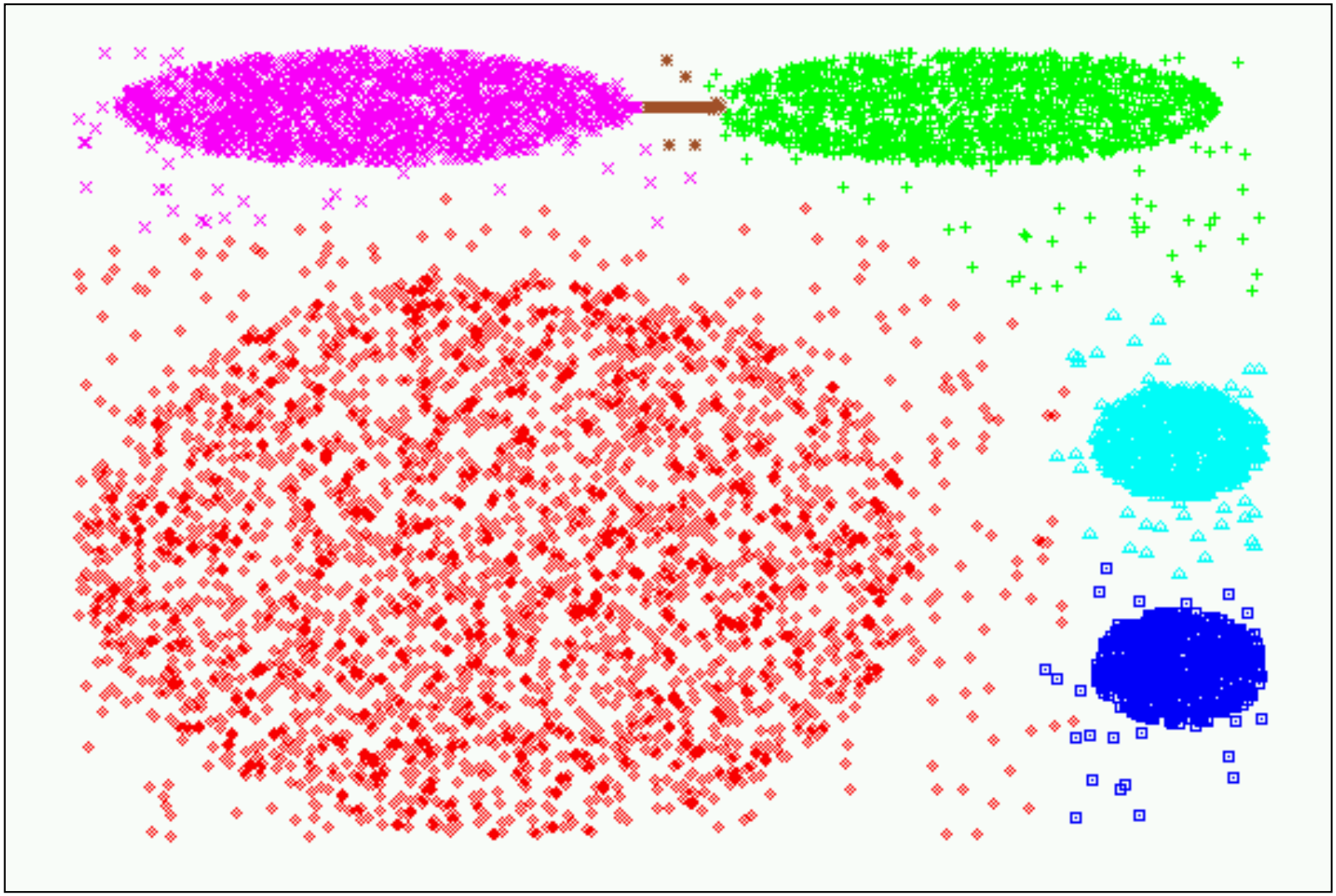
Chameleon: Steps

- *Preprocessing*: Represent the Data by a Graph
  - Construct the k-NN graph to capture the relationship between a point and its k nearest neighbors
  - Data points far apart are disconnected, and the edge weights capture the population density of the space.
  - 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
- *Why not stop with Phase-I? We've got the clusters, haven't we?*

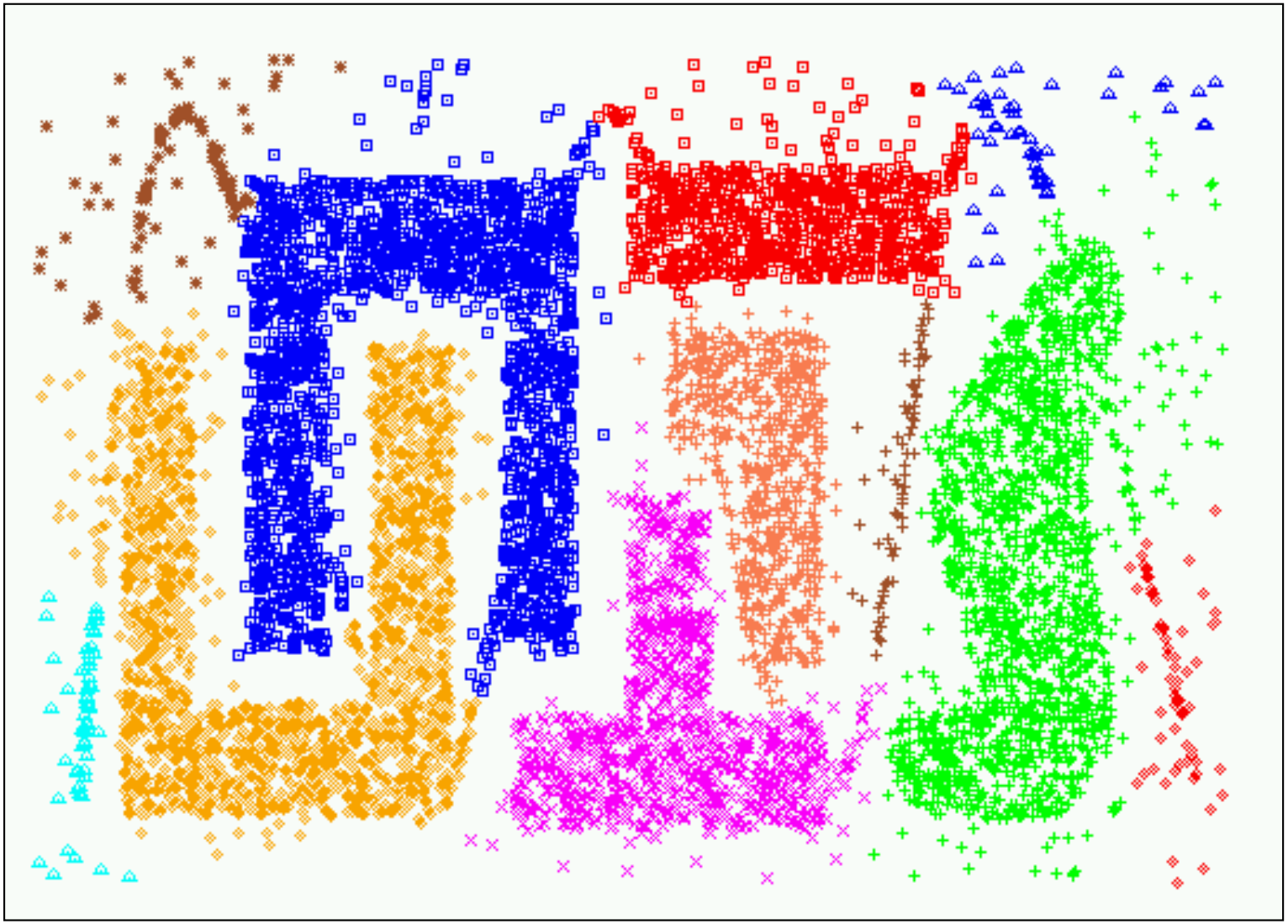
Chameleon: Steps …

- *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:
    - Absolute Interconnectivity: defined in terms of *edge cut*, i.e., the total weight (*sum*) of the edges that connect the two clusters
    - Relative Interconnectivity (RI): Absolute interconnectivity of two clusters normalized by the internal connectivity of the clusters
    - Absolute Closeness: the *average* weight of the edges that connect vertices in two clusters
    - Relative Closeness (RC): Absolute closeness of two clusters normalized by the internal closeness of the clusters
  - RI and RC are combined to model the self-similarity

