

Data Mining 2025

Clustering Analysis and Unsupervised Learning

Dept. of Computer Science and Information Engineering

National Cheng Kung University

Kun-Ta Chuang

ktchuang@mail.ncku.edu.tw



What is Cluster Analysis?

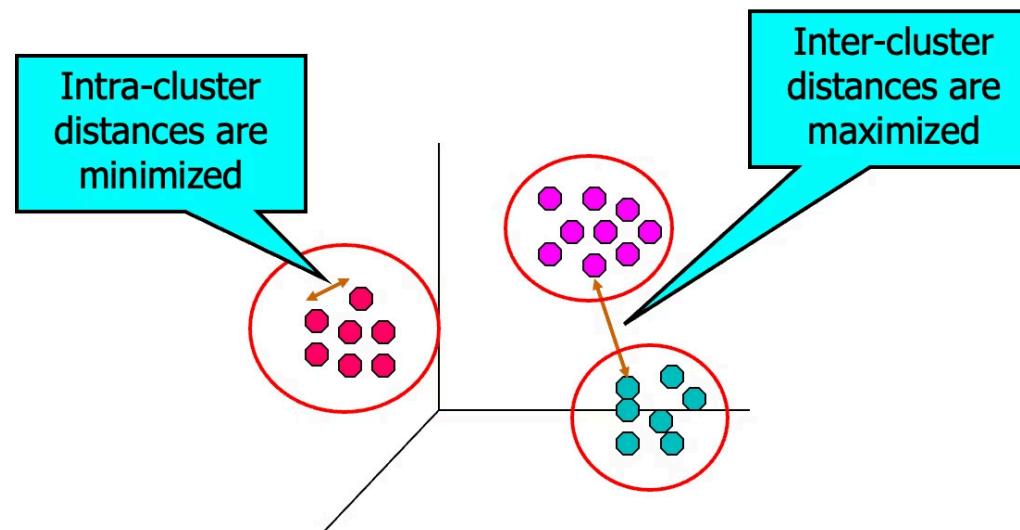
Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups

Intra-cluster distances are minimized

Objects within the same cluster are close to each other

Inter-cluster distances are maximized

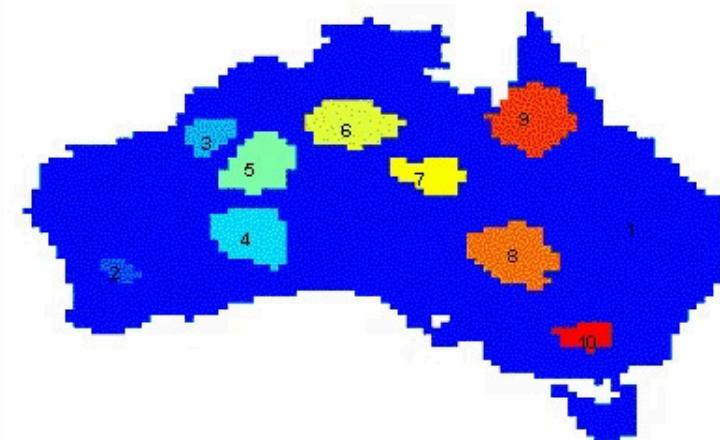
Objects in different clusters are far from each other



Applications of Cluster Analysis

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

	<i>Discovered Clusters</i>	<i>Industry Group</i>
1	Applied-Matt-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



What is not Cluster Analysis?



Supervised classification

Have class label information



Simple segmentation

Dividing students into different registration groups
alphabetically, by last name



Results of a query

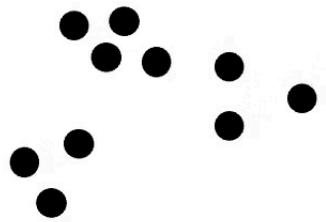
Groupings are a result of an external specification



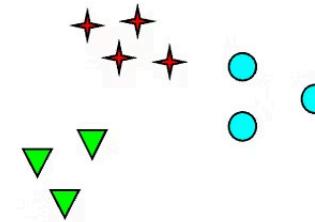
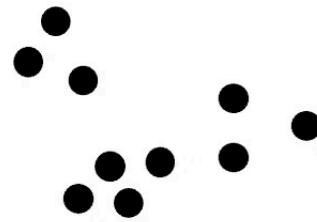
Graph partitioning

Some mutual relevance and synergy, but areas are not identical

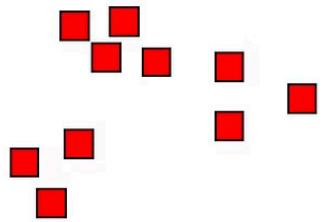
Notion of a Cluster can be Ambiguous



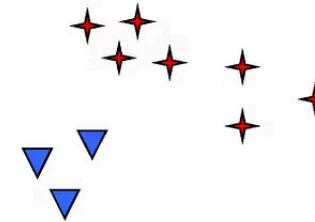
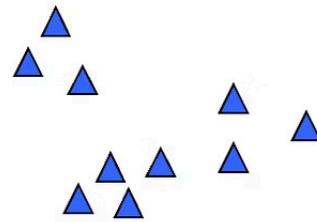
How many clusters?



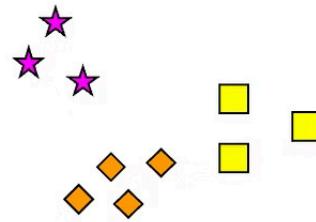
Six Clusters



Two Clusters



Four Clusters



Types of Clusterings



A **clustering** is a set of clusters



Important distinction between **hierarchical** and **partitional** sets of clusters



Partitional Clustering

A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset



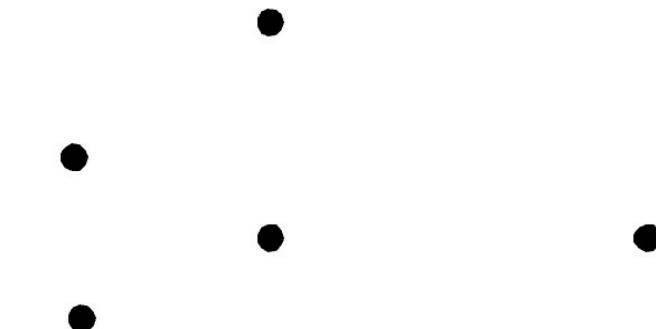
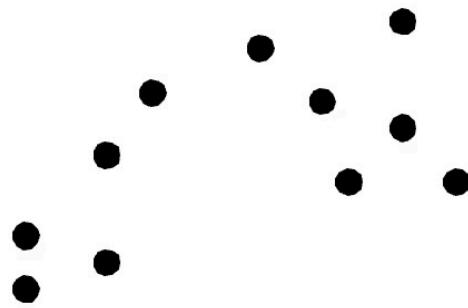
Hierarchical clustering

A set of nested clusters organized as a hierarchical tree

Partitional Clustering

Original Points

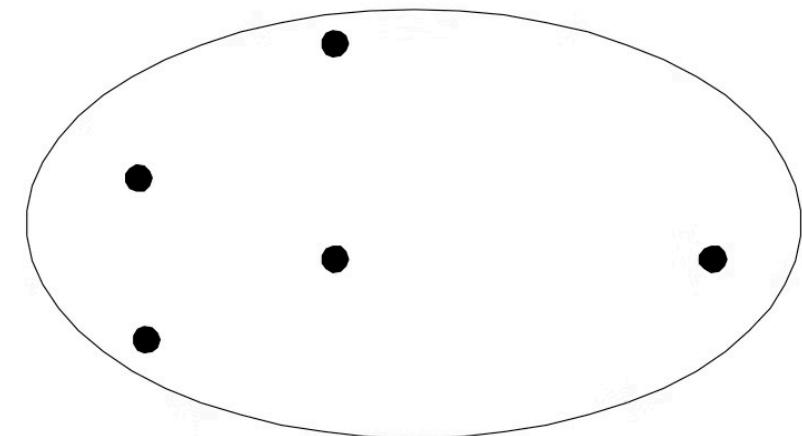
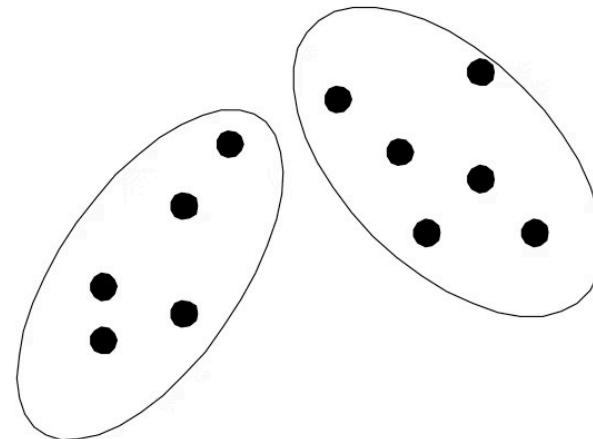
A collection of data points without any grouping



Original Points

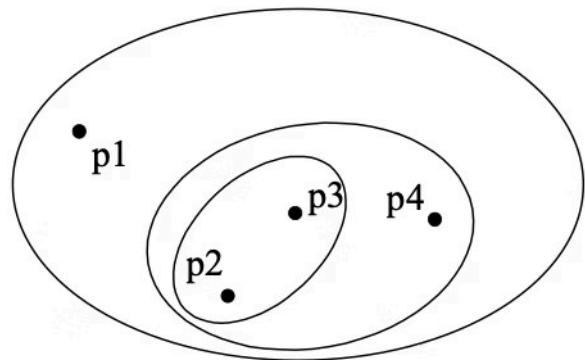
A Partitional Clustering

Data points grouped into distinct clusters based on similarity

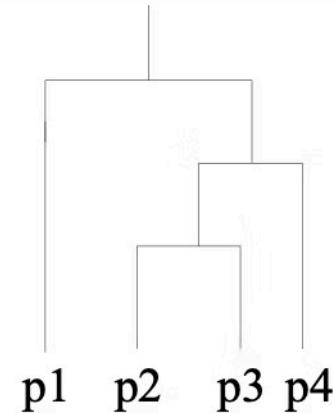


A Partitional Clustering

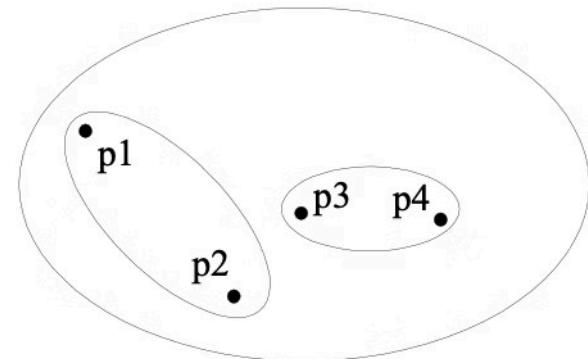
Hierarchical Clustering



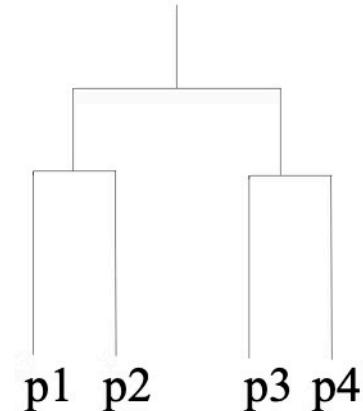
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

Other Distinctions Between Sets of Clusters



Exclusive versus non-exclusive

- In non-exclusive clusterings, points may belong to multiple clusters.
- Can represent multiple classes or '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

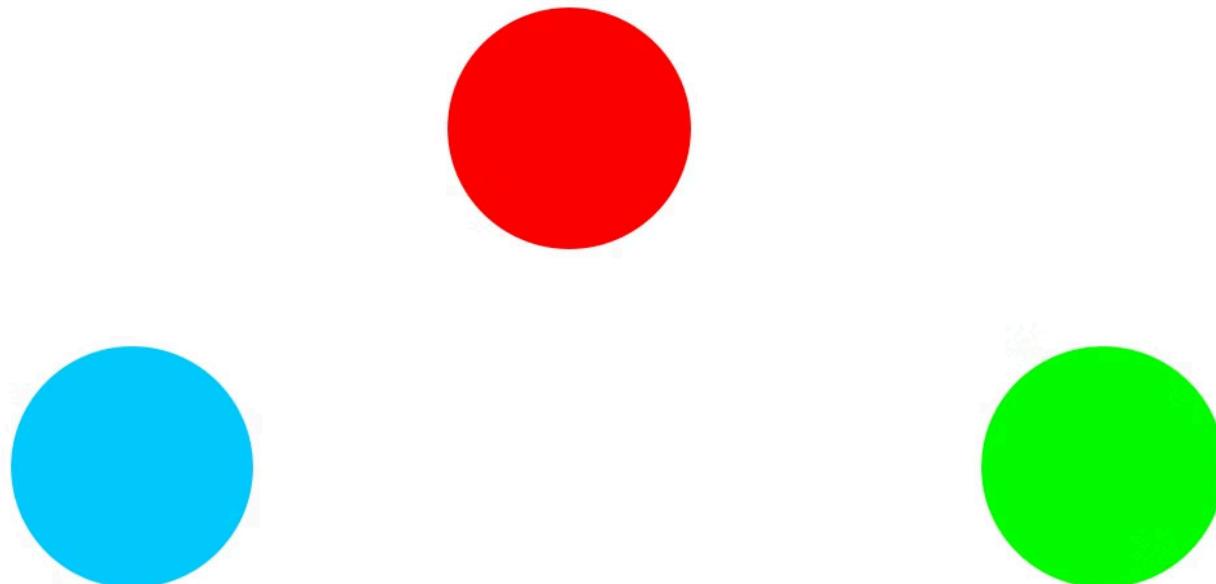
Types of Clusters

- 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:

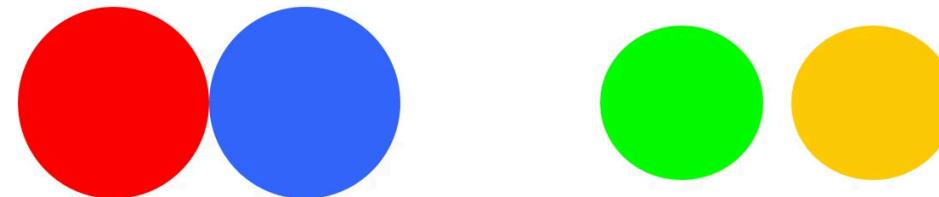
A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.



