

---

# Clustering

## Basic Concepts and Algorithms 1

# Machine learning tasks

---

## Supervised

- Classification
- Regression
- Recommender systems
- Reinforcement learning

We will cover tasks highlighted in red

## Unsupervised

- Clustering
  - Association analysis
- 
- Ranking
  - Anomaly detection

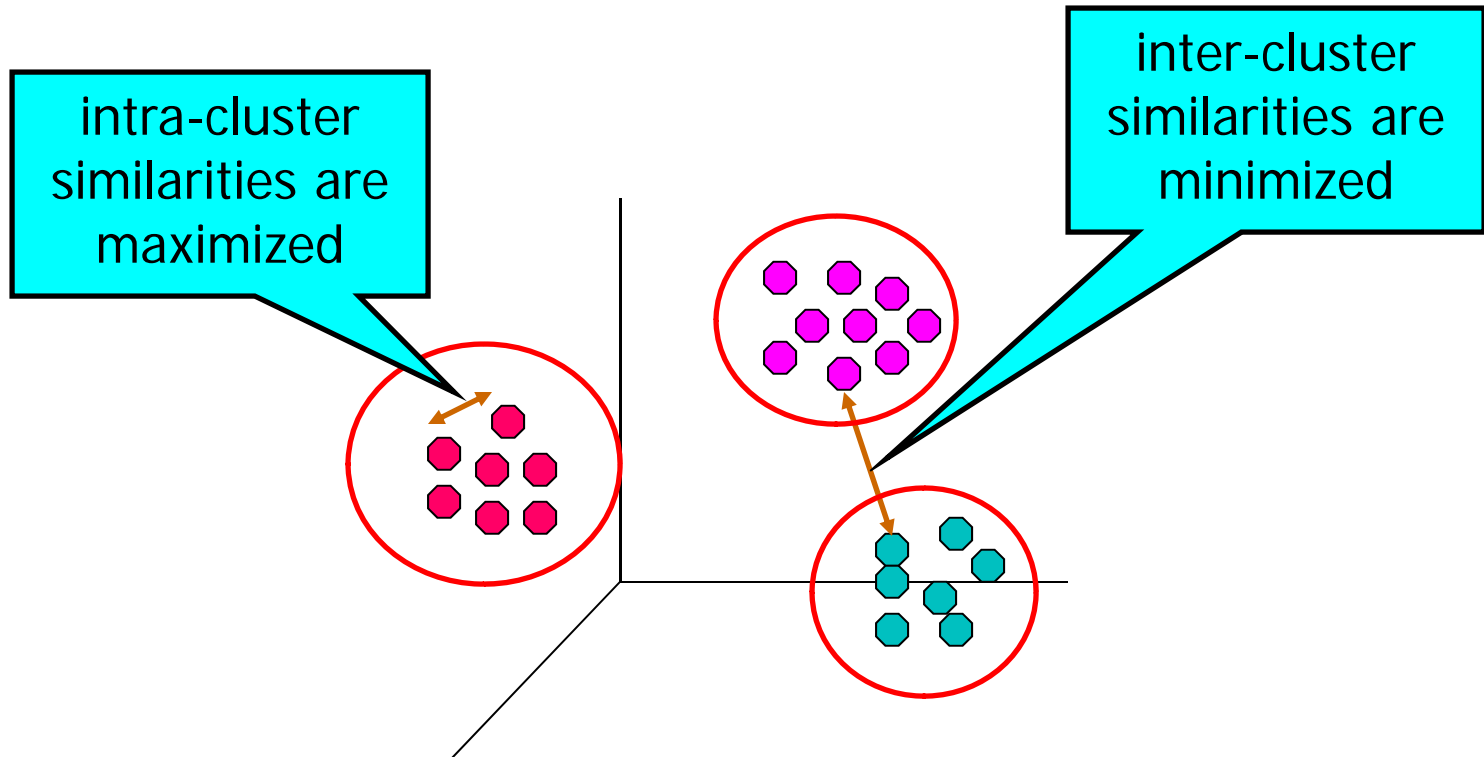
# Clustering definition

---

- Given:
  - Set of data points
  - Set of attributes on each data point
  - A measure of similarity (or distance) between data points
- Find clusters such that:
  - Data points within a cluster are more similar to one another
  - Data points in separate clusters are less similar to one another
- Similarity measures:
  - Euclidean distance if attributes are continuous
  - Other problem-specific measures