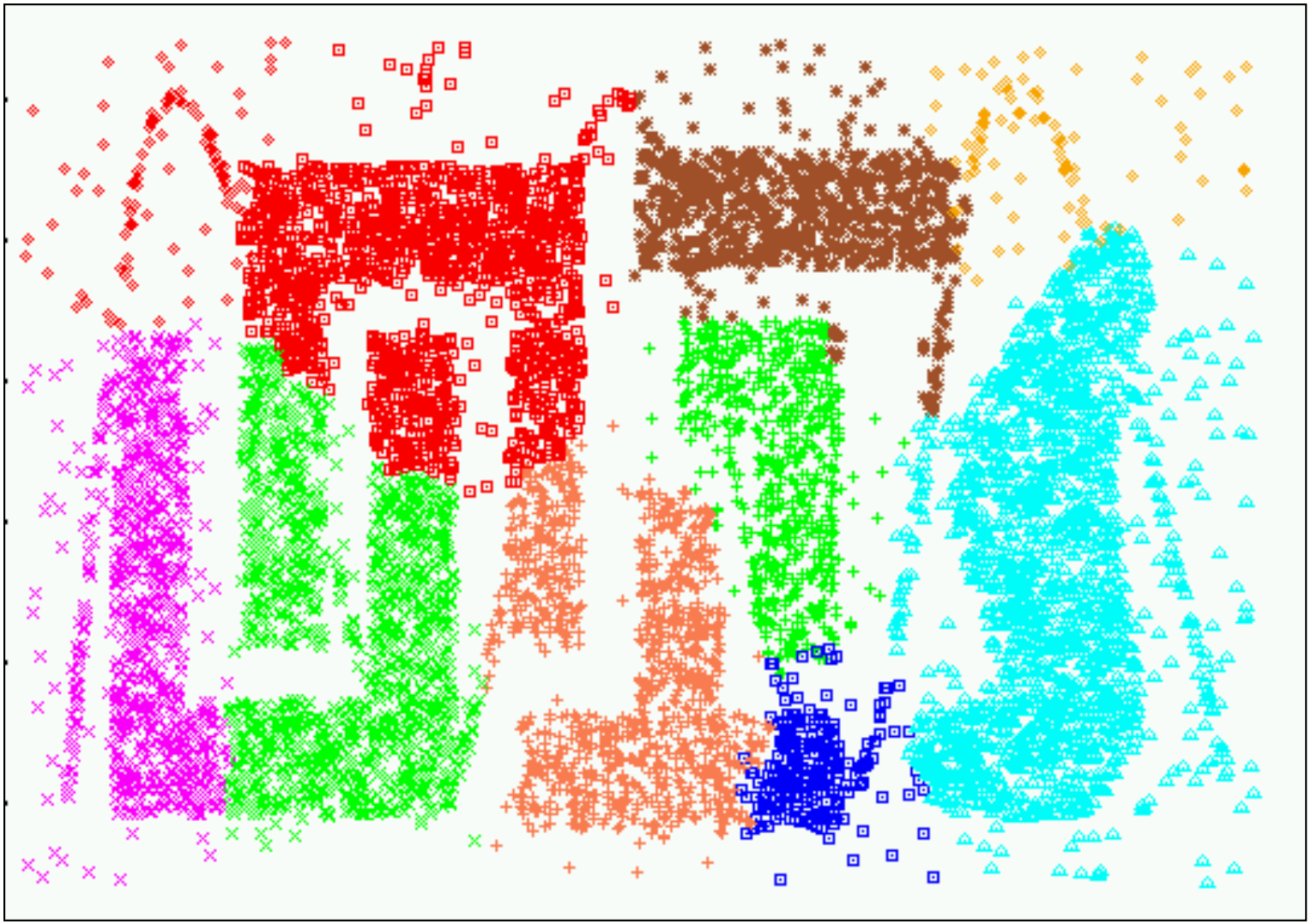
Experimental Results: CHAMELEON



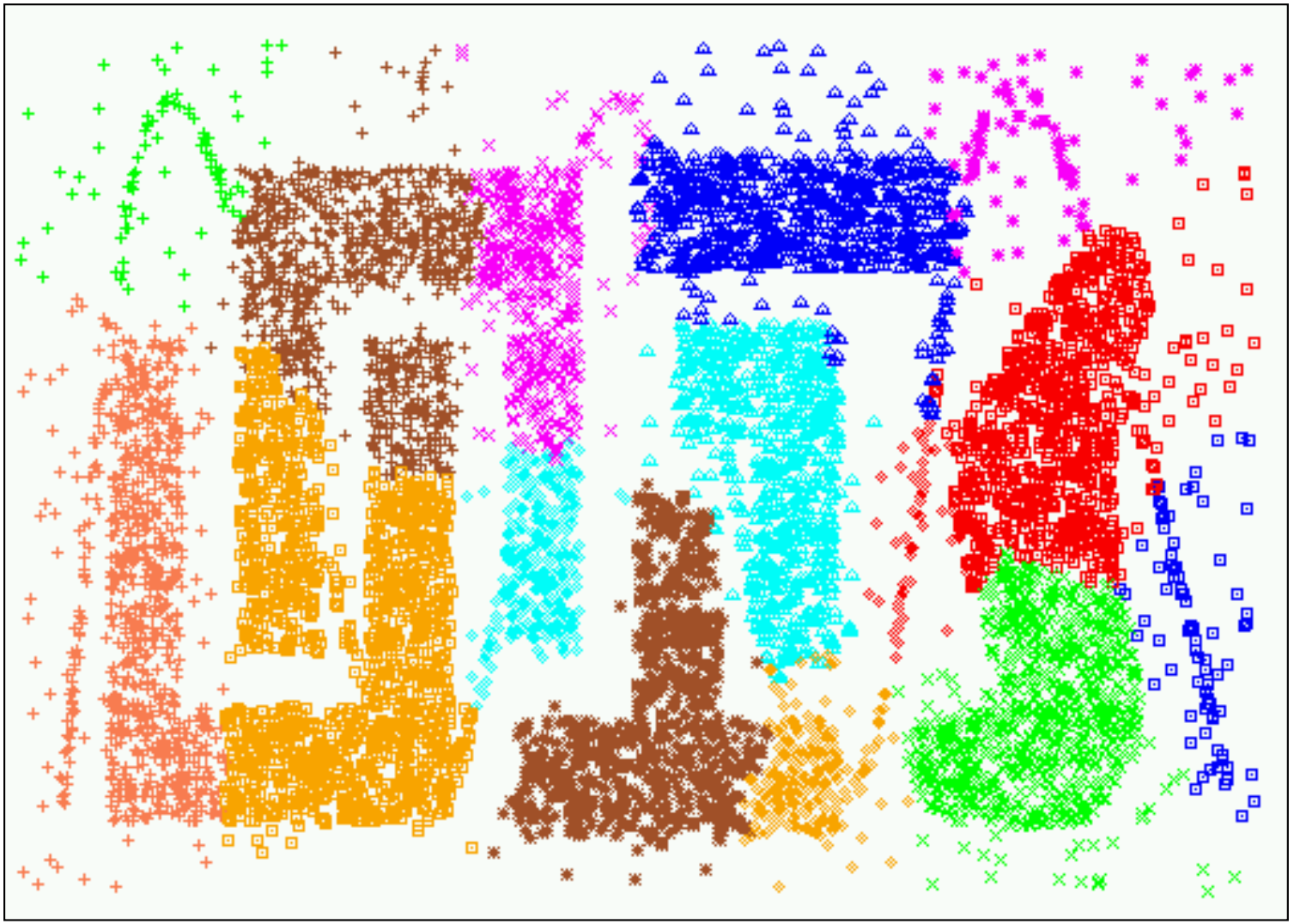
Experimental Results: CHAMELEON



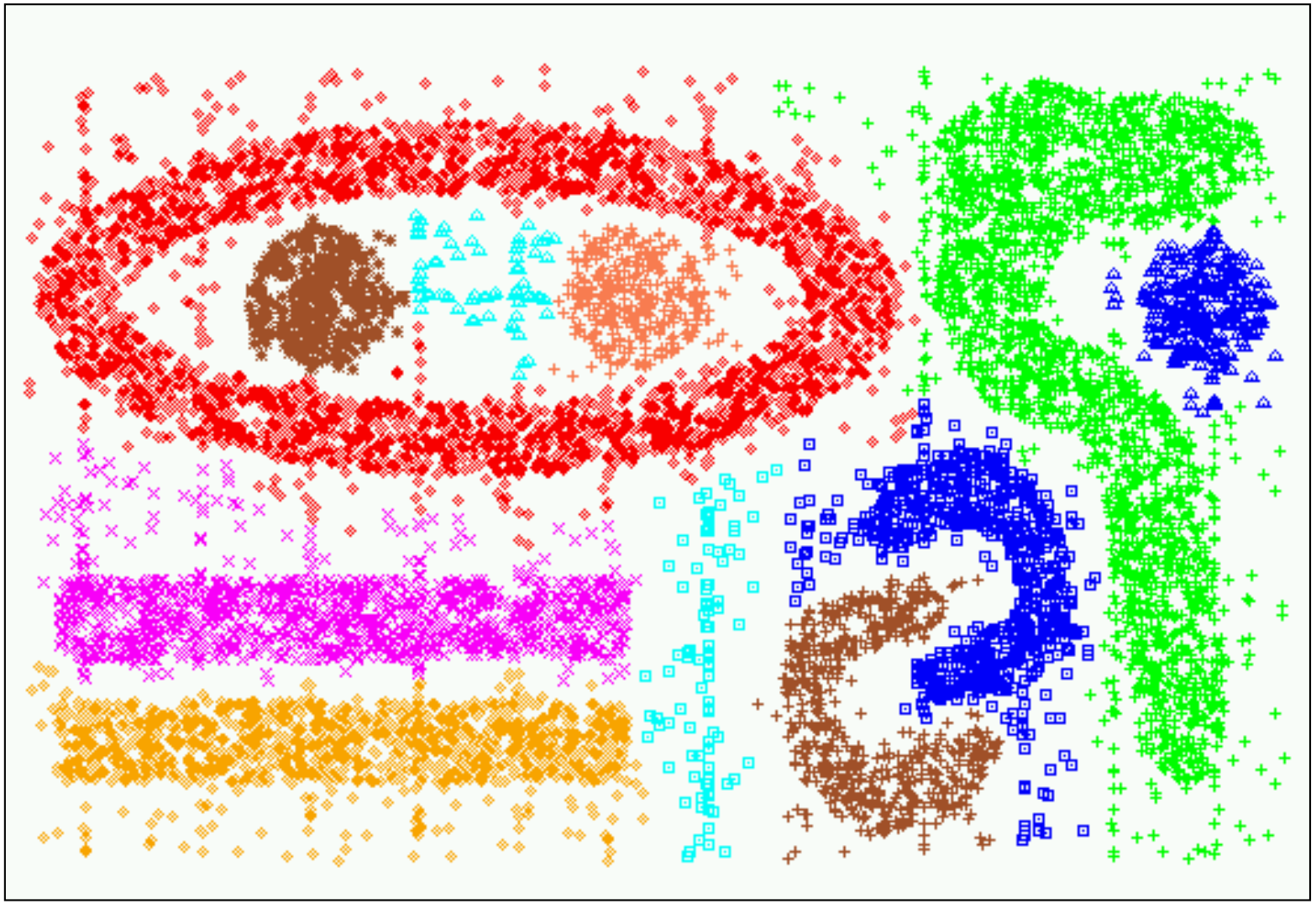
Experimental Results: CURE (*9 clusters*)