3 well-separated clusters

Types of Clusters: Center-Based

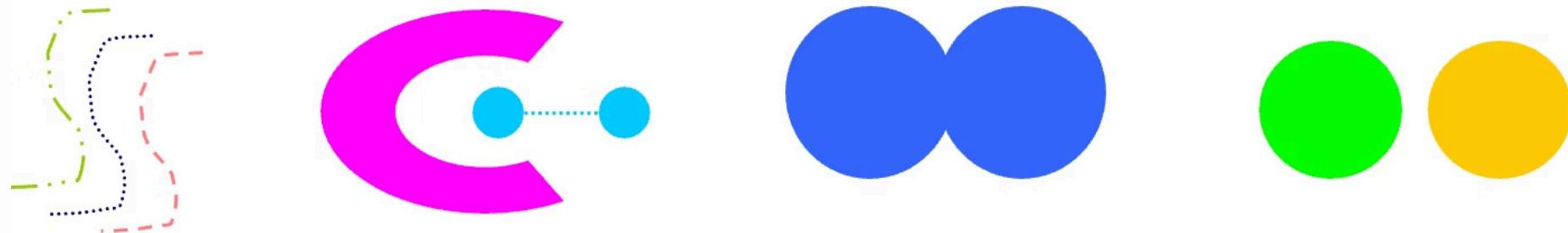
- Center-based
 - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the "center" of a cluster, than to the center of any other cluster
 - The center of a cluster is often a **centroid**, the average of all the points in the cluster, or a **medoid**, the most "representative"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



4 center-based clusters

Types of Clusters: Contiguity-Based

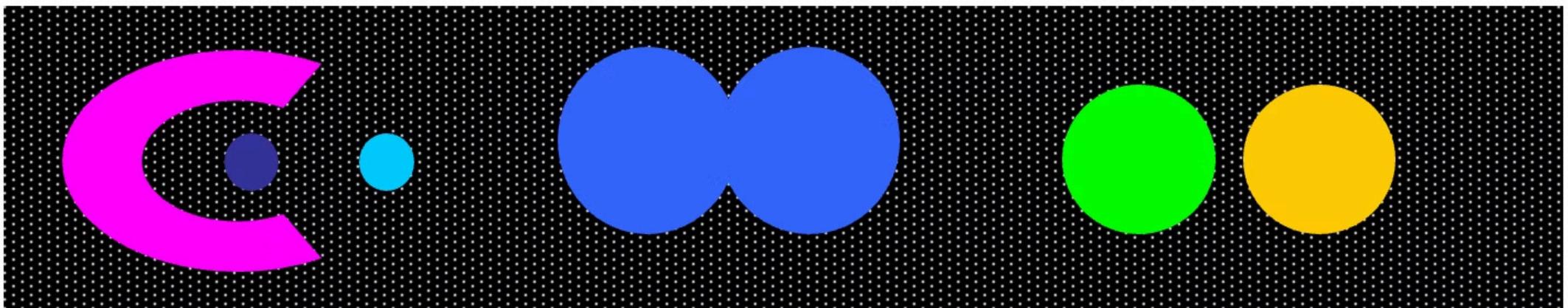
- Contiguous Cluster (Nearest neighbor or Transitive)
- A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.



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.

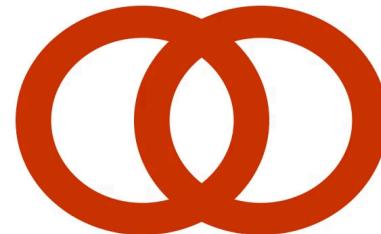


6 density-based clusters

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 or maximize an objective function.
- Enumerate all possible ways of dividing the points into clusters and evaluate the 'goodness' of each potential set of clusters by 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 of the global objective function approach 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 Are Important

- 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

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

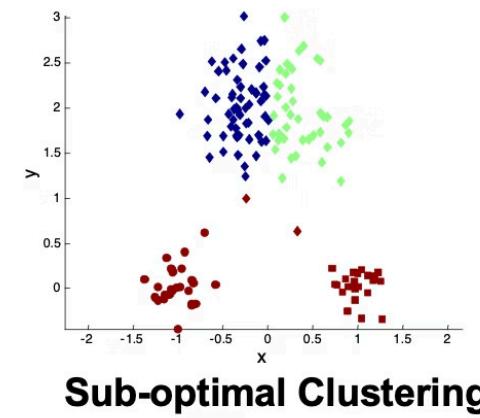
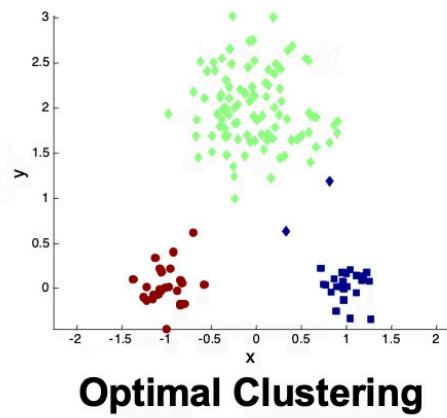
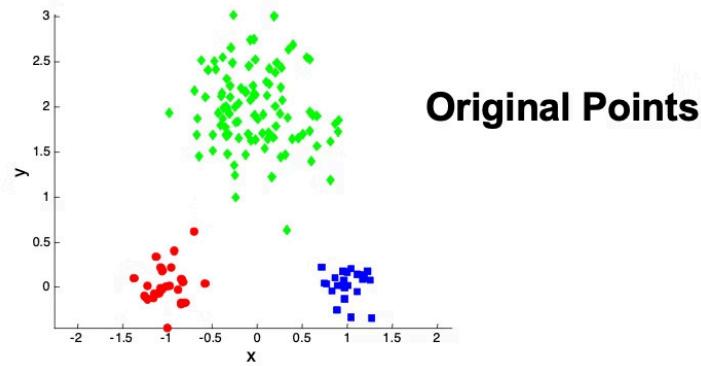
Algorithm 1 Basic K-means Algorithm.

- 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

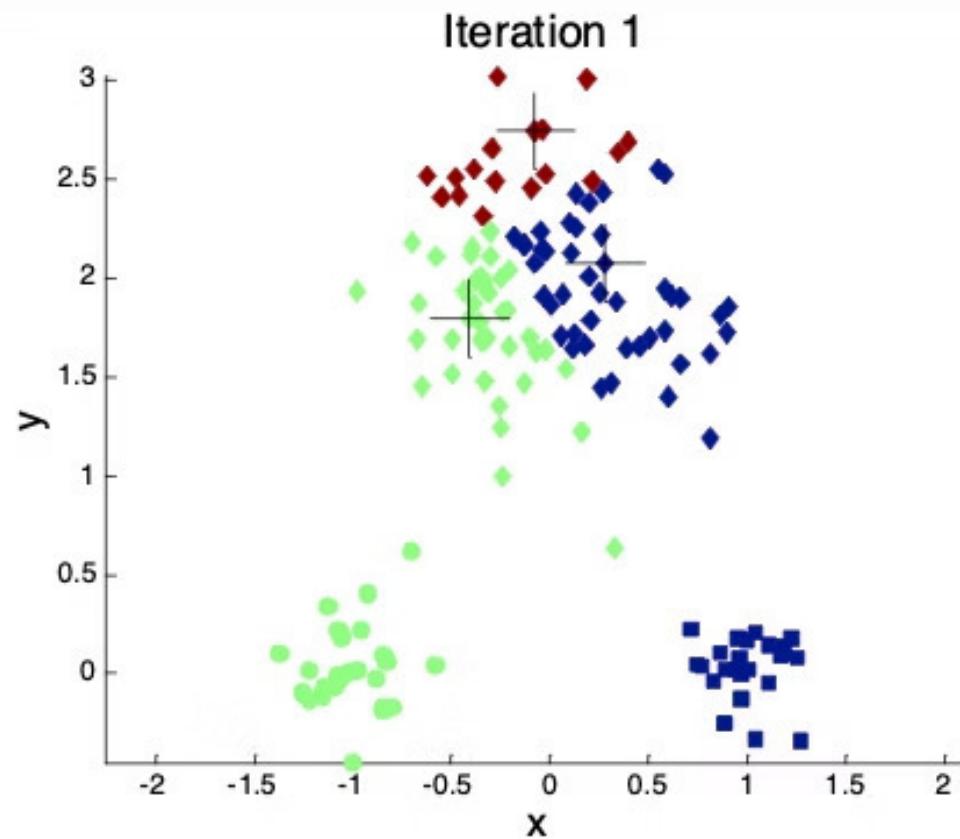
K-means Clustering – Details

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

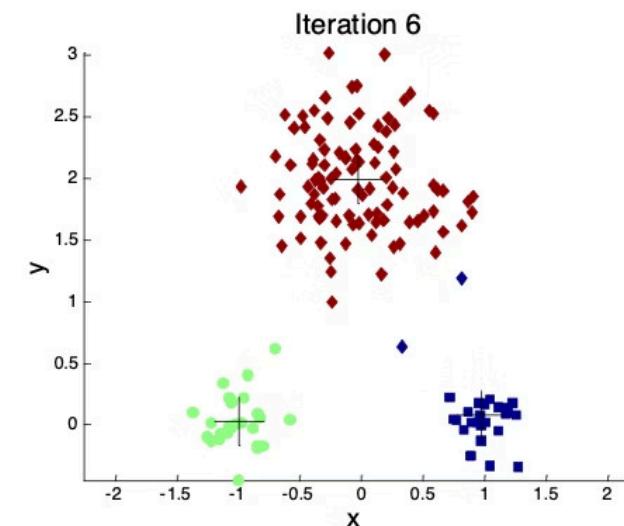
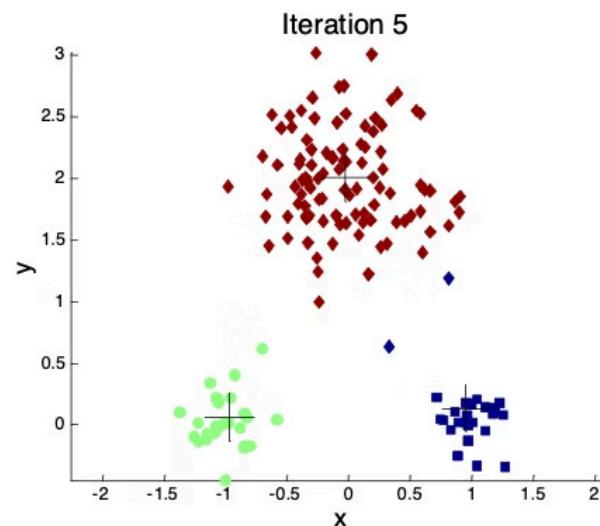
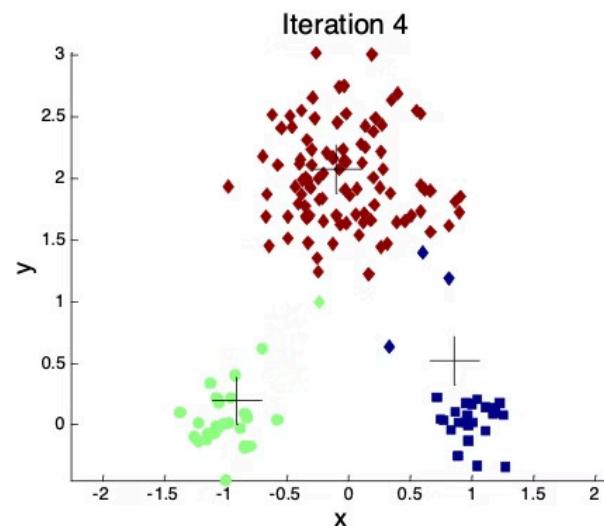
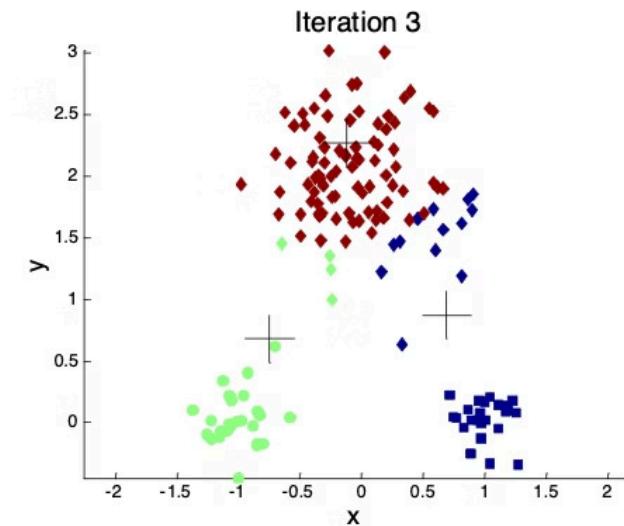
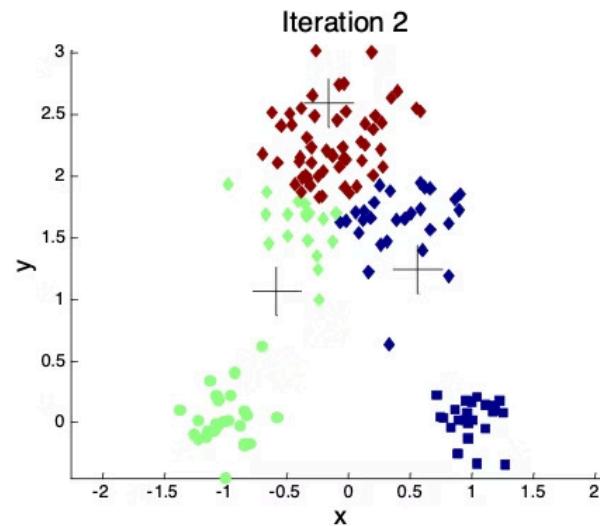
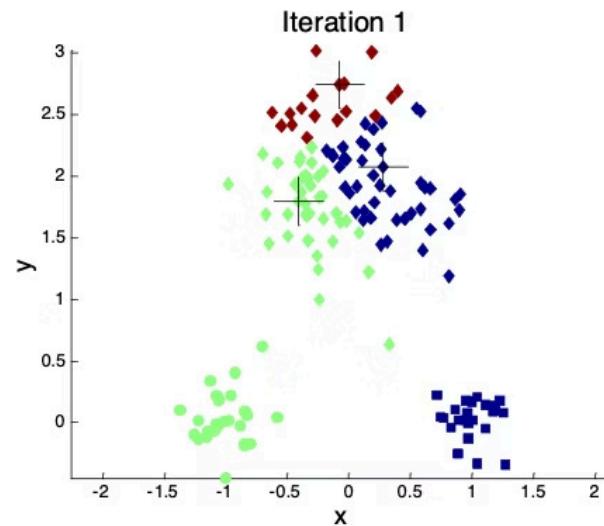
Two different K-means Clusterings