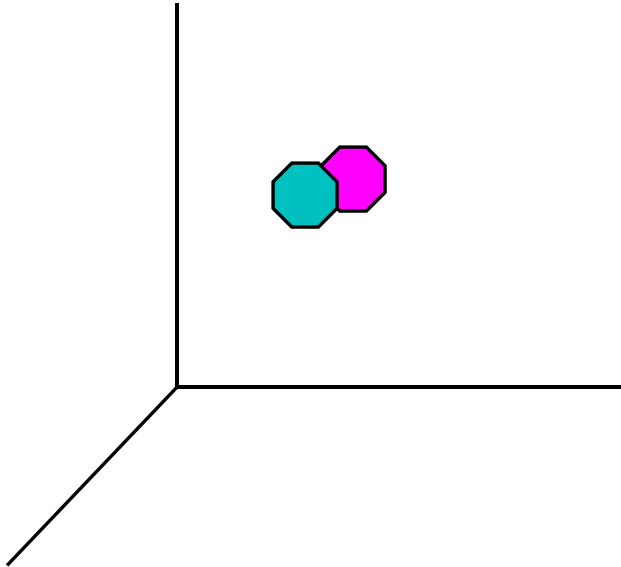
# Clustering definition

- Find groups (clusters) of data points such that data points in a group will be similar (or related) to one another and different from (or unrelated to) the data points in other groups



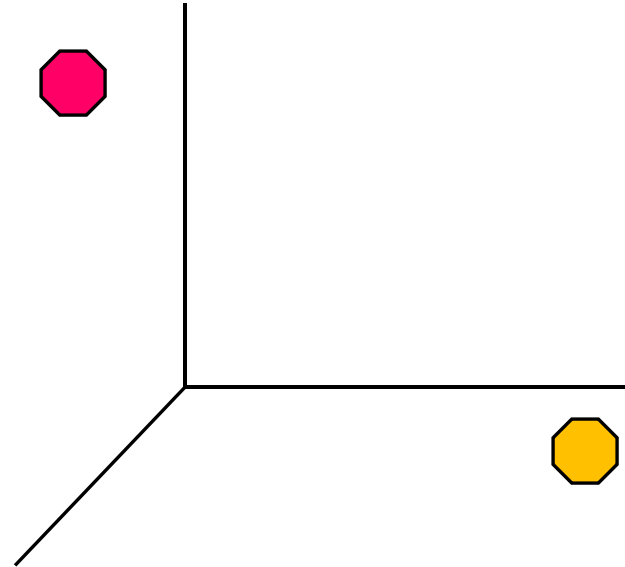
# Similarity and distance

---



**high similarity**

**small distance**



**low similarity**

**large distance**

**Examples:**

**Euclidean distance**

**cosine similarity**

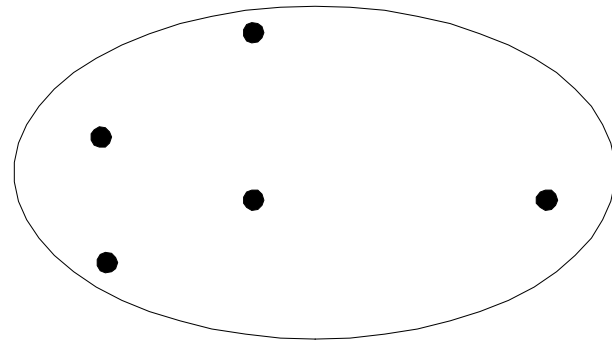
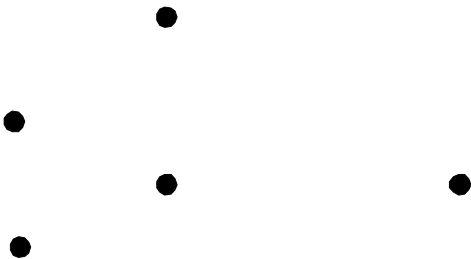
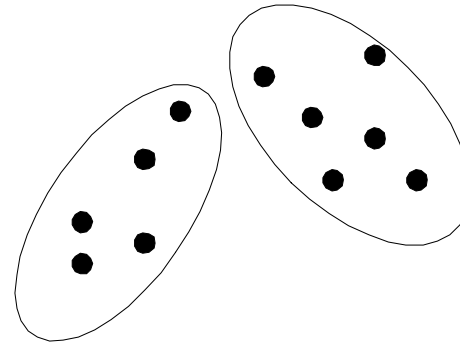
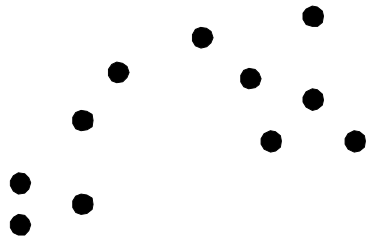
# Approaches to clustering