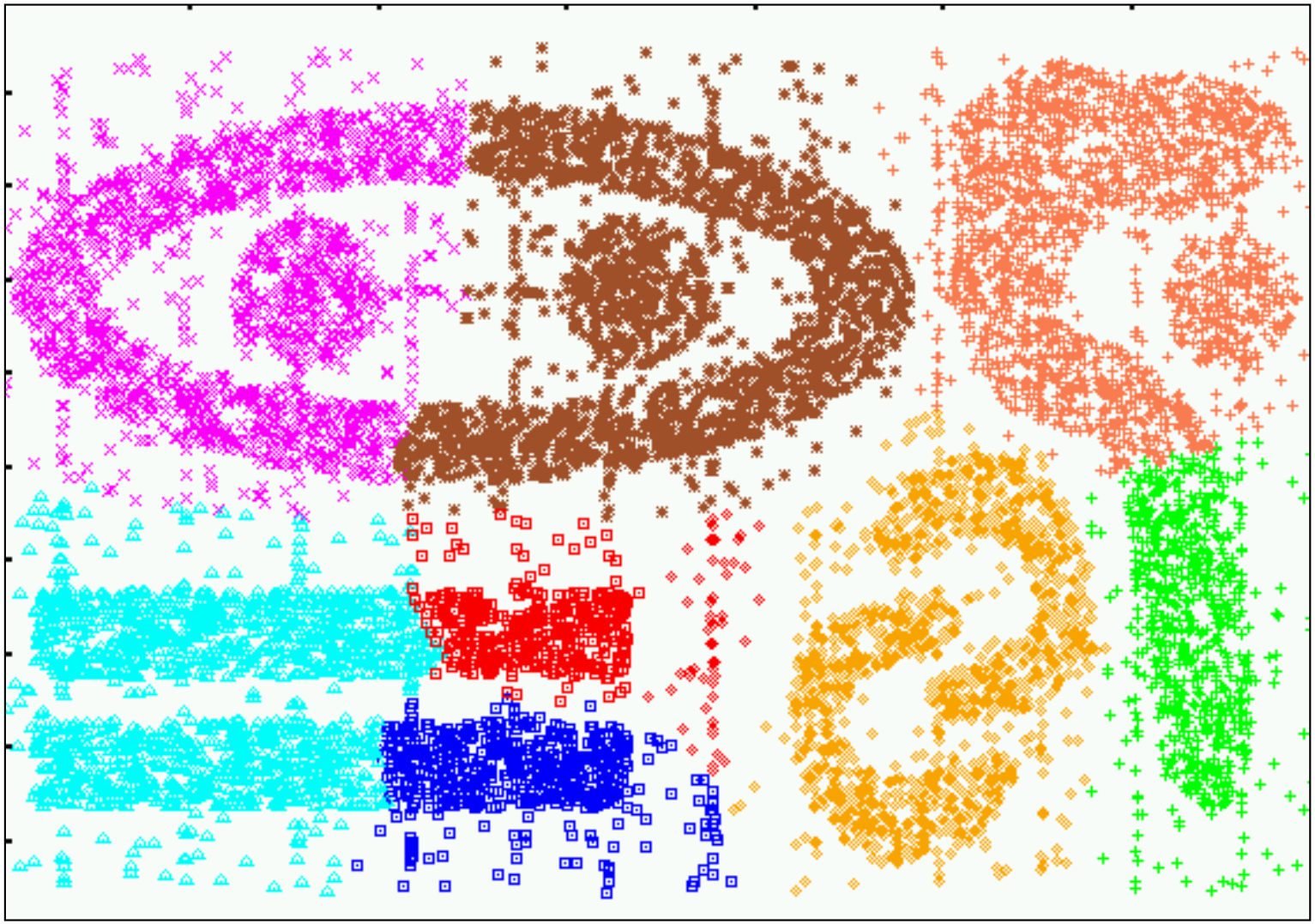
Experimental Results: CURE (*15 clusters*)



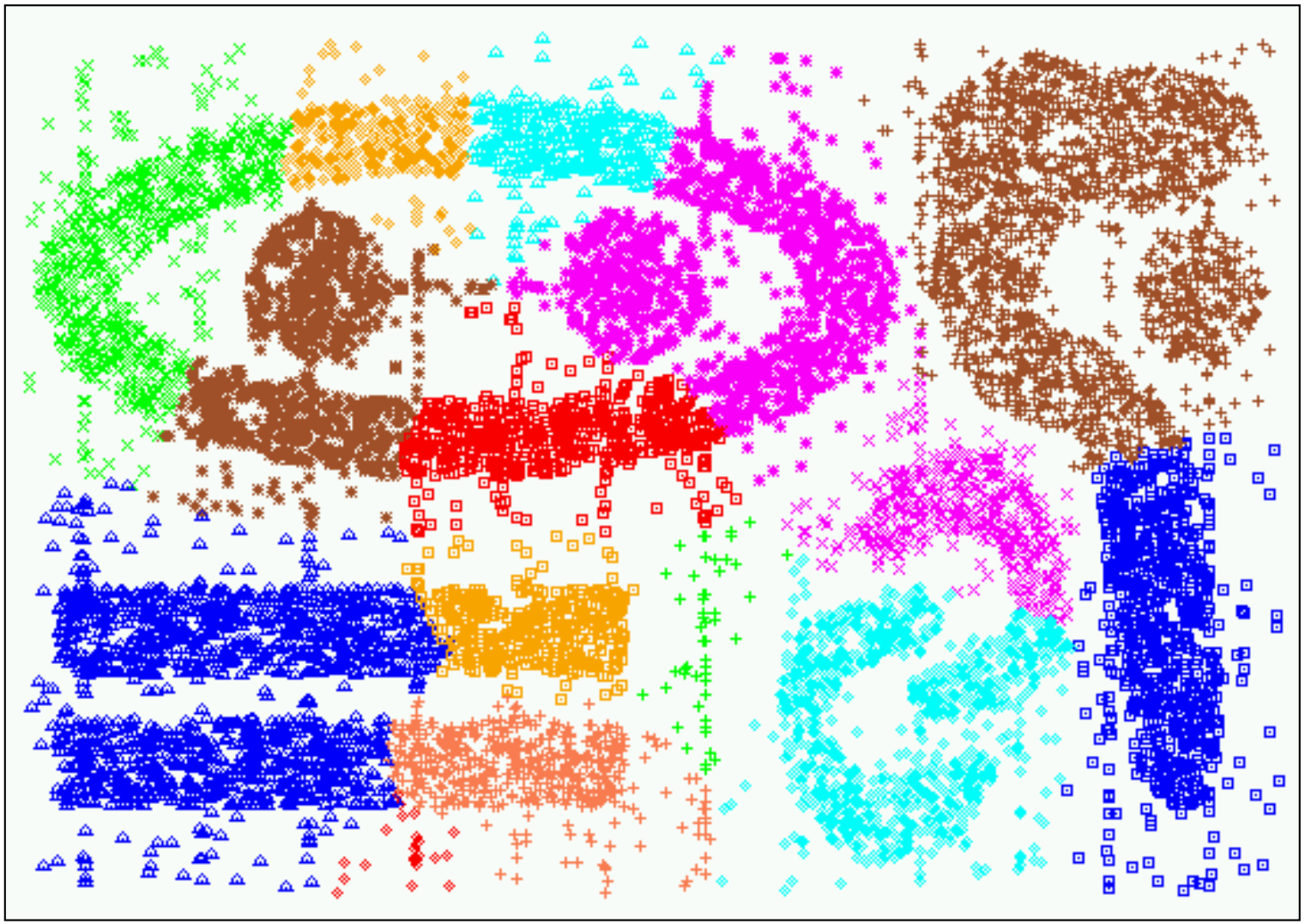
Experimental Results: CHAMELEON



Experimental Results: CURE (*9 clusters*)



Experimental Results: CURE (*15 clusters*)



Clusters with Different Density

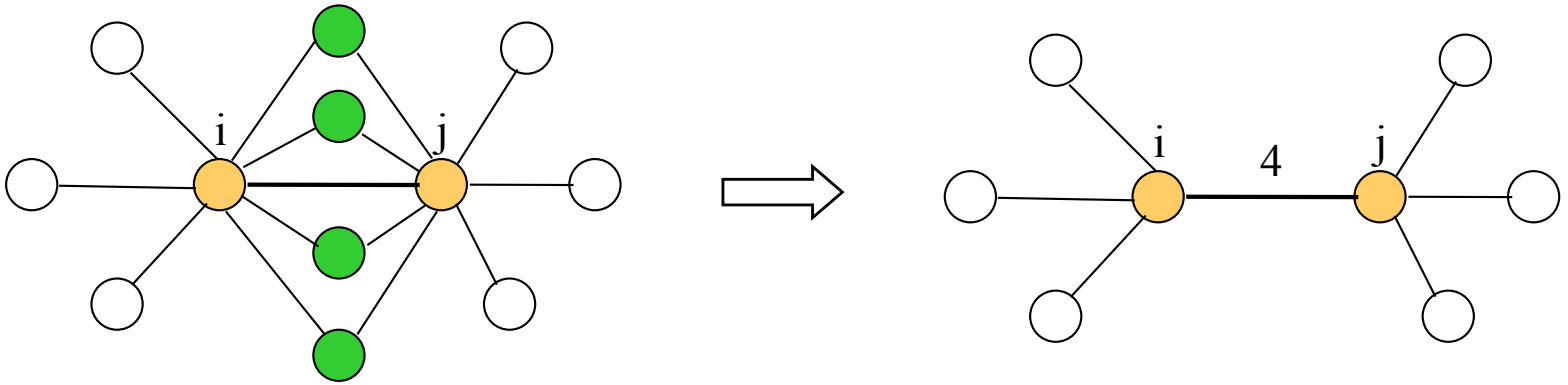
Figure 9.22. Two circular clusters of 200 uniformly distributed points.

- Distance (similarity) does not provide a good guidance for clustering.

Explore different notion of similarity!