Importance of Choosing Initial Centroids



Importance of Choosing Initial Centroids



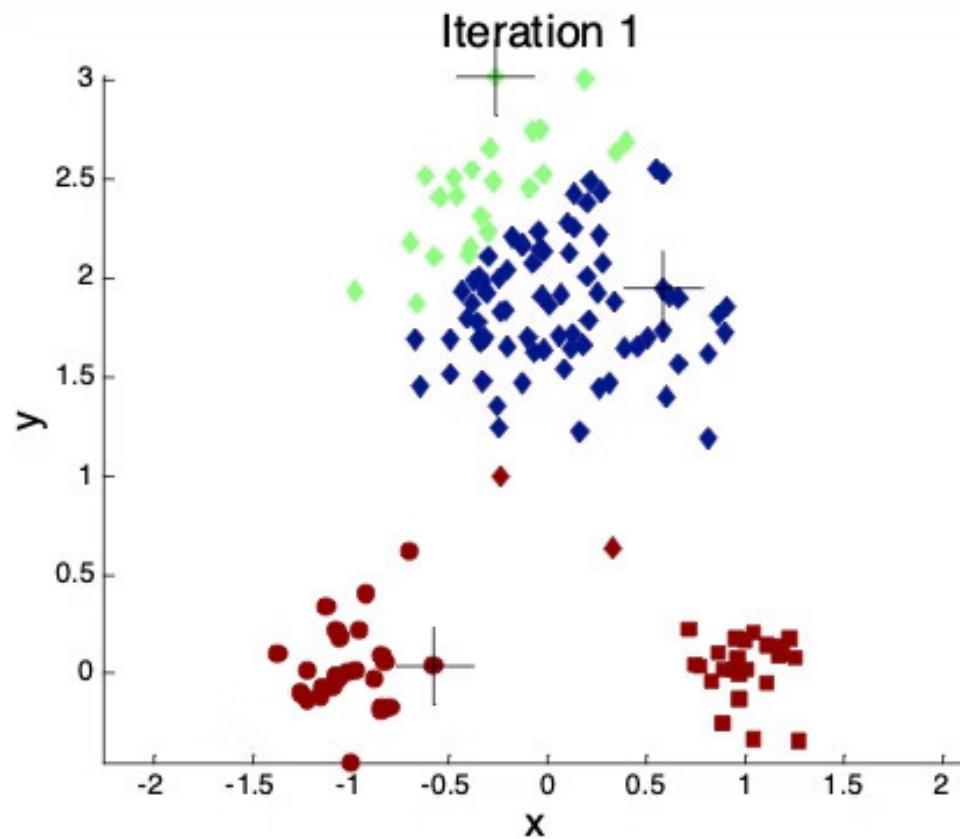
Evaluating K-means Clusters

- Most common measure is Sum of Squared Error (SSE)
 - For each point, the error is the distance to the nearest cluster
 - To get SSE, we square these errors and sum them.

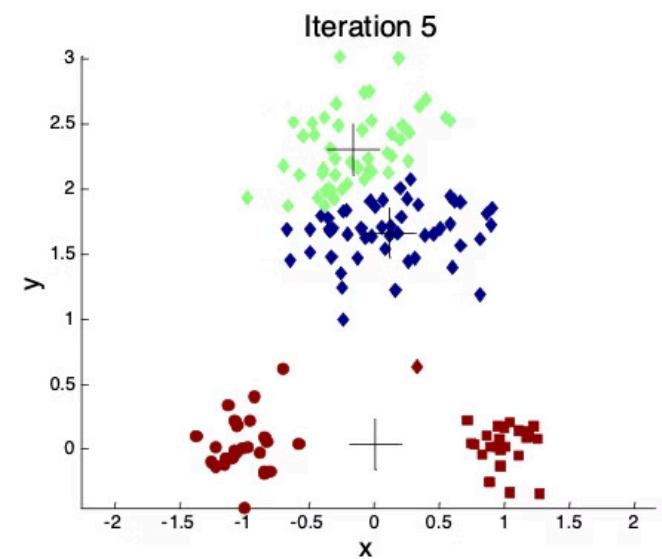
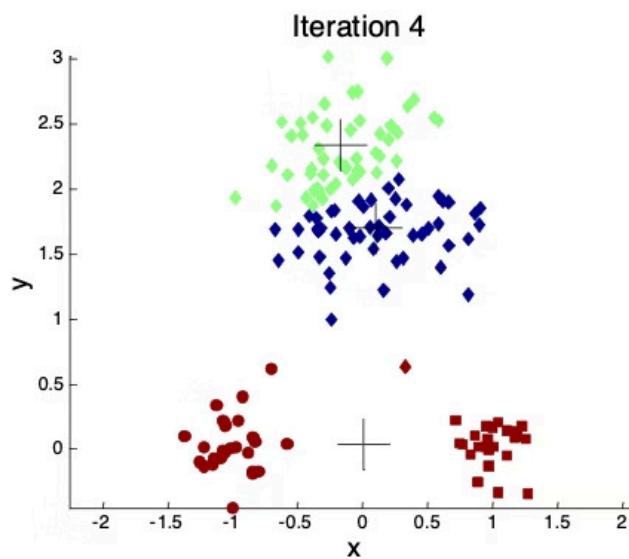
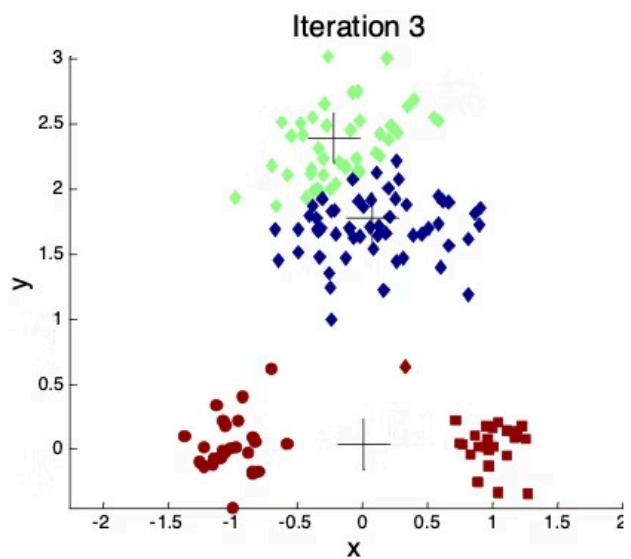
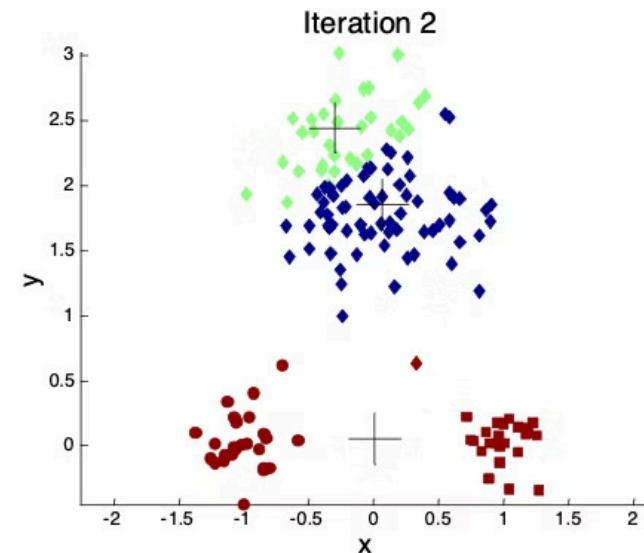
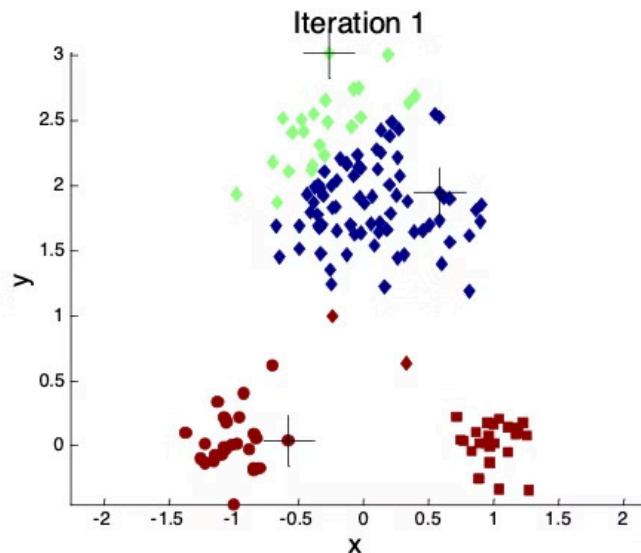
$$SSE = \sum_{i=1}^K \sum_{x \in C_i} dist^2(m_i, x)$$

- x is a data point in cluster C_i and m_i is the representative point for cluster C_i
 - can show that m_i corresponds to the center (mean) of the cluster
- Given two clusters, we can choose the one with the smallest error
- One easy way to reduce SSE is to increase K , the number of clusters
 - A good clustering with smaller K can have a lower SSE than a poor clustering with higher K

Importance of Choosing Initial Centroids...



Importance of Choosing Initial Centroids ...