---

- A **clustering** is a set of clusters
- Important distinction between **hierarchical** and **partitional** clustering
  - Partitional: data points divided into finite number of *partitions* (non-overlapping subsets)
    - ◆ each data point is assigned to exactly one subset
  - Hierarchical: data points placed into a set of nested clusters, organized into a *hierarchical tree*
    - ◆ tree expresses a continuum of similarities and clustering

# Partitional clustering

---

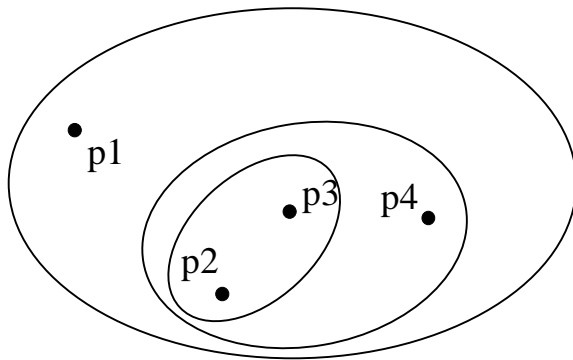


**Original points**

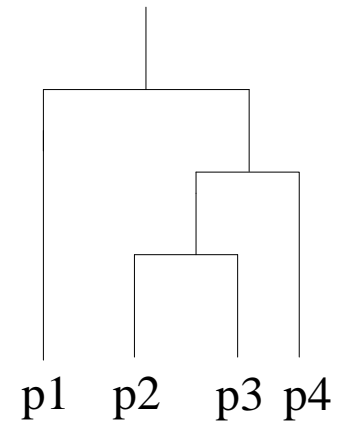
**Partitional clustering**

# Hierarchical clustering

---