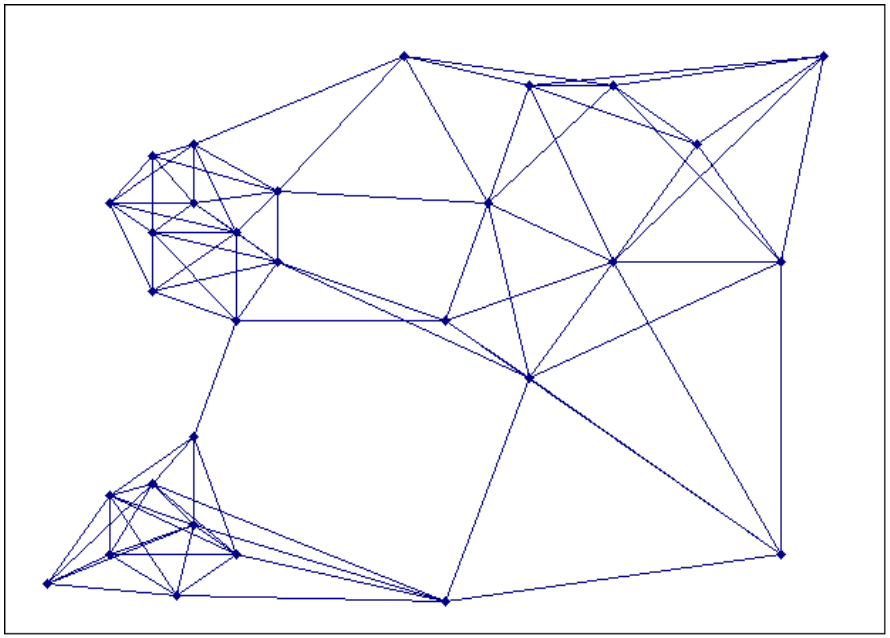
Shared Near Neighbors (SNN)

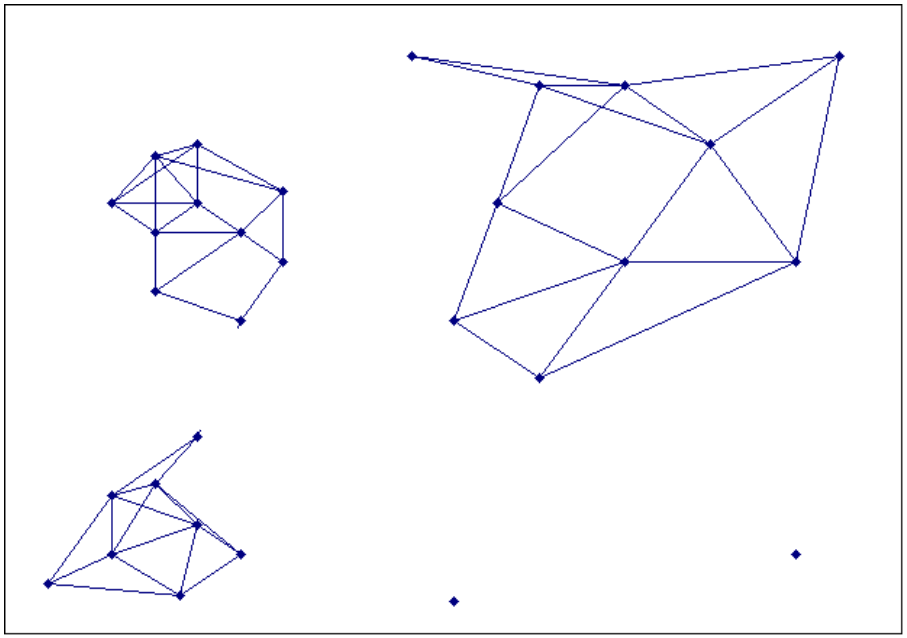
- Two items sharing many similar neighbors are similar.
- SNN Graph: Given a pair of vertices which are in each other's NN list, the weight of their edge is the number of shared neighbors between them. Two parameters:
  - NE: the number of *Neighbors to Examine*
  - NC: the minimum required number of *Neighbors in Common*



  - SNN Based algorithms, e.g., ROCK and Jarvis-Patrick.

Creating the SNN Graph





Sparse Graph

Link weights are similarities between neighboring points

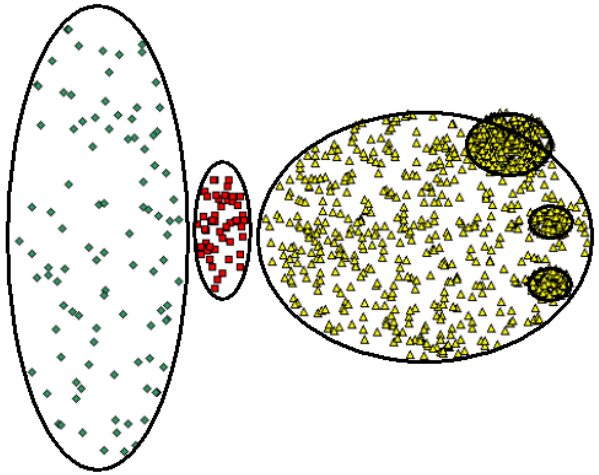
Shared NN Graph

Link weights are the number of Shared Nearest Neighbors

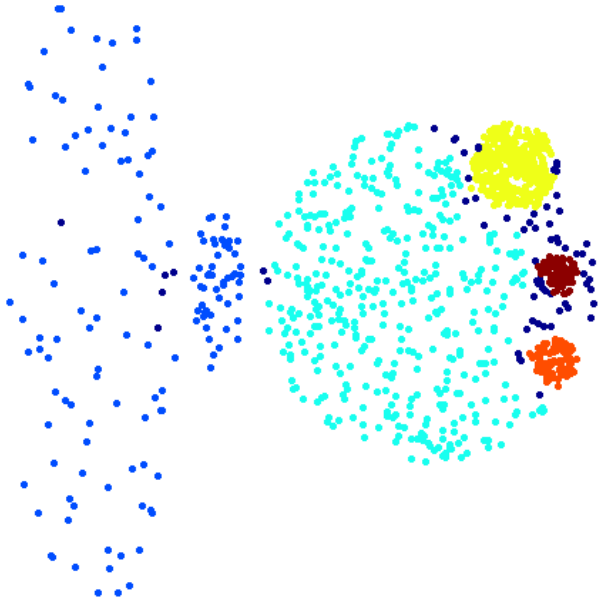
Jarvis-Patrick Clustering

- First find the *NE* nearest neighbors of all points
  - In graph terms, this removes all but the NE strongest links from a point to other points in the proximity graph
- Generate SNN, i.e., a pair of points is linked if
  - the two points are in each others' *NE* nearest neighbors list, and the two points share more than *NC* neighbors.
  - e.g., 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
- Sparsify the SNN graph by applying a similarity threshold
- The connected components remaining are clusters.

When Jarvis-Patrick Works Reasonably Well



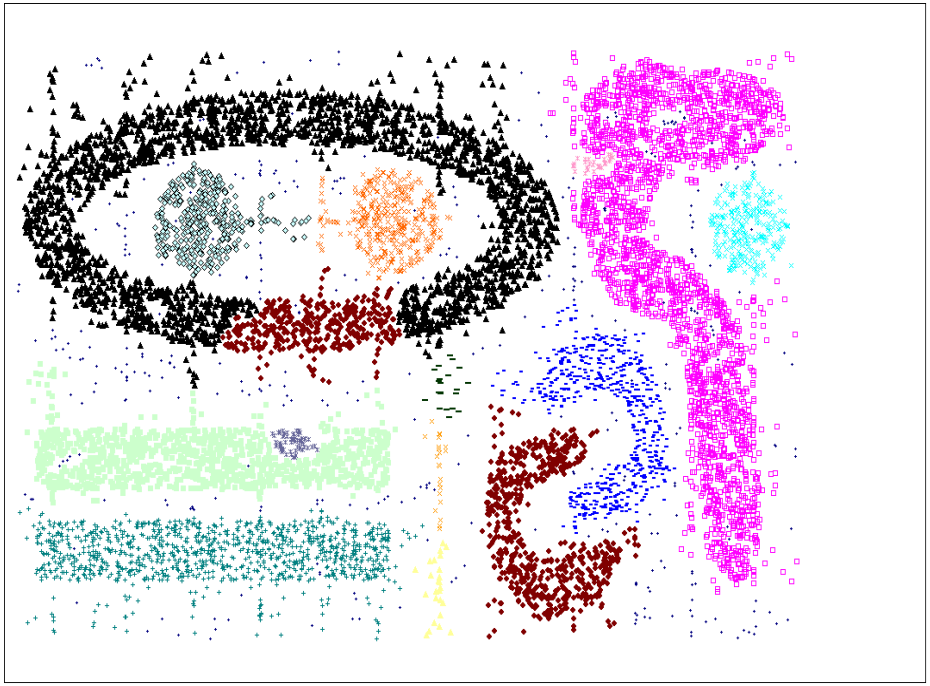
Original Points



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

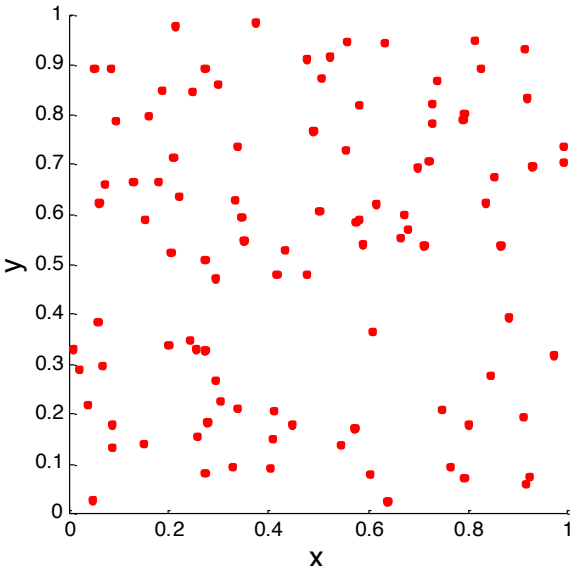
J-P algorithm deals with noise and handle clusters of different sizes, shapes and density very well. However, it's sensitive to the threshold setting.