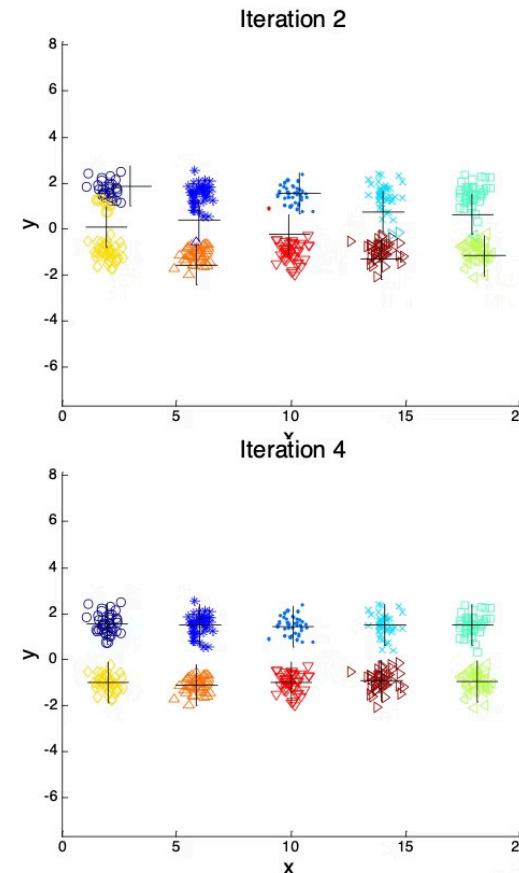
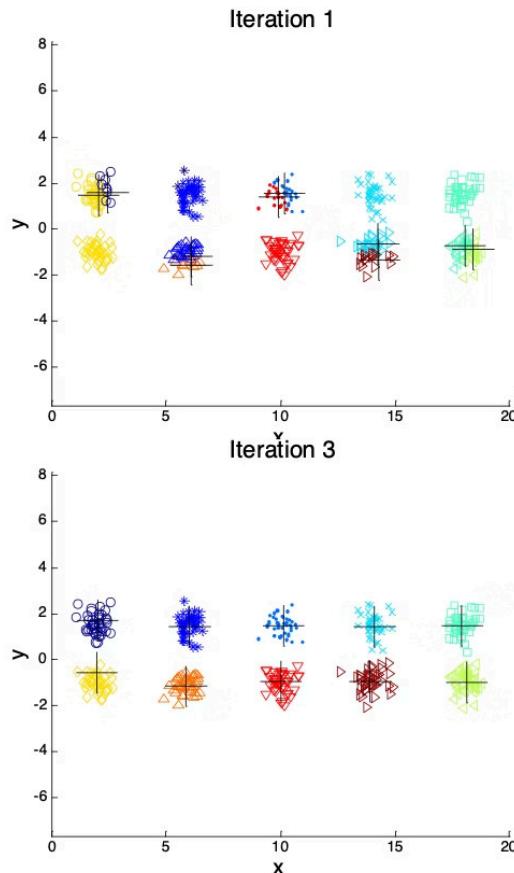
Problems with Selecting Initial Points

- 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 readjust themselves in 'right' way, and sometimes they don't
- Consider an example of five pairs of clusters

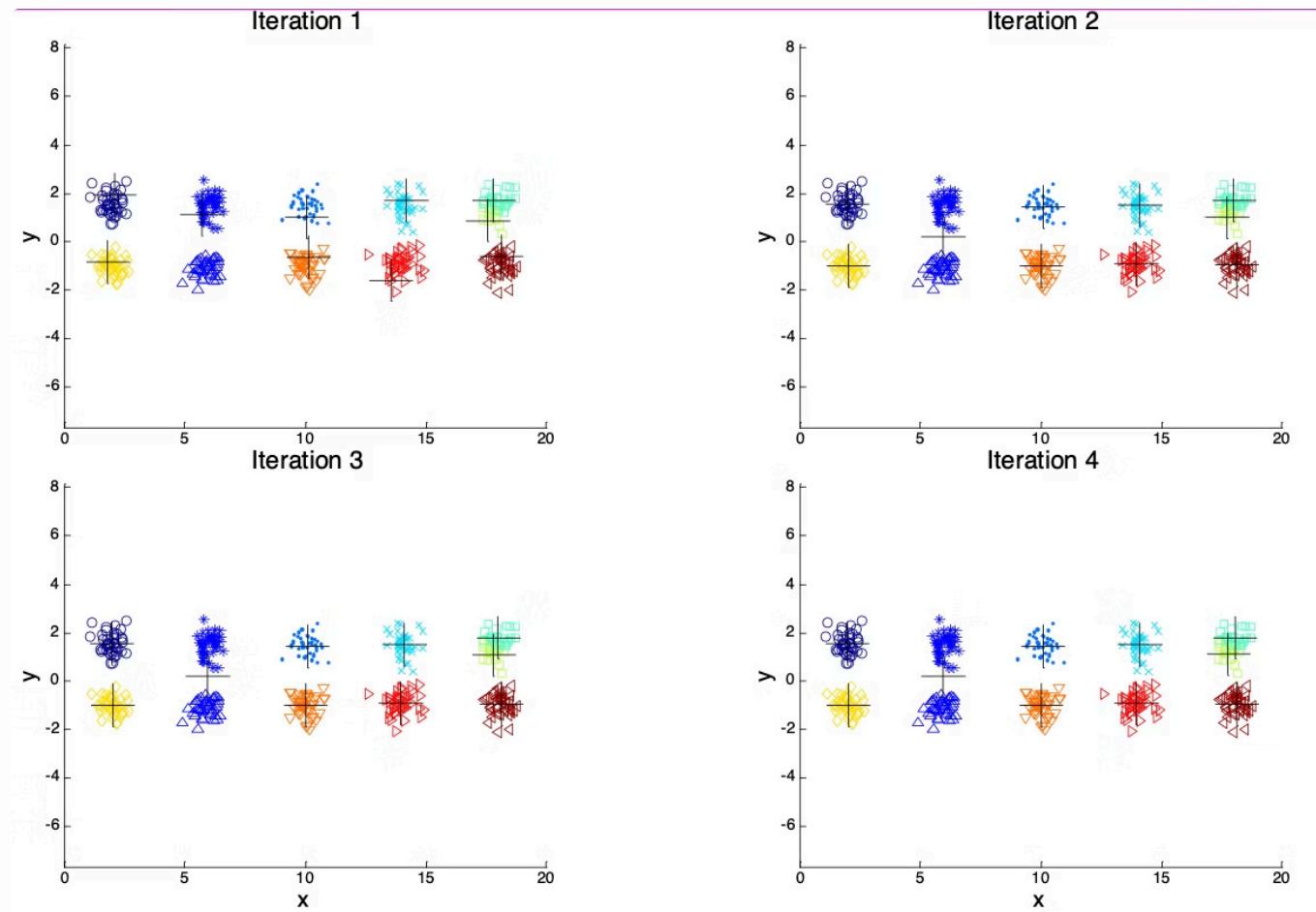
10 Clusters Example



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

10 Clusters Example

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



Solutions to Initial Centroids Problem

- Multiple runs
 - Helps, but probability is not on your side
- Sample and use hierarchical clustering to determine initial centroids
- 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
- Several strategies
 - Choose the point that contributes most to SSE
 - Choose a point from the cluster with the highest SSE
 - If there are several empty clusters, the above can be repeated several times.

Updating Centers Incrementally

- In the basic K-means algorithm, centroids are updated after all points are assigned to a centroid
- An alternative is to update the centroids after each assignment (incremental approach)
 - Each assignment updates zero or two centroids
 - More expensive
 - Introduces an order dependency
 - Never get an empty cluster
 - Can use "weights" to change the impact

Pre-processing and Post-processing



Pre-processing

- Normalize the data
- Eliminate outliers



Post-processing

- Eliminate small clusters that may represent outliers
- Split 'loose' clusters, i.e., clusters with relatively high SSE
- Merge clusters that are 'close' and that have relatively low SSE
- Can use these steps during the clustering process
 - ISODATA

Bisecting K-means

- Bisecting K-means algorithm
 - Variant of K-means that can produce a partitional or a hierarchical clustering

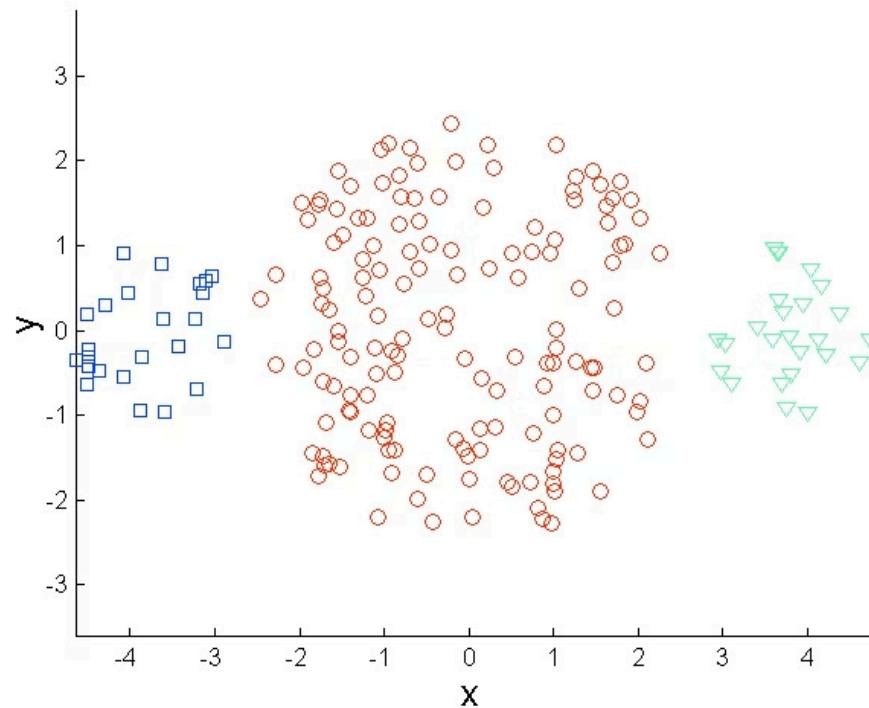
Algorithm 3 Bisecting K-means Algorithm.

- 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

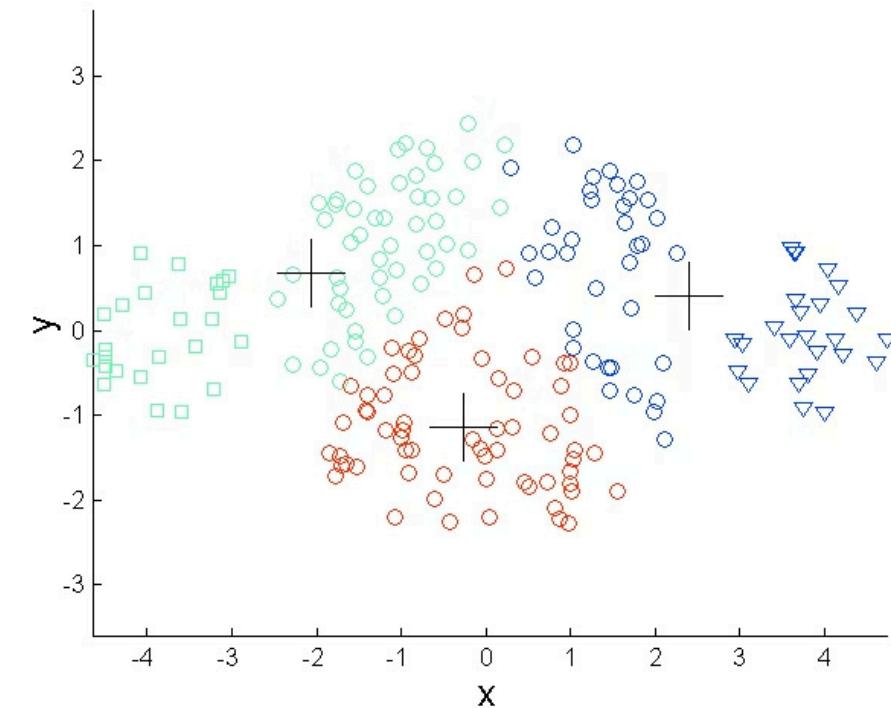
Limitations of K-means

- K-means has problems when clusters are of differing
 - Sizes
 - Densities
 - Non-globular shapes
- K-means has problems when the data contains outliers.

Limitations of K-means: Differing Sizes

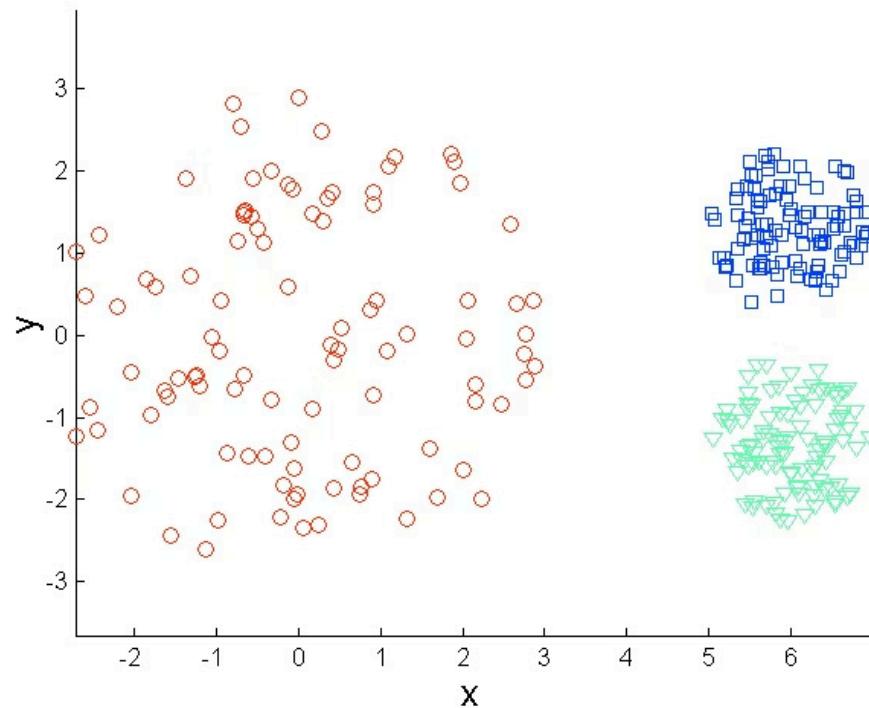


Original Points

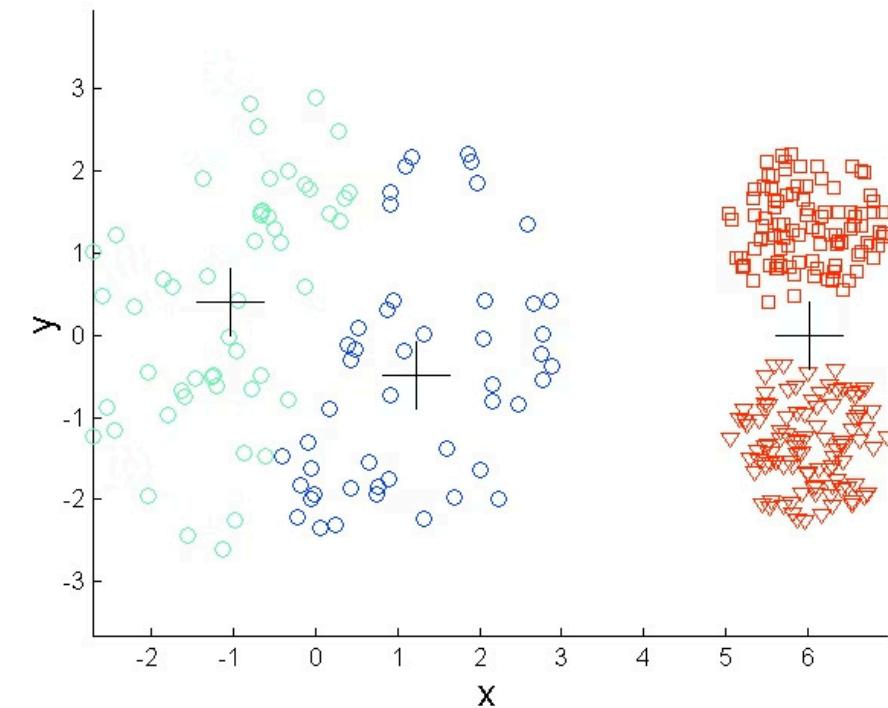


K-means (3 Clusters)