**Hierarchical clustering**

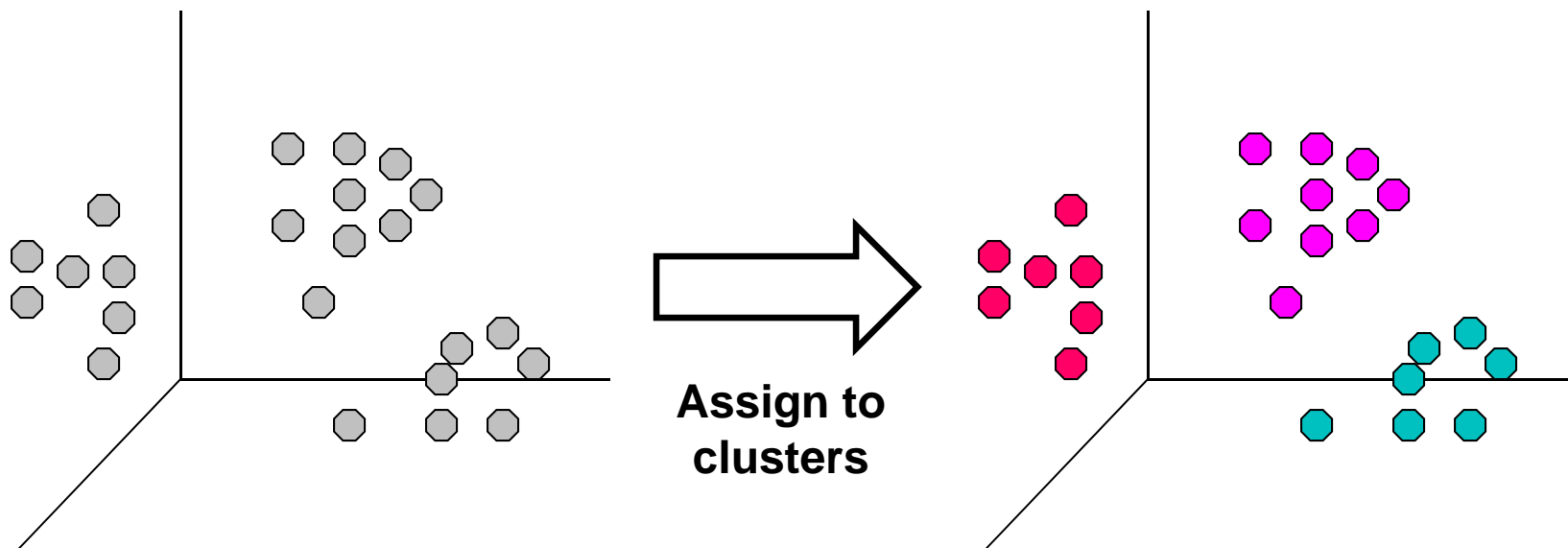


**Dendrogram**



# Partitional clustering illustrated

## Euclidean distance-based clustering in 3D space

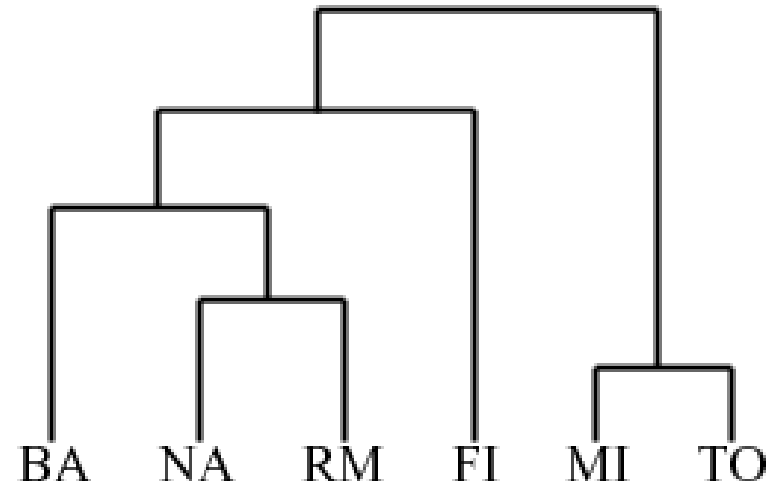


**intra-cluster distances  
are minimized**

**inter-cluster distances  
are maximized**

# Hierarchical clustering illustrated

## Driving distances between Italian cities



# Applications of clustering

## ● Understanding

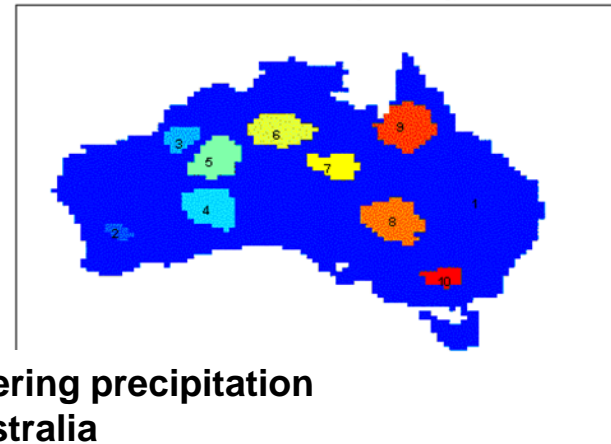
- Group related documents for browsing
- Group genes and proteins that have similar functions
- Group stocks with similar price fluctuations