Cluster Validity

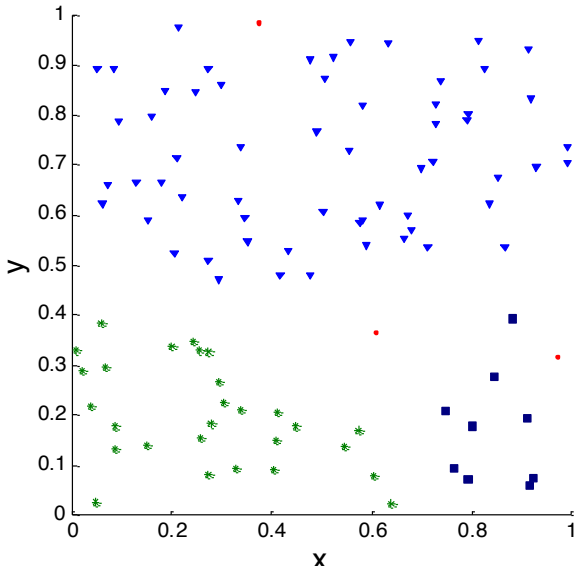
- For supervised classification, we have a variety of measures to evaluate how good our model is
  - Accuracy, precision, recall
- For cluster analysis, how to evaluate the "goodness" of the resulting clusters?
  - But "*clusters are in the eye of the beholder*"!
- Why do we still want to evaluate them?
  - To avoid finding patterns in noise
  - To compare clustering algorithms
  - To compare two clusterings
  - To compare two clusters

Clusters found in Random Data

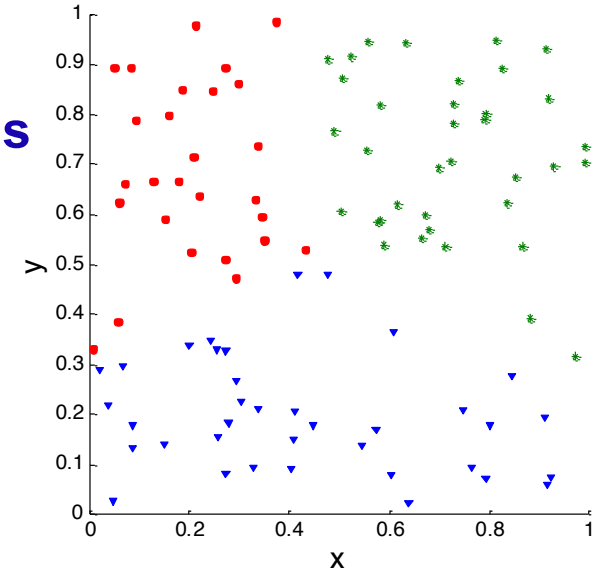
Random Points



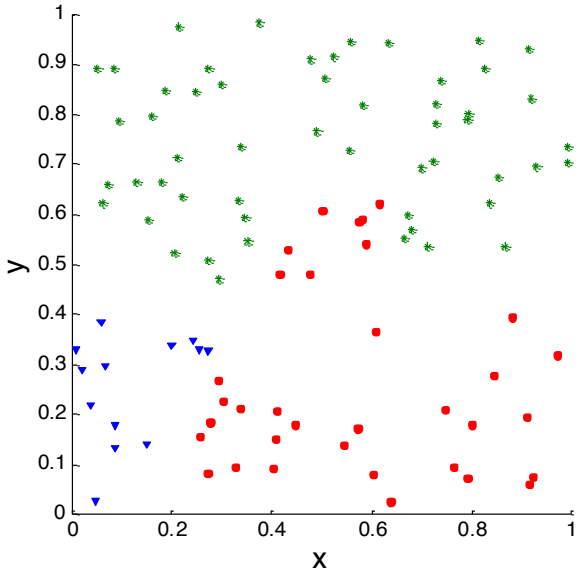
DBSCAN



K-Means



Complete Link (MAX)



Aspects of Cluster Validation

1. Determining the *clustering tendency* of data, i.e., whether non-random structure actually exists.
2. Comparing the results of a cluster analysis to *externally known results*, e.g., given class labels.
3. Evaluating how well the results of a cluster analysis fit the data *without reference to external information*, i.e., use only the data
4. Comparing the results of two different cluster analyses to determine which is *relatively* better.
5. Determining the 'correct' number of clusters.

For 2, 3, and 4, we further distinguish whether we want to evaluate the *entire clustering* or just *individual clusters*.

Measures of Cluster Validity

- Numerical measures for judging various aspects of cluster validity are classified into three types:
  - *External Index:* Used to measure the extent to which cluster labels match externally supplied class labels.
    - Entropy or purity.
  - *Internal Index:* Used to measure the goodness of a clustering structure *without* respect to external information.
    - Sum of Squared Error (SSE)
  - *Relative Index:* Used to compare two different clusterings or clusters.
    - Could be an external or internal index, e.g., SSE or entropy, depending on type of comparison.
- Above sometimes referred to as *criteria* instead of *indices*
  - However, sometimes criterion is the general strategy and index is the numerical measure that implements the criterion.

Clustering Tendency

- *Hopkins Statistic:* H_{ind} statistic is computed as:

$$H_{ind} = \frac{\sum_{i=1}^{n} q_i}{\sum_{i=1}^{n} q_i + \sum_{i=1}^{n} w_i}$$

$\sum_{i=1}^{n} q_i$ — NN distance of a point in the given data set

$\sum_{i=1}^{n} q_i + \sum_{i=1}^{n} w_i$ — NN distance of a point in the random data set

- If objects are uniformly distributed, $\mathbf{q}_i$ and $\mathbf{w}_i$ will be similar, and the statistic will be close to 0.5.
- The more H_{ind} closes to 0, the higher the clustering tendency.
  - If clustering are present, the distances for artificial objects *w* will be larger than for the real ones *q*, because these artificial objects are homogeneously distributed whereas the real ones are grouped together, and the value of H_{ind} will decrease.

Internal Measures: Cohesion and Separation

- Measure validity of individual clusters or clustering
- *Cluster Cohesion:* Measures how closely related are objects in a cluster
  - Example: SSE
- *Cluster Separation:* Measure how distinct or well-separated a cluster is from other clusters
- Example: Squared Error
  - Cohesion is measured by the *Within cluster Sum of Squared errors*

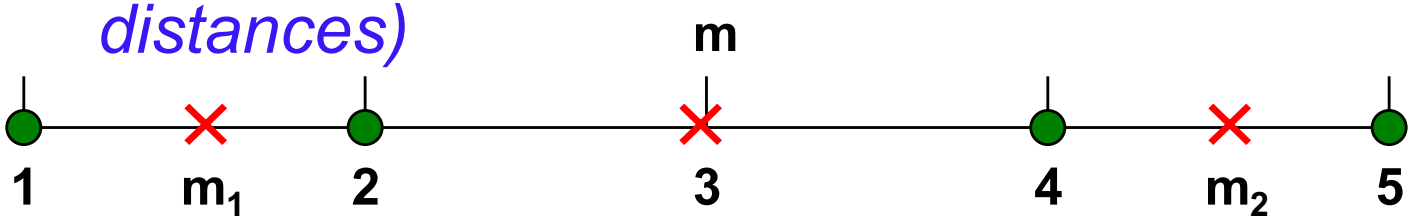
$$WSS = \sum_i \sum_{x \in C_i} (x - m_i)^2$$

  - Separation is measured by the *Between cluster Sum of Squared errors*

$$BSS = \sum_i |C_i| (m - m_i)^2$$

Internal Measures: Cohesion and Separation

- Example: SSE
  - *BSS + WSS = constant (i.e., the sum of pair-wise distances)*



K=1 cluster:

$$WSS = (1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2 = 10$$

$$BSS = 4 \times (3-3)^2 = 0$$

$$Total = 10 + 0 = 10$$

K=2 clusters:

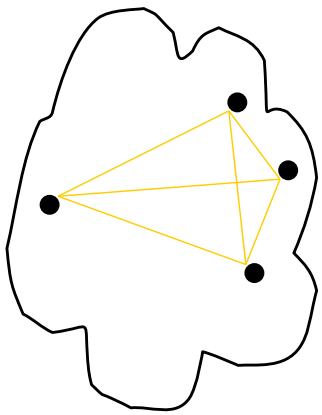
$$WSS = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$

$$BSS = 2 \times (3-1.5)^2 + 2 \times (4.5-3)^2 = 9$$

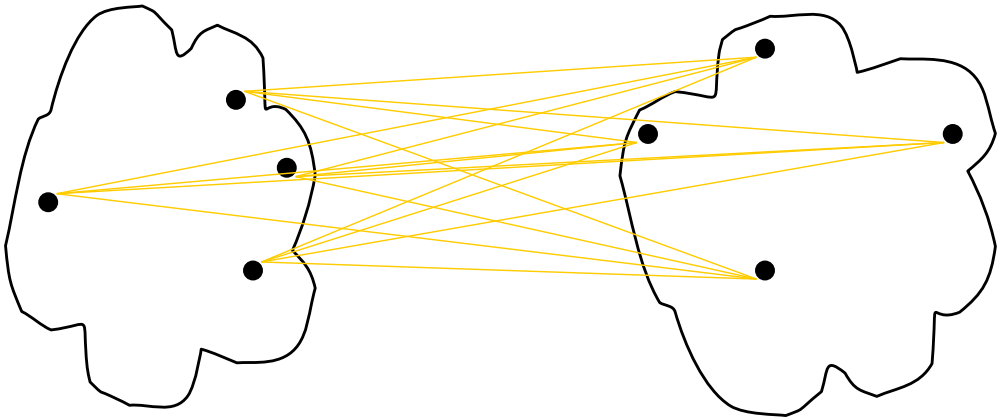
$$Total = 1 + 9 = 10$$

Internal Measures: Cohesion and Separation

- Cohesion and separation can be defined based on graph.
  - Cluster cohesion is the sum of the weight of all links within a cluster.
  - Cluster separation is the sum of the weights between nodes in the cluster and nodes outside the cluster.