Limitations of K-means: Differing Density

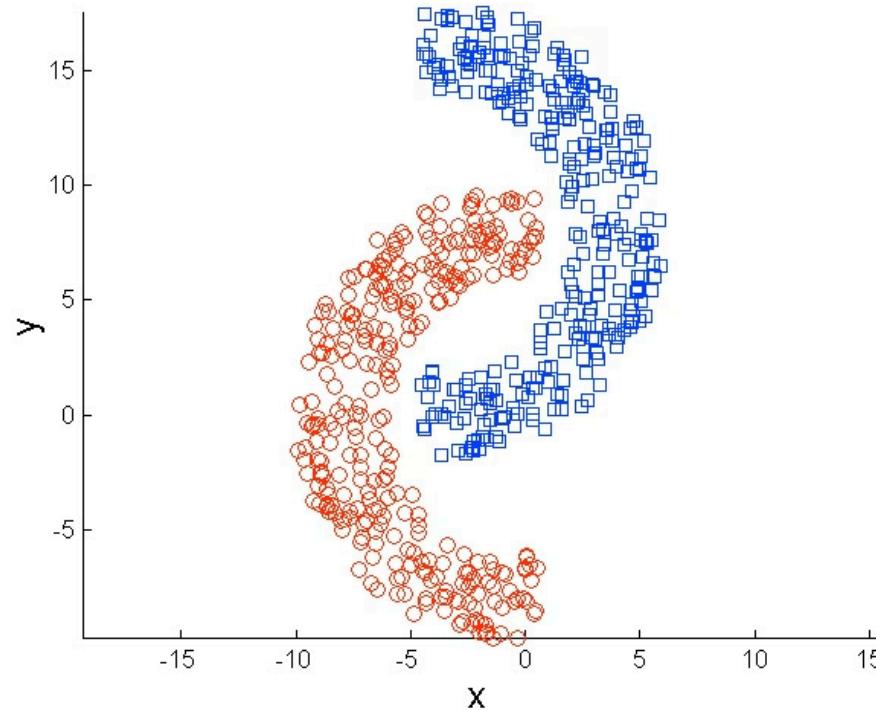


Original Points

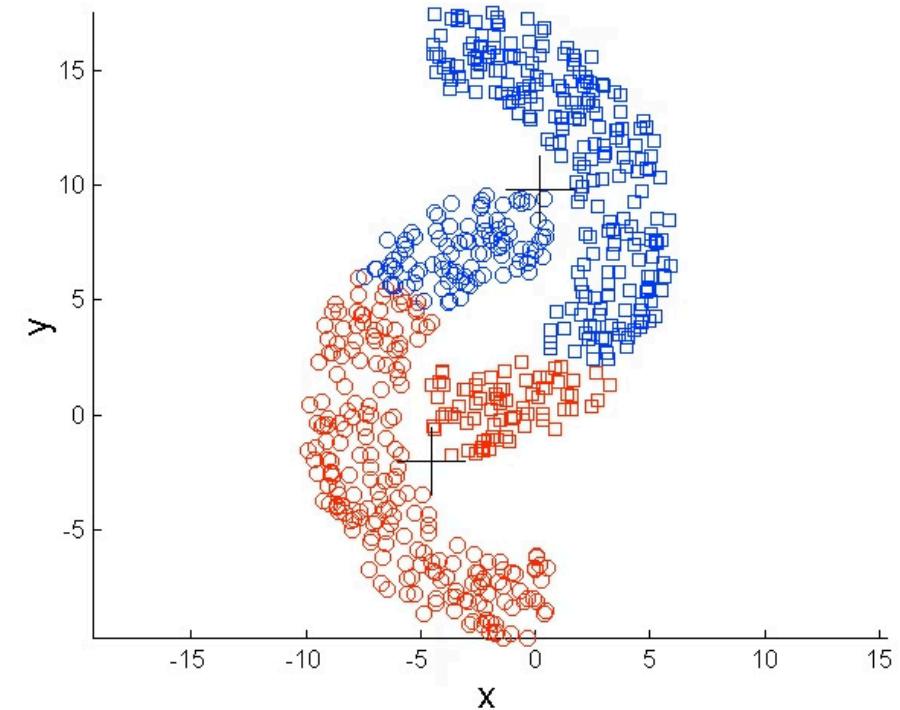


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes

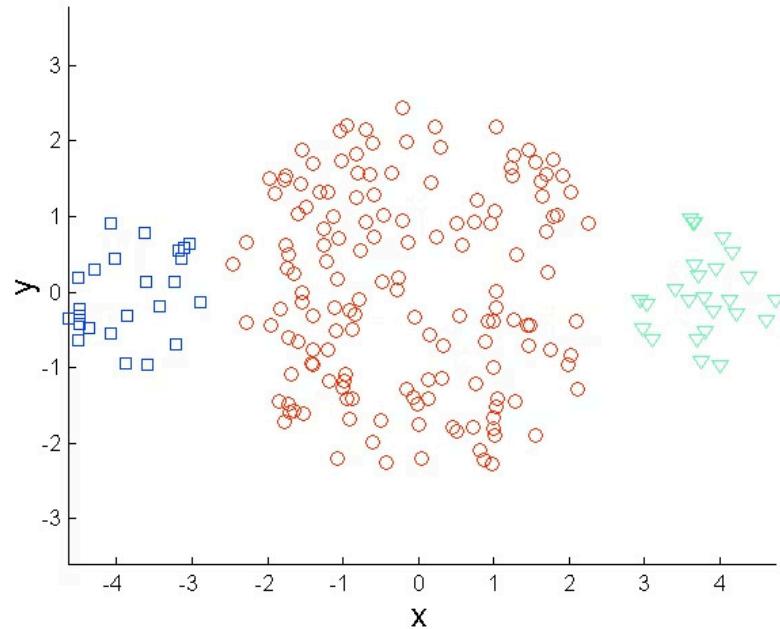


Original Points

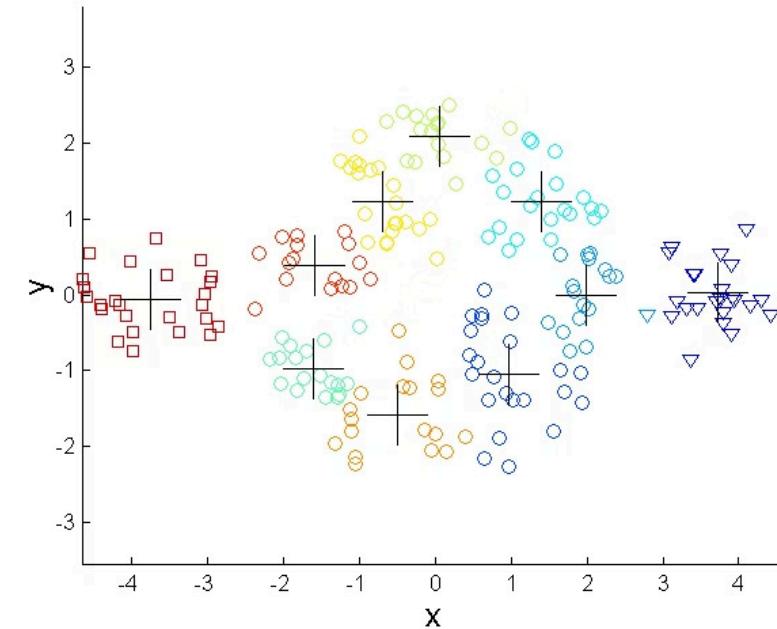


K-means (2 Clusters)

Overcoming K-means Limitations



Original Points

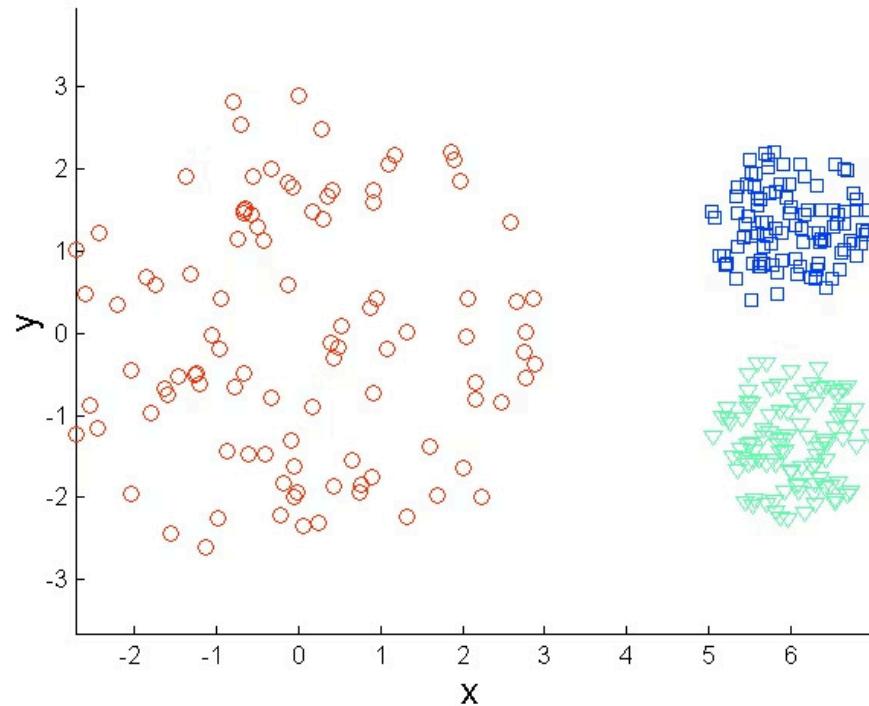


K-means Clusters

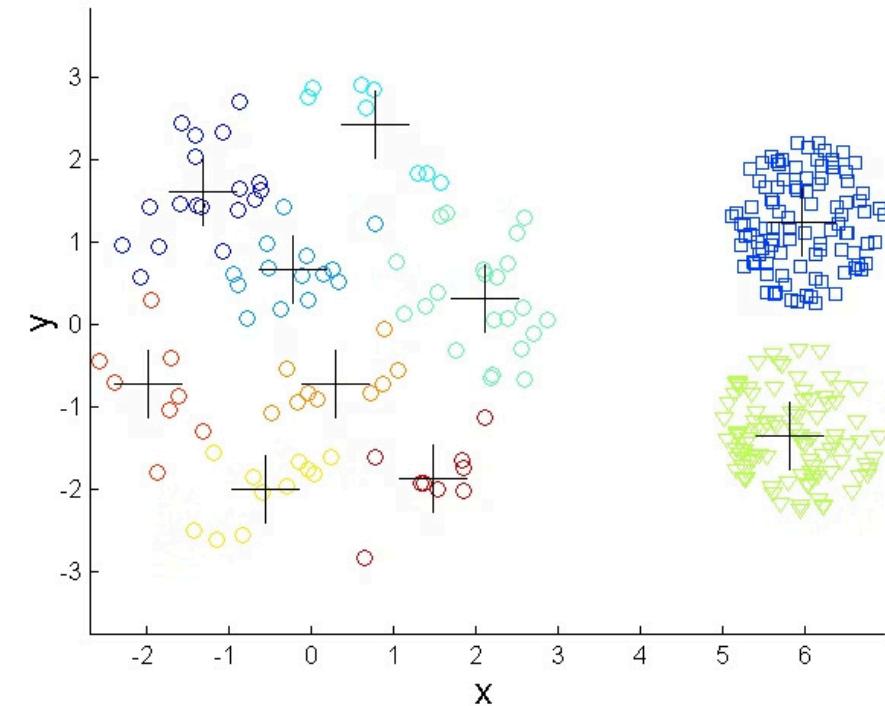
One solution is to use many clusters.