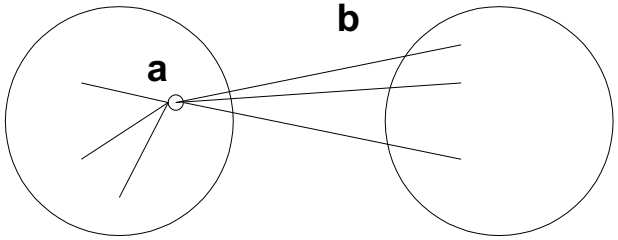
Cohesion



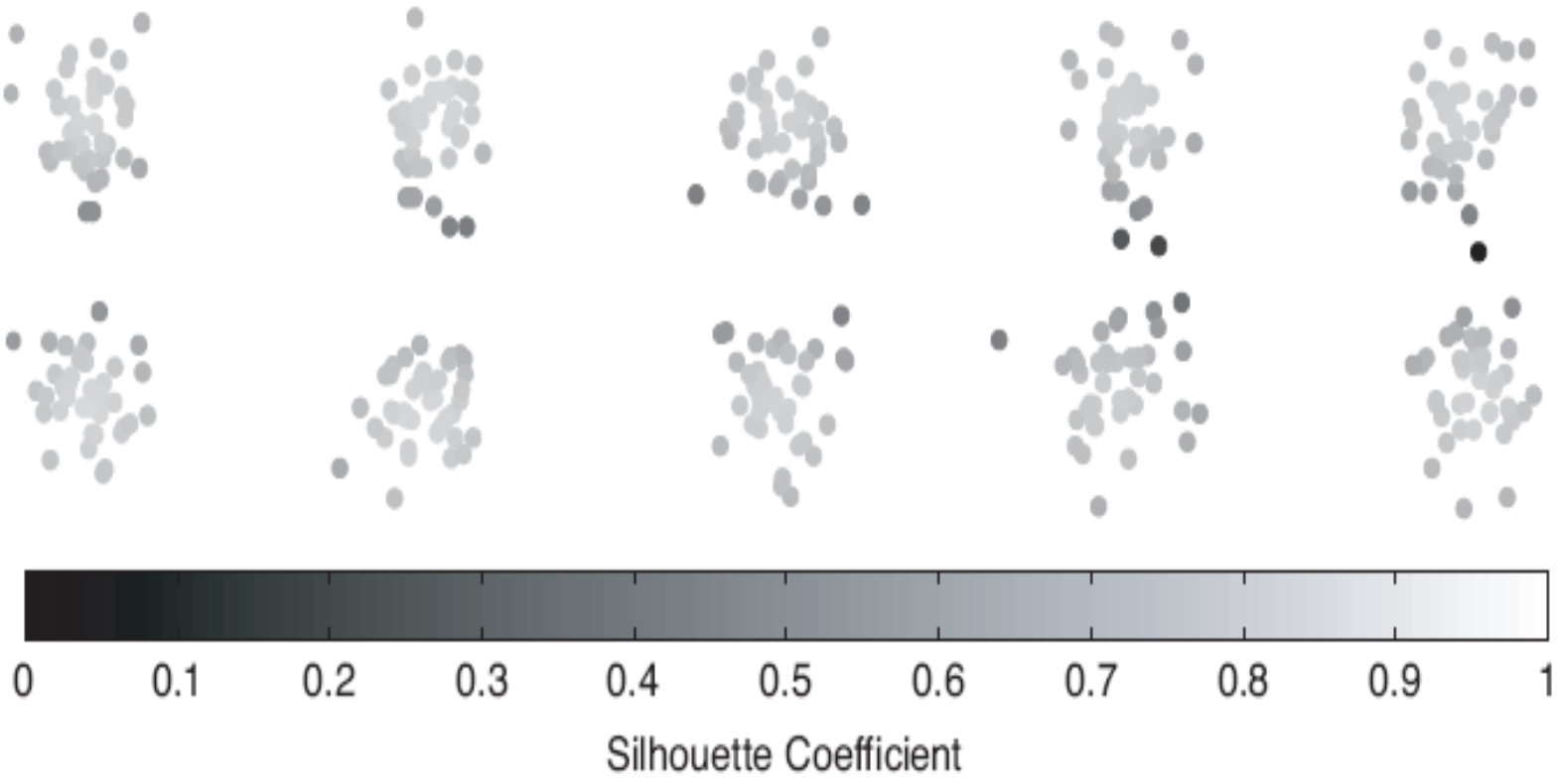
Separation

Internal Measures: Silhouette Coefficient

- Silhouette Coefficient combine cohesion and separation, for *individual points*.
- For an individual point *i*
  - Calculate ***a*** = average distance of *i* to the points in its cluster
  - Calculate ***b*** = MIN (average distance of *i* to points in another cluster)
  - The silhouette coefficient for a point is then given by
    $s = 1 - a/b$ if $a < b$, or
    $s = b/a - 1$ if $a \geq b$ (not the usual case)
  - Typically between 0 and 1, the closer to 1 the better.
- Can calculate Average Silhouette width for a cluster or a clustering

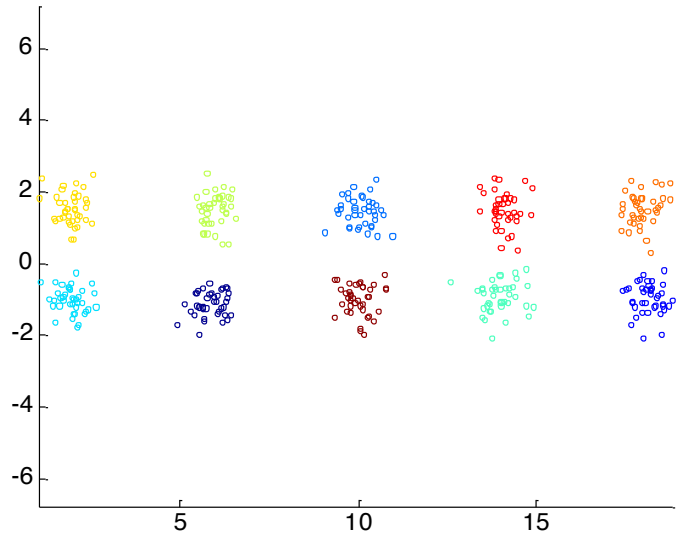


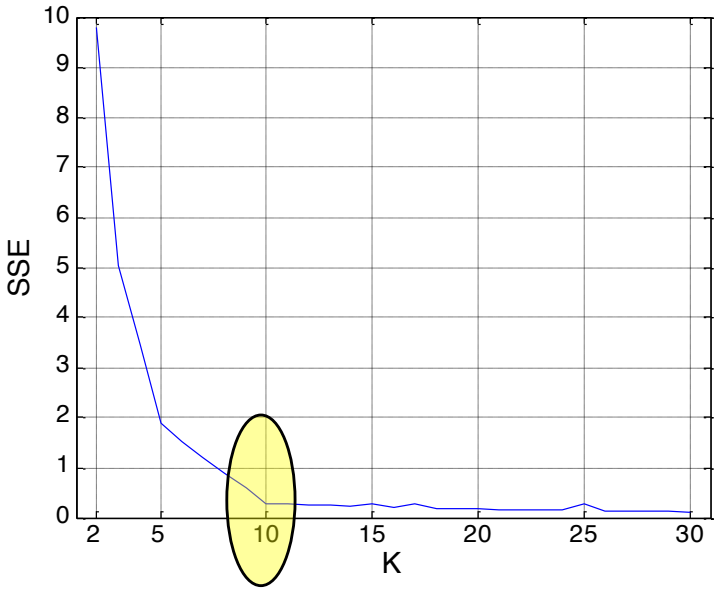
Silhouette Coefficient - Examples

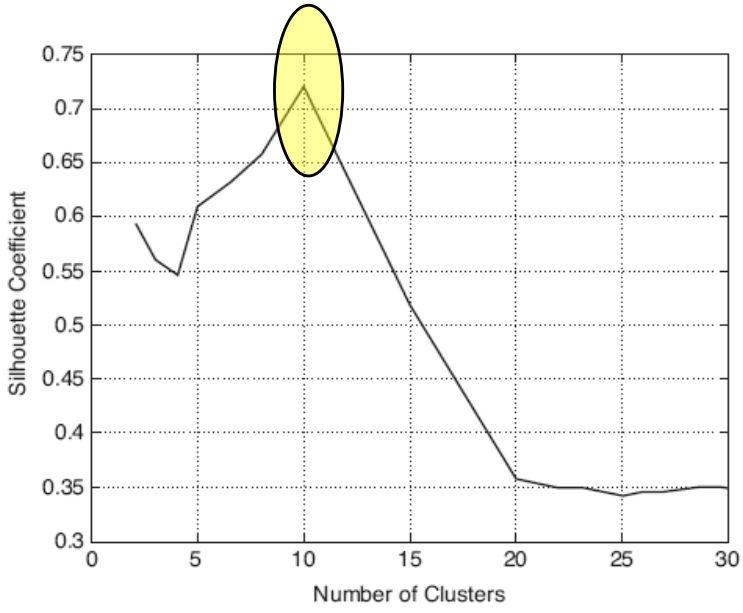


Estimate Number of Clusters

- Clusters in some datasets may not separate well.
- Some measures, e.g., SSE and Average Silhouette coefficient, may be used to estimate # of clusters





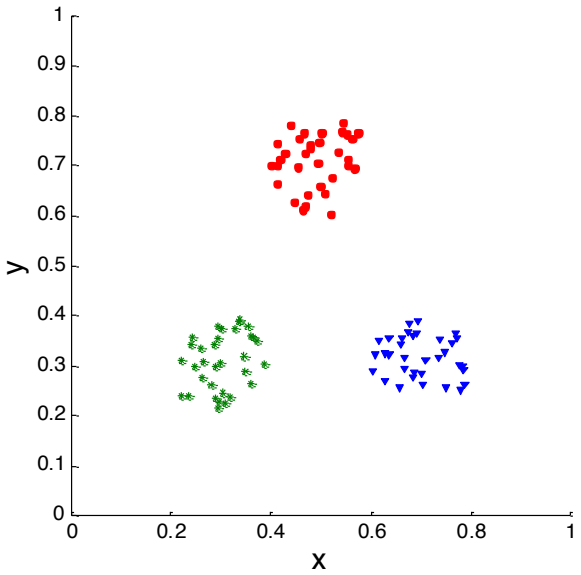


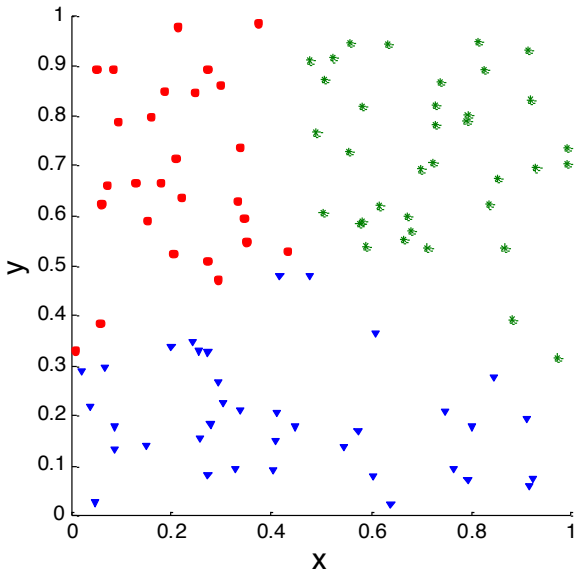
Cluster Validity by Correlation

- Actual Matrix: Proximity (Similarity) Matrix
- Ideal Matrix: "Incidence" Matrix
  - Ideally, all points in the same cluster are the same (i.e., similarity = 1) and every pair of points in different clusters are totally different (i.e., similarity = 0)
- Compute *correlation* between the two matrices
  - Since the matrices are symmetric, only the correlation between n(n-1) / 2 entries needs to be calculated.
- High correlation indicates that points that belong to the same cluster are close to each other.
  - Good for clustering schemes based on proximity
  - Not a good measure for some density or contiguity based clusters.

Measuring Cluster Validity Via Correlation

- Correlation of incidence and proximity matrices for the K-means clusterings of the following two data sets.



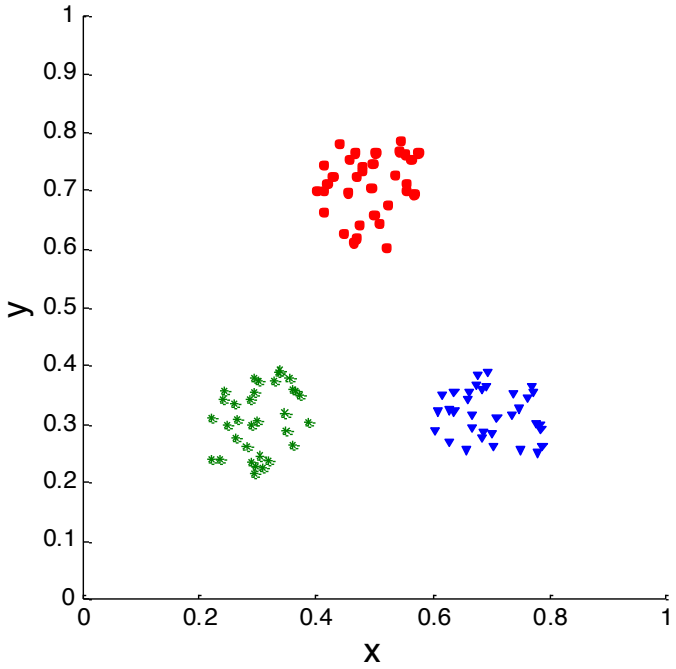


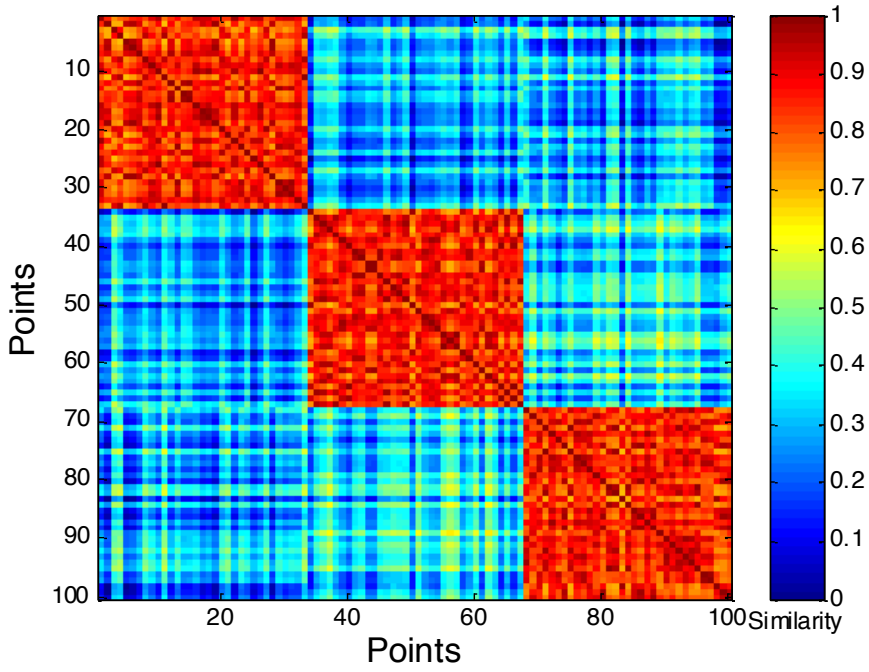
Corr = -0.9235

Corr = -0.5810

Using Similarity Matrix for Cluster Validation

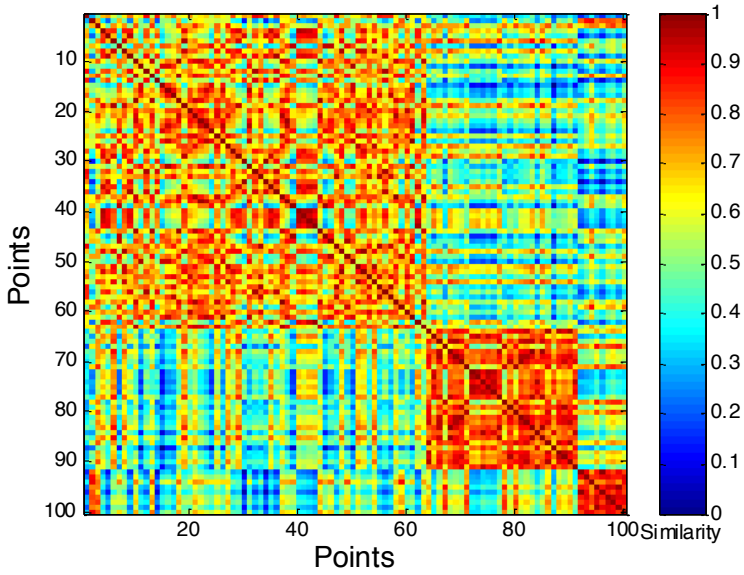
- *Visualization*: Order the similarity matrix with respect to cluster labels and inspect visually.