Find parts of clusters, but need to put together.

Overcoming K-means Limitations

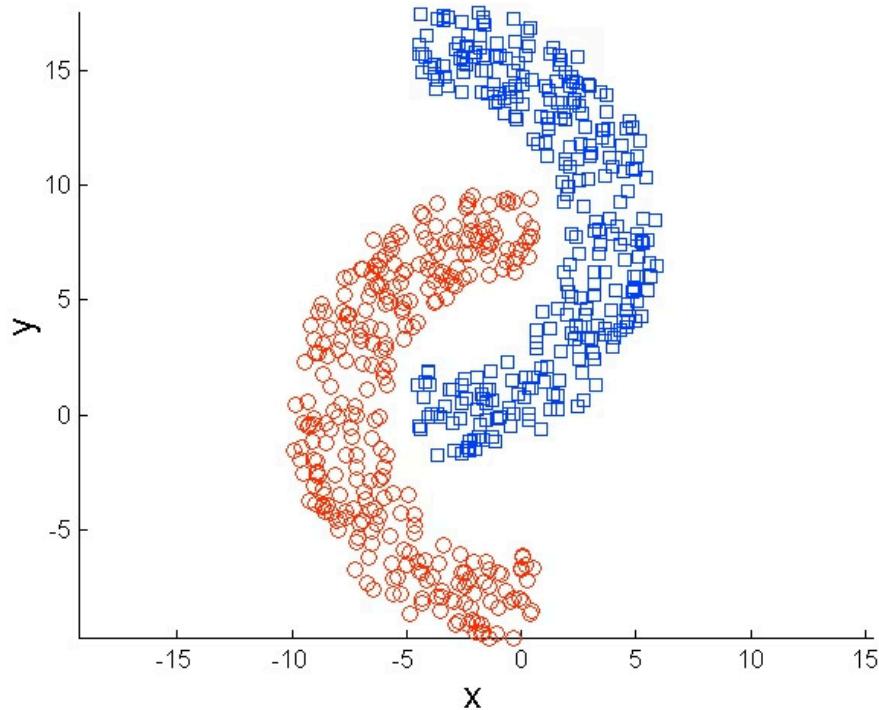


Original Points

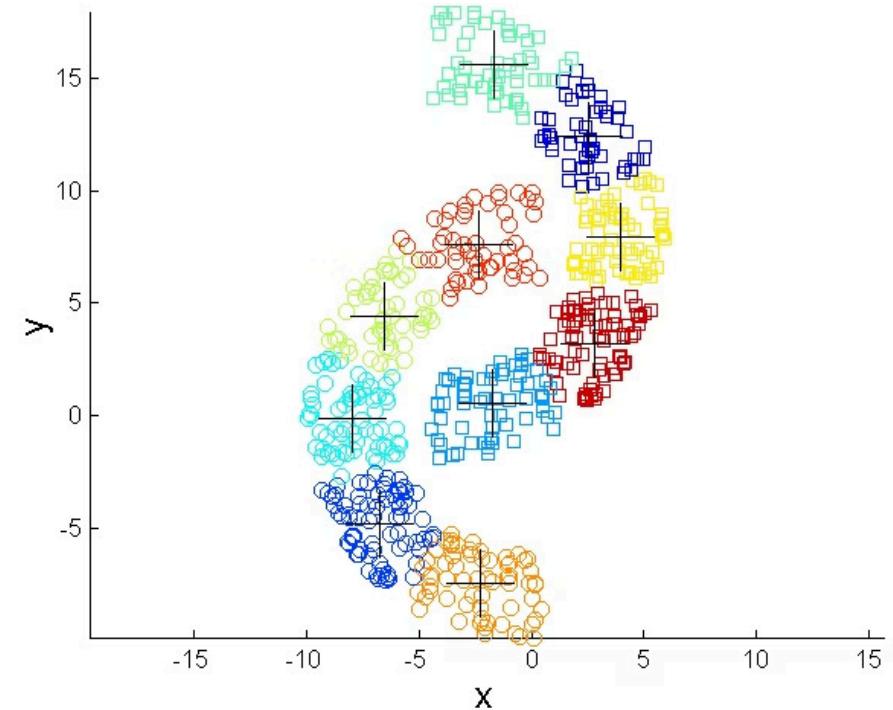


K-means Clusters

Overcoming K-means Limitations



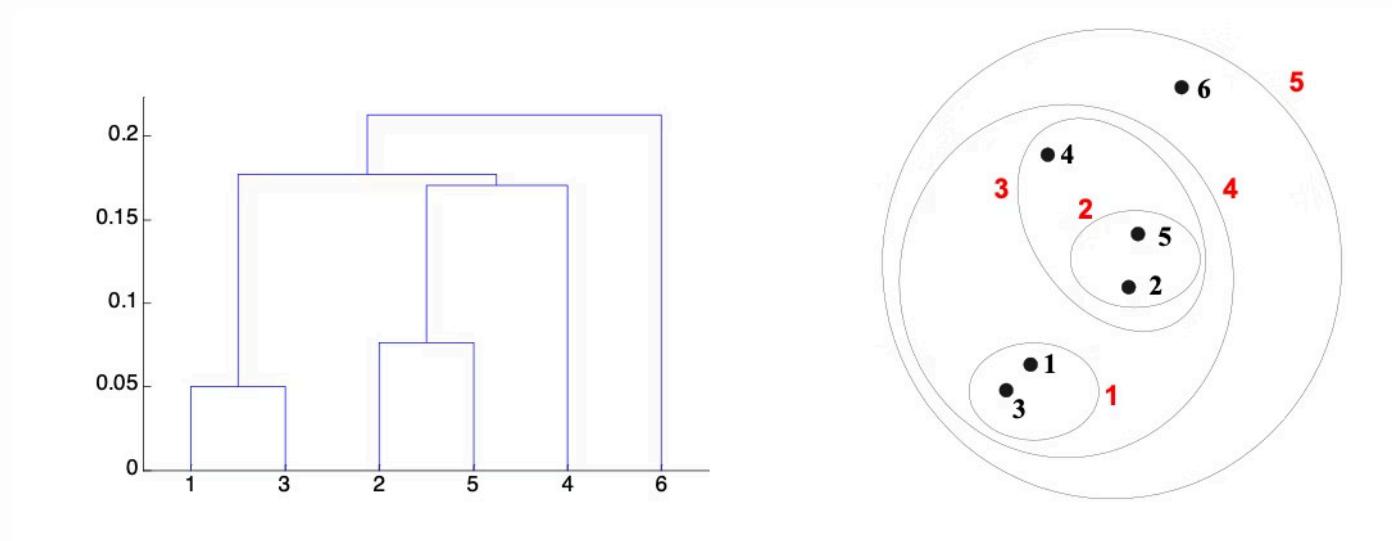
Original Points



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 dendrogram at the proper level
- They may correspond to meaningful taxonomies
 - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, ...)

Hierarchical Clustering

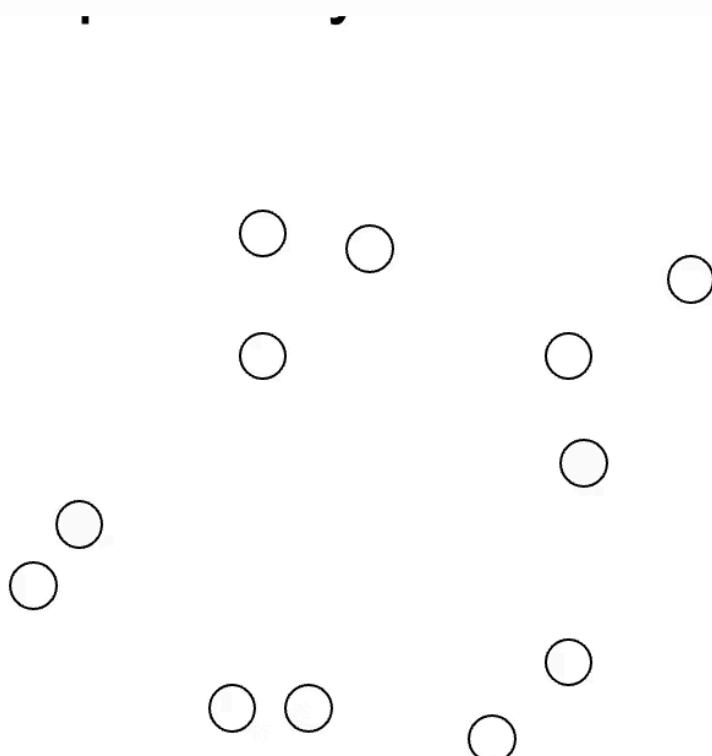
- Two main types of hierarchical clustering
- Agglomerative:
 - Start with the points as individual clusters
 - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
- Divisive:
 - Start with one, all-inclusive cluster
 - At each step, split a cluster until each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

Agglomerative Clustering Algorithm

- More popular hierarchical clustering technique
- Basic algorithm is straightforward
 - a. Compute the proximity matrix
 - b. Let each data point be a cluster
 - c. Repeat
 - i. Merge the two closest clusters
 - ii. Update the proximity matrix
 - d. **Until** only a single cluster remains
- Key operation is the computation of the proximity of two clusters
 - Different approaches to defining the distance between clusters distinguish the different algorithms

Starting Situation

Start with clusters of individual points and a proximity matrix



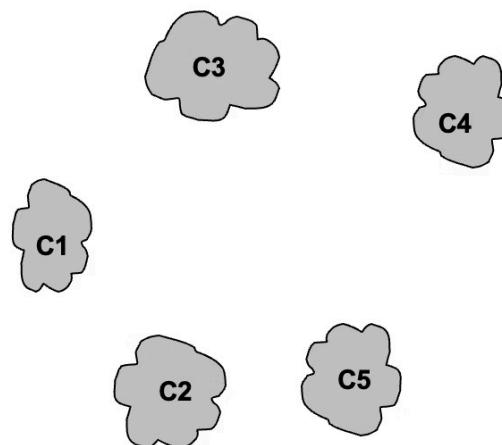
	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
...						

Proximity Matrix



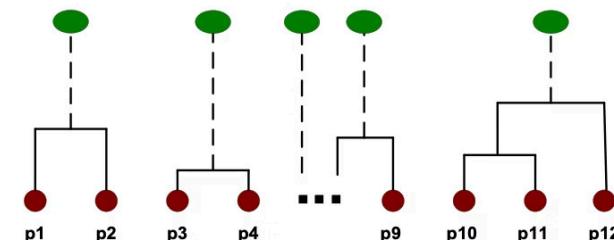
Intermediate Situation

- After some merging steps, we have some clusters



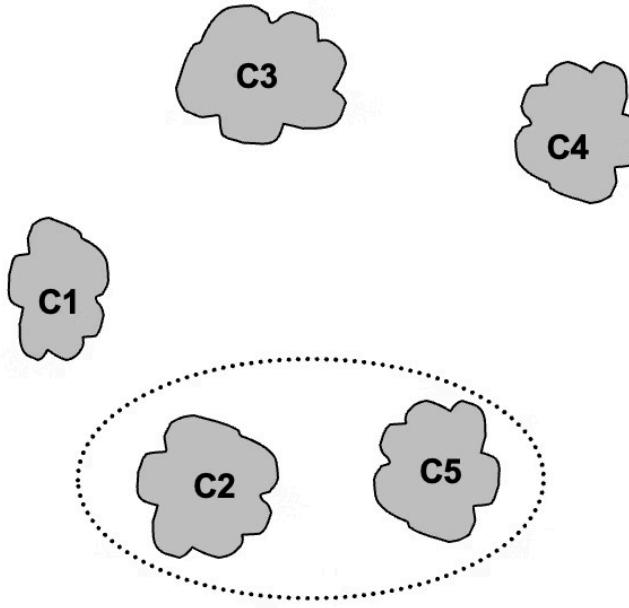
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



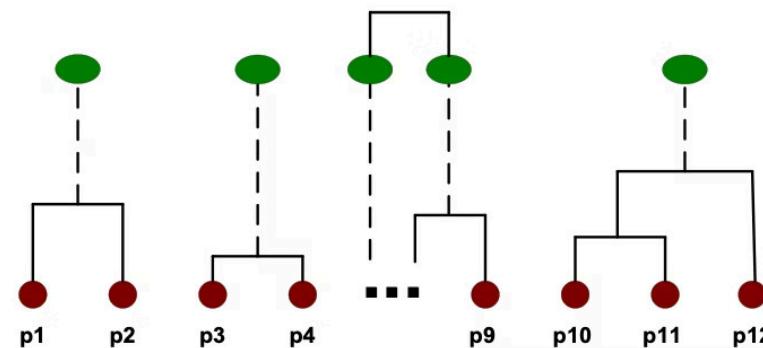
Intermediate Situation

We want to merge the two closest clusters (C_2 and C_5) and update the proximity matrix



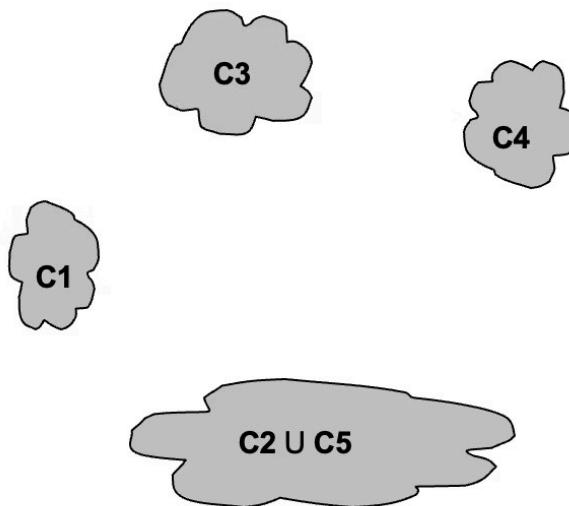
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



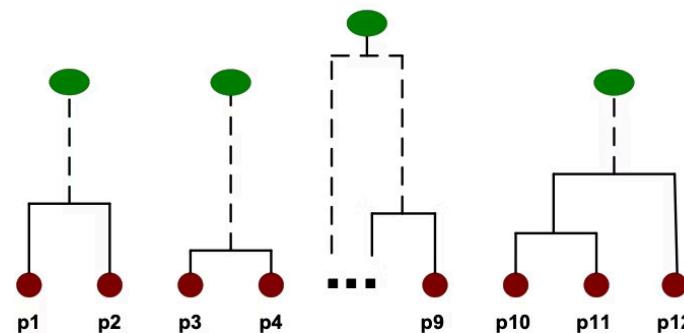
After Merging

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



		C_2	C_1	C_5	C_3	C_4
		C_1	?			
		$C_2 \cup C_5$?	?	?	?
		C_3	?			
		C_4	?			