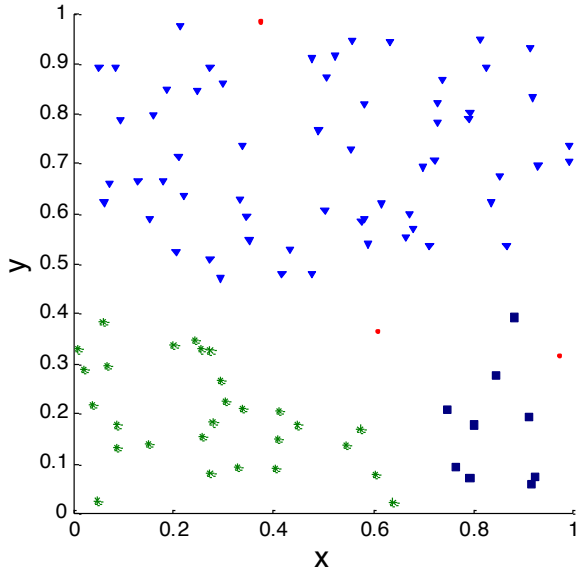




Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

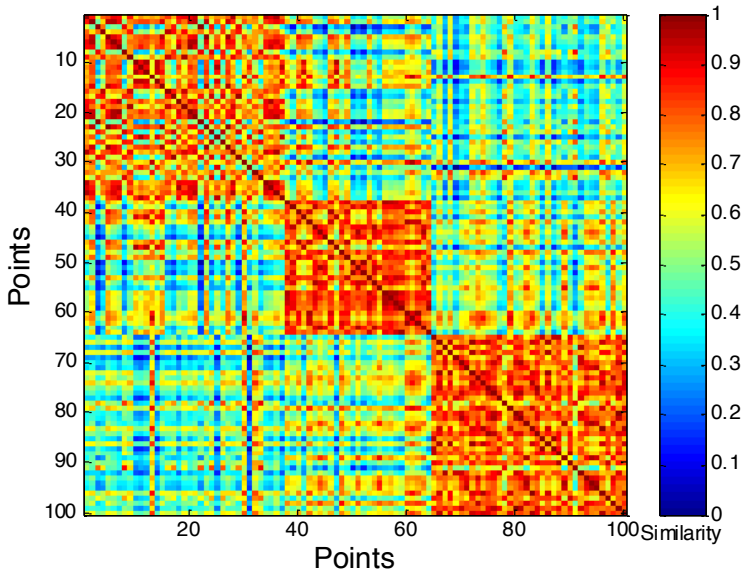


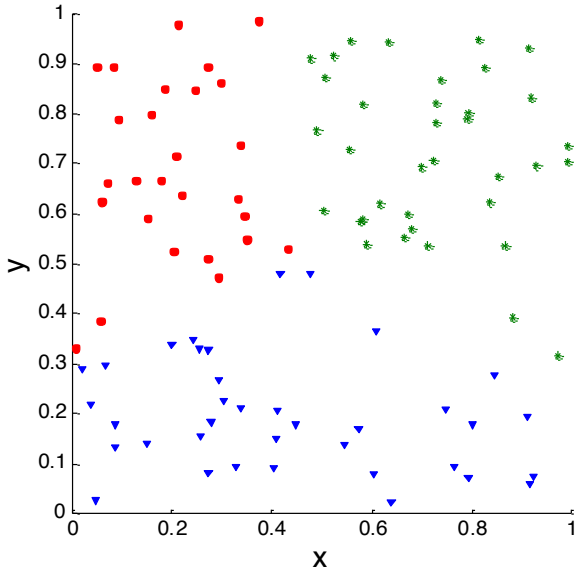


DBSCAN

Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

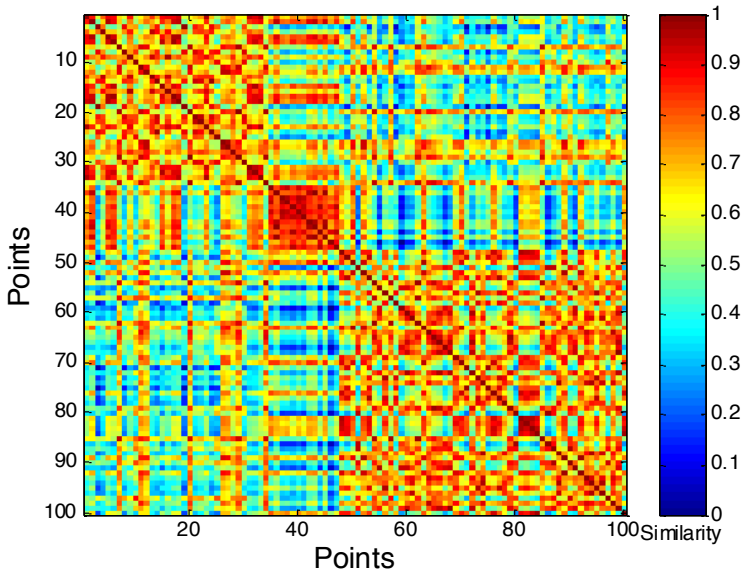


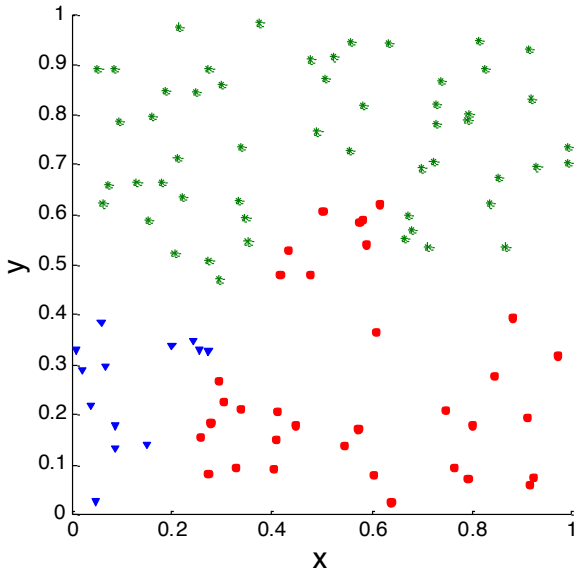


K-means

Using Similarity Matrix for Cluster Validation

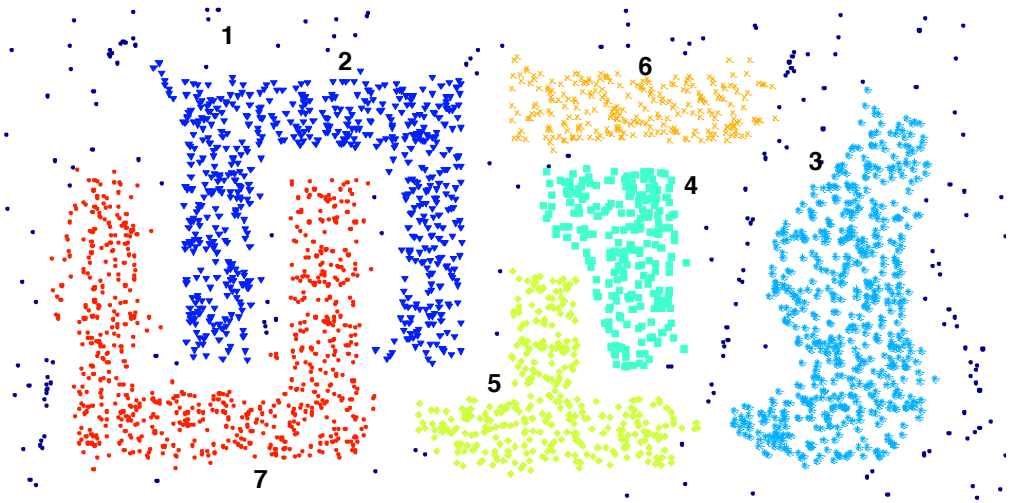
- Clusters in random data are not so crisp

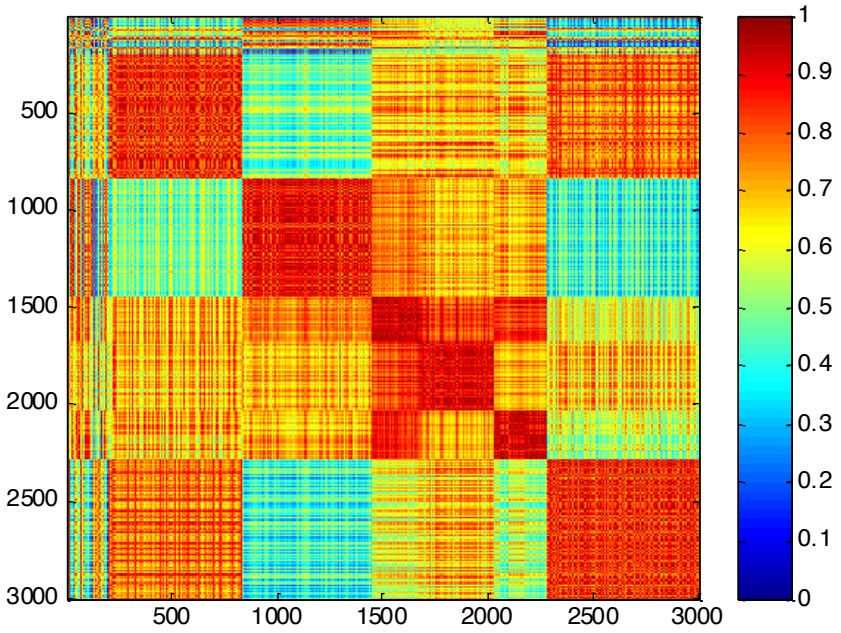




Complete Link (HC-MAX)

Using Similarity Matrix for Cluster Validation





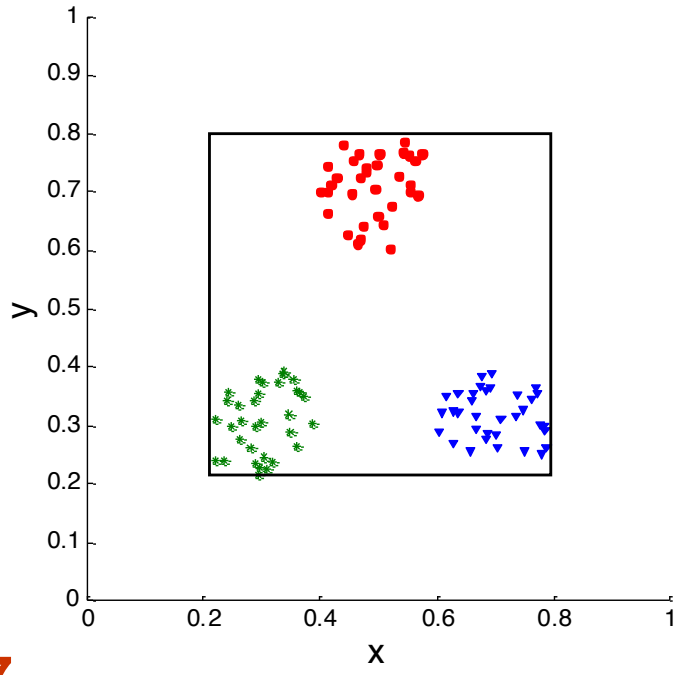
DBSCAN

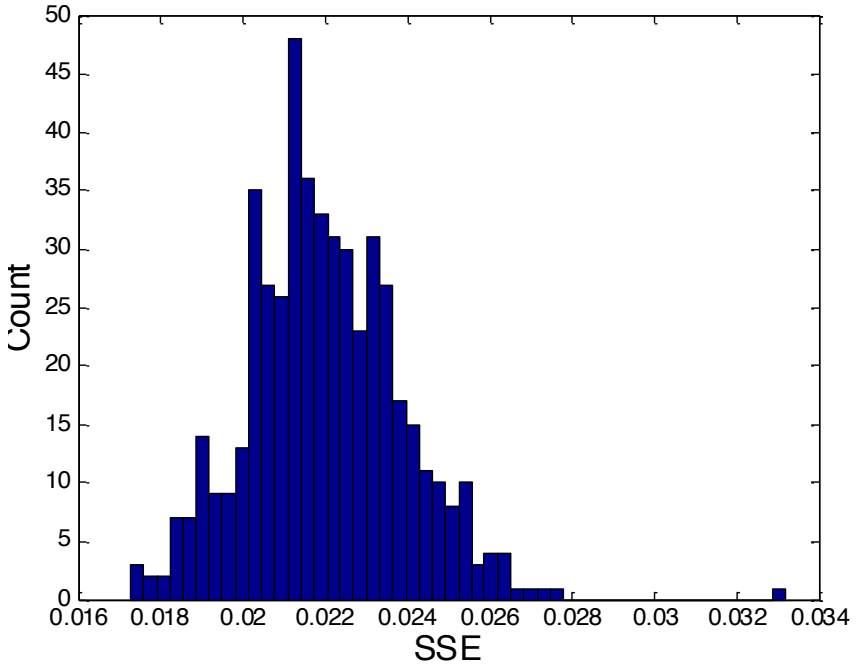
Significance of Cluster Validity Measure

- How to interpret the significance of a measure?
  - Cluster validity measure, e.g., SSE, usually provides a number, e.g., SSE=0 is good!
  - How about SSE=10? Good, fair, or poor?
- Statistics provide a framework for cluster validity
  - Compare the values of a measure from *random data* (or corresponding clustering) to those from a clustering under examination.
  - The more "atypical" a clustering result is, the more likely it represents valid structure in the data
    - If the measure value is unlikely (to be from random data), then the cluster results are valid
  - These approaches are more complicated and harder to understand.

Statistical Framework for SSE

- Given SSE of 0.005 from three clusters by K-Means
- Compare it with SSE's of three clusters in 500 random data sets
- Histogram shows SSE of three clusters in random data points of size 500 distributed over the range 0.2 – 0.8 for x and y values





External Measures of Cluster Validity: Entropy and Purity

Table 5.9. K-means Clustering Results for LA Document Data Set

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Entropy	Purity
1	3	5	40	506	96	27	1.2270	0.7474
2	4	7	280	29	39	2	1.1472	0.7756
3	1	1	1	7	4	671	0.1813	0.9796
4	10	162	3	119	73	2	1.7487	0.4390
5	331	22	5	70	13	23	1.3976	0.7134
6	5	358	12	212	48	13	1.5523	0.5525
Total	354	555	341	943	273	738	1.1450	0.7203

entropy For each cluster, the class distribution of the data is calculated first, i.e., for cluster j we compute p_{ij}, the 'probability' that a member of cluster j belongs to class i as follows: $p_{ij} = m_{ij}/m_j$, where m_j is the number of values in cluster j and m_{ij} is the number of values of class i in cluster j. Then using this class distribution, the entropy of each cluster j is calculated using the standard formula $e_j = \sum_{i=1}^{L} p_{ij} \log_2 p_{ij}$, where the L is the number of classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e., $e = \sum_{i=1}^{K} \frac{m_i}{m} e_j$, where m_j is the size of cluster j, K is the number of clusters, and m is the total number of data points.

purity Using the terminology derived for entropy, the purity of cluster j, is given by $purity_j = \max p_{ij}$ and the overall purity of a clustering by $purity = \sum_{i=1}^{K} \frac{m_i}{m} purity_j$.

Final Comment on Cluster Validity

"The validation of clustering structures is the most difficult and frustrating part of cluster analysis.

Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage."

Algorithms for Clustering Data, Jain and Dubes