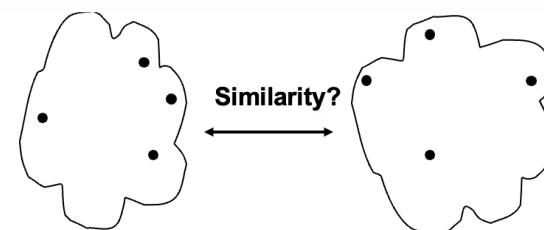
Proximity Matrix



How to Define Inter-Cluster Similarity

Similarity?

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

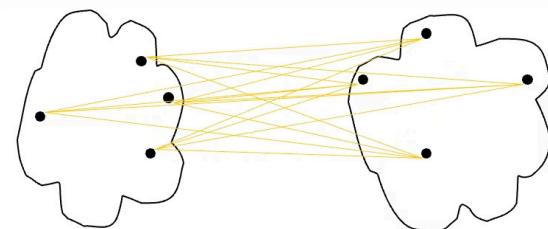


	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						

How to Define Inter-Cluster Similarity

Methods for Measuring Similarity

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

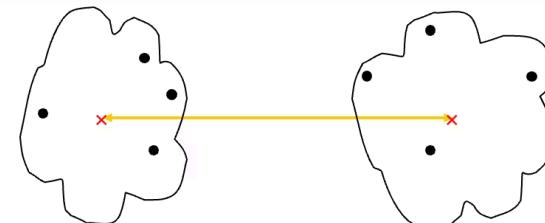


	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						

Proximity Matrix

How to Define Inter-Cluster Similarity

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



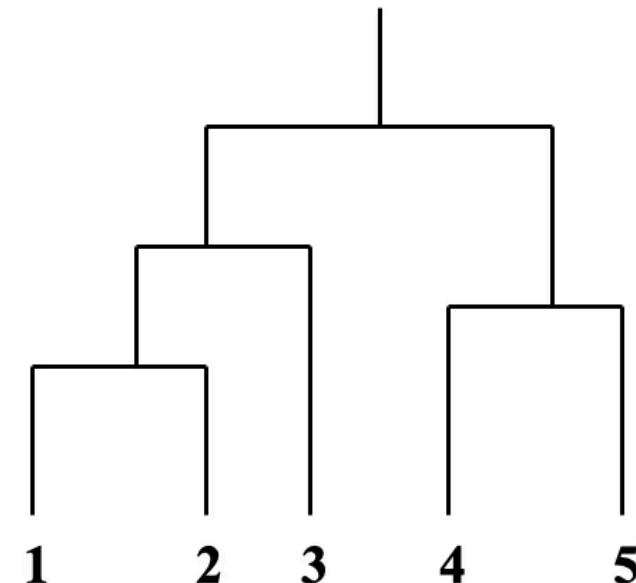
	p1	p2	p3	p4	p5	...
p1						
.

Proximity Matrix

Cluster Similarity: MIN or Single Link

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

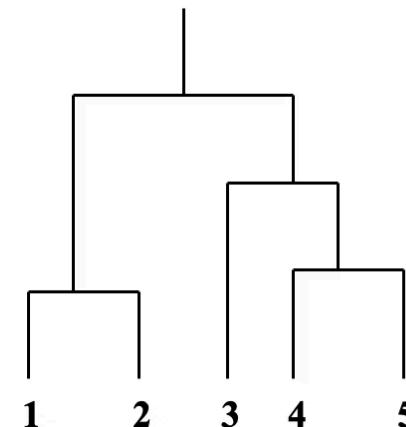
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



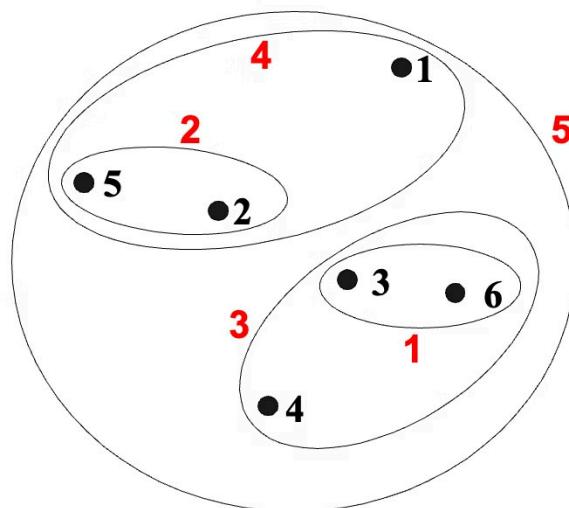
Cluster Similarity: MAX or Complete Linkage

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

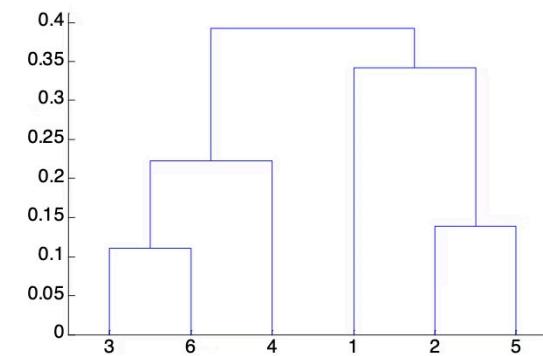
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: MAX

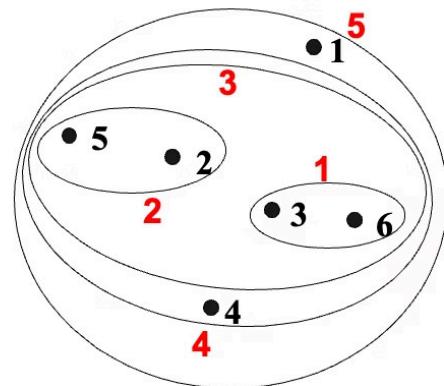


Nested Clusters

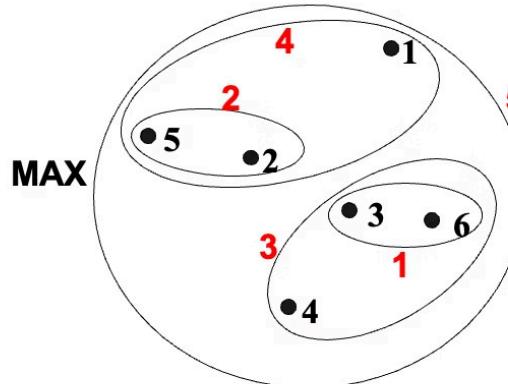


Dendrogram

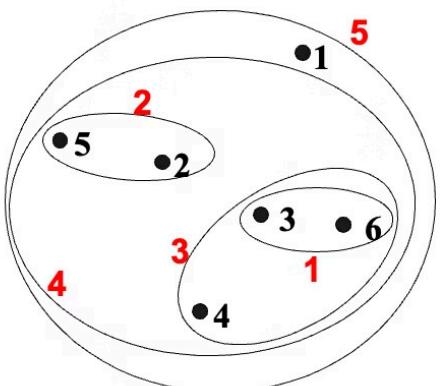
Hierarchical Clustering: Comparison



MIN

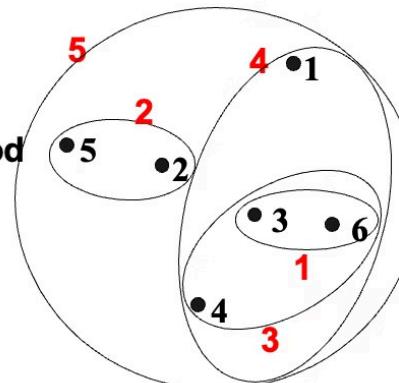


MAX



Group Average

Ward's Method



Hierarchical Clustering: Time and Space requirements

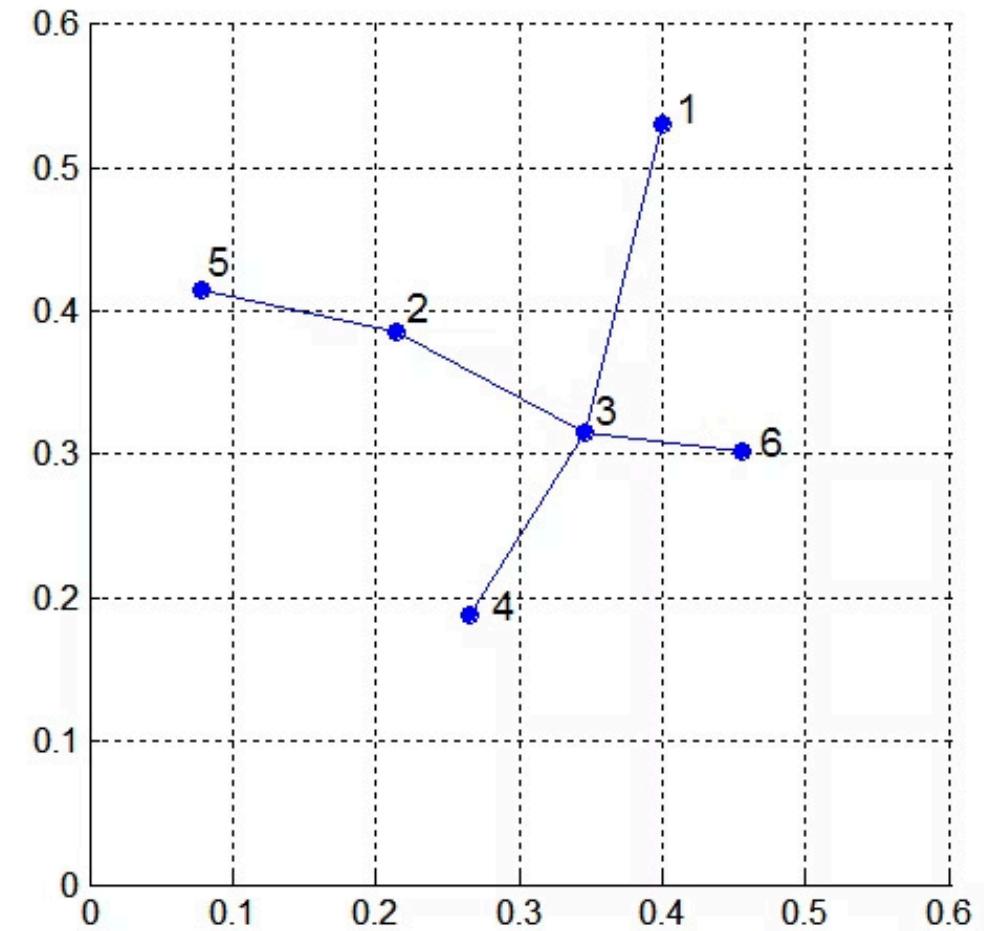
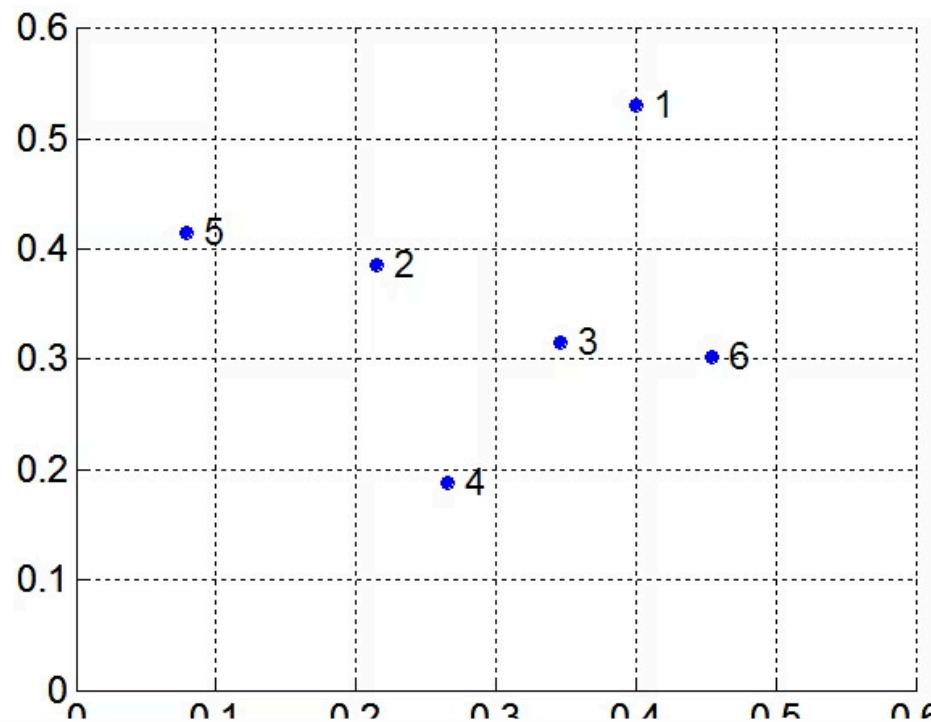
- $O(N^2)$ space since it uses the proximity matrix.
 - N is the number of points.
- $O(N^3)$ time in many cases
 - There are N steps and at each step the size, N^2 , proximity matrix must be updated and searched
 - Complexity can be reduced to $O(N^2 \log(N))$ time for some approaches

Hierarchical Clustering: Problems and Limitations

- Once a decision is made to combine two clusters, it cannot be undone
- No objective function is directly minimized
- Different schemes have problems with one or more of the following:
 - Sensitivity to noise and outliers
 - Difficulty handling different sized clusters and convex shapes
 - Breaking large clusters

MST: Divisive Hierarchical Clustering

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



MST: Divisive Hierarchical Clustering

- Use MST for constructing hierarchy of clusters

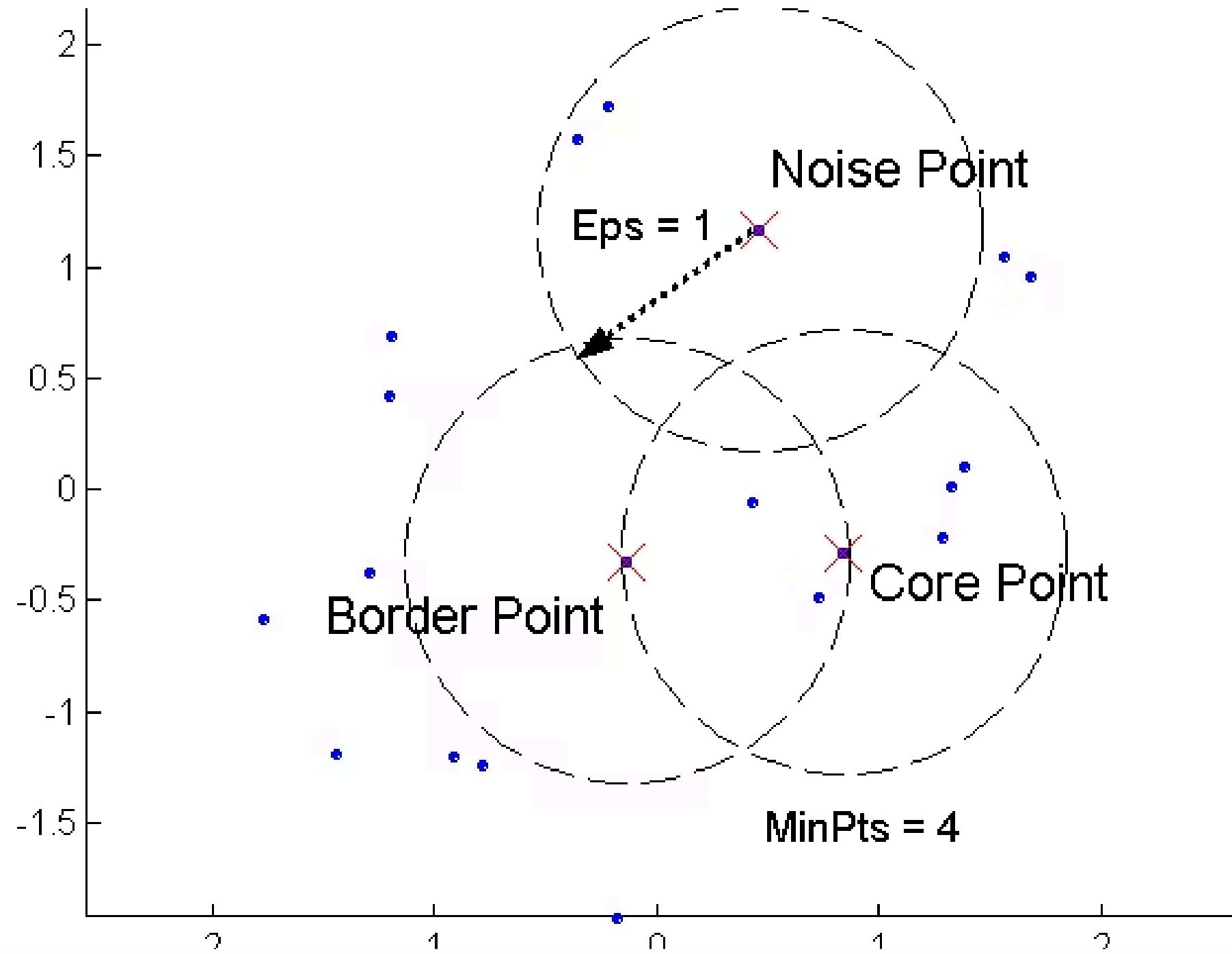
Algorithm 7.5 MST Divisive Hierarchical Clustering Algorithm

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

DBSCAN

- DBSCAN is a density-based algorithm.
 - Density = number of points within a specified radius (Eps)
 - A point is a **core point** if it 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 not a core point or a border point.

DBSCAN: Core, Border, and Noise Points

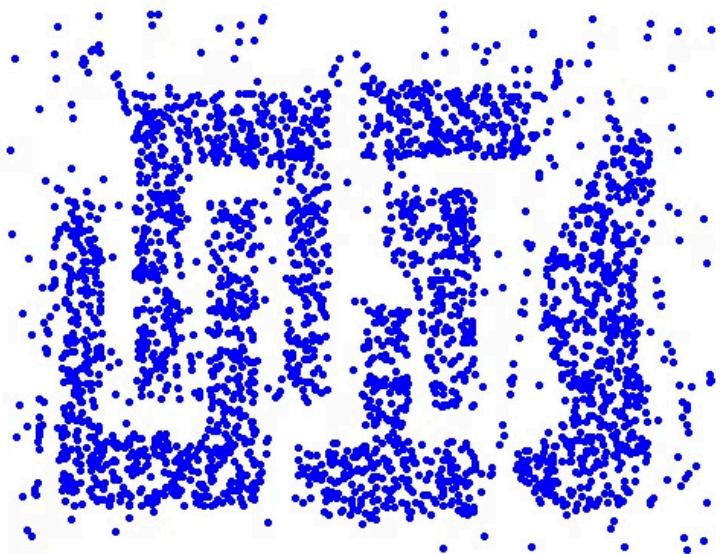


DBSCAN Algorithm

- Eliminate noise points
- Perform clustering on the remaining points

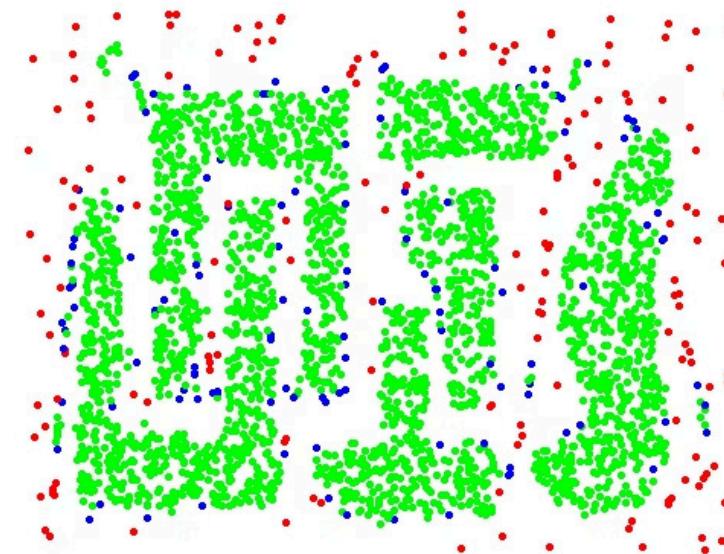
```
current_cluster_label ← 1
for all core points do
    if the core point has no cluster label then
        current_cluster_label ← current_cluster_label + 1
        Label the current core point with cluster label current_cluster_label
    end if
    for all points in the  $Eps$ -neighborhood, except  $i^{th}$  the point itself do
        if the point does not have a cluster label then
            Label the point with cluster label current_cluster_label
        end if
    end for
end for
```

DBSCAN: Core, Border and Noise Points



Original Points

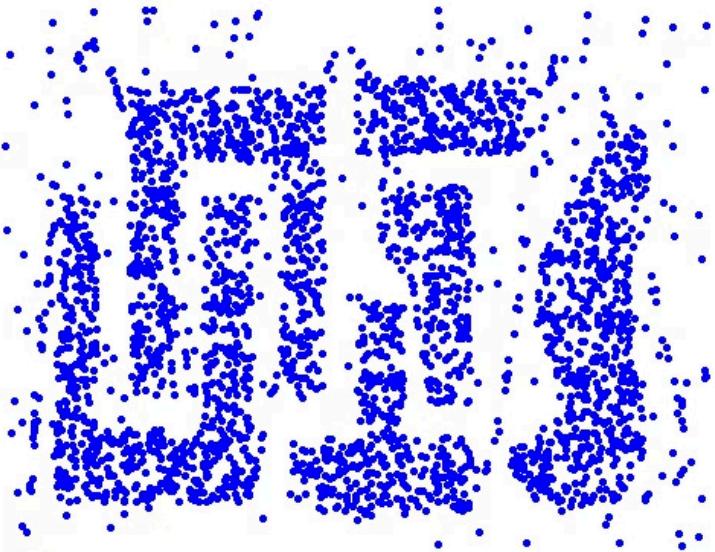
Eps = 10, MinPts = 4



Point types: core, border and noise

When DBSCAN Works Well

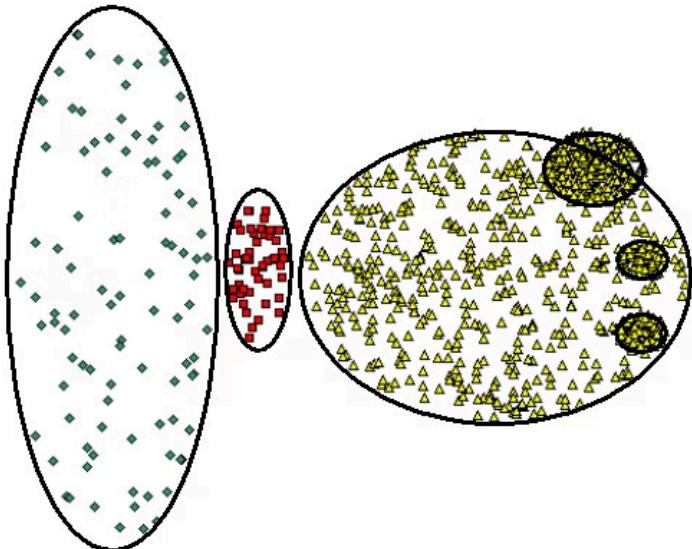
Clusters



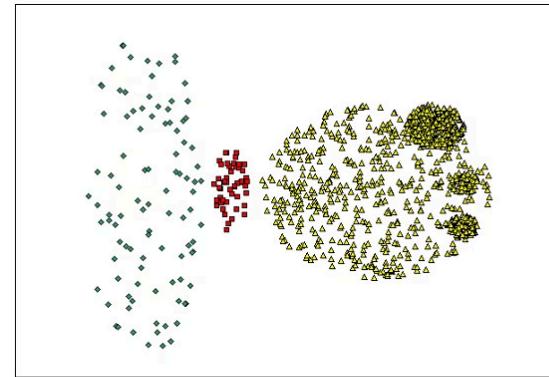
Original Points

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

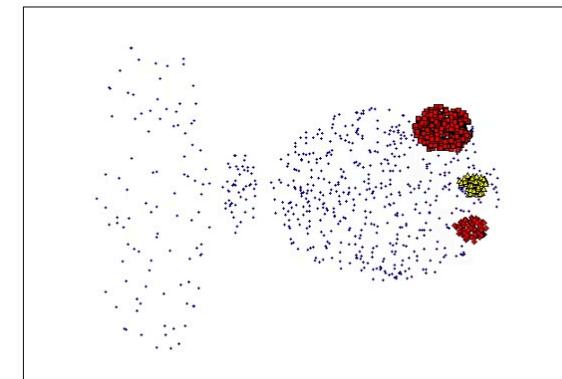
When DBSCAN Does NOT Work Well



Original Points



(MinPts=4, Eps=9.75).



(MinPts=4, Eps=9.92)

DBSCAN: Determining EPS and MinPts

- Idea is that for points in a cluster, their kth nearest neighbors are at roughly the same distance
- Noise points have the kth nearest neighbor at farther distance
- So, plot sorted distance of every point to its kth nearest neighbor