	<i>Discovered Clusters</i>	<i>Industry Group</i>
<b>1</b>	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
<b>2</b>	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
<b>3</b>	Fannie-Mae-DOWN, Fed-Home-Loan-DOWN, MBNA-Corp-DOWN, Morgan-Stanley-DOWN	Financial-DOWN
<b>4</b>	Baker-Hughes-UP, Dresser-Inds-UP, Halliburton-HLD-UP, Louisiana-Land-UP, Phillips-Petro-UP, Unocal-UP, Schlumberger-UP	Oil-UP

## ● Summarization

- Reduce the size of large data sets by finding representative samples

10 Precip Clusters usin SNN Clustering (12 mo. avg, NN = 100 )



Clustering precipitation in Australia

# Clustering application 1

---

- Market segmentation
  - Goal: subdivide a market into distinct subsets of customers, such that each subset is conceivably a submarket which can be reached with a customized marketing mix.
  - Approach:
    - ◆ Collect different attributes of customers based on their geographical and lifestyle related information.
    - ◆ Find clusters of similar customers.
    - ◆ Measure the clustering quality by observing buying patterns of customers in same cluster vs. those from different clusters.

# Clustering application 2

---

- Document clustering
  - Goal: Find groups of documents that are similar to each other based on the important terms appearing in them.
  - Approach: Identify frequently occurring terms in each document. Form a similarity measure based on the frequencies of different terms. Use it to cluster.
  - Benefit: Information retrieval can utilize the clusters to relate a new document or search term to clustered documents.

# Document clustering example

---

- Items to cluster: 3204 articles of Los Angeles Times.
- Similarity measure: Number of words in common between a pair of documents (after some word filtering).

<i>Category</i>	<i>Total Articles</i>	<i>Correctly Placed</i>
<i>Financial</i>	555	364
<i>Foreign</i>	341	260
<i>National</i>	273	36
<i>Metro</i>	943	746
<i>Sports</i>	738	573
<i>Entertainment</i>	354	278

# Clustering application 3

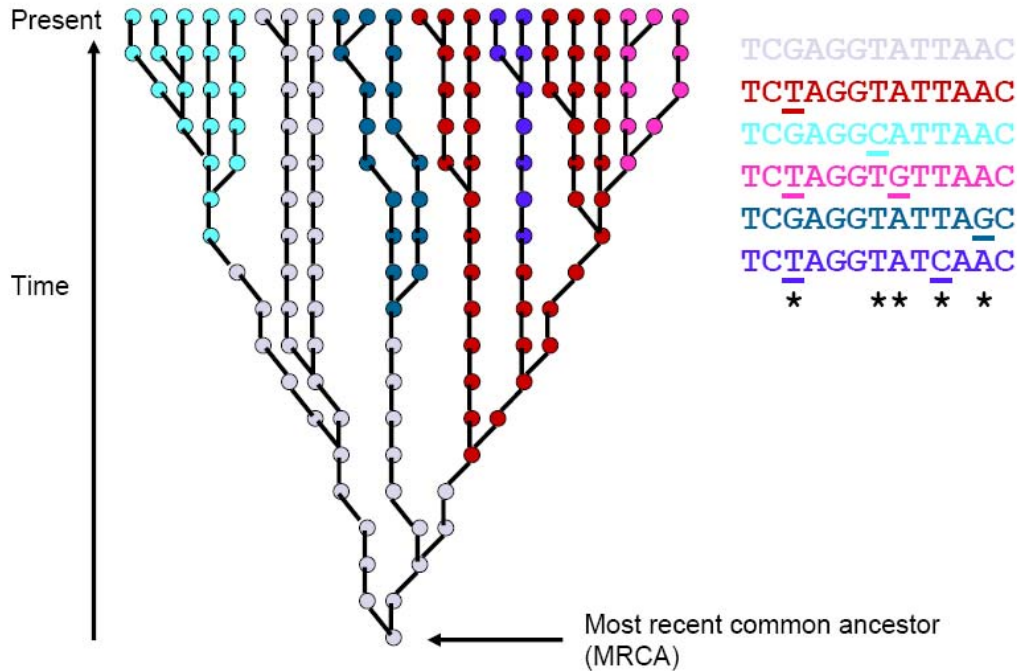
---

- Image segmentation with mean-shift algorithm
- Allows clustering of pixels in combined (R, G, B) plus (x, y) space



# Clustering application 4

- Genetic demography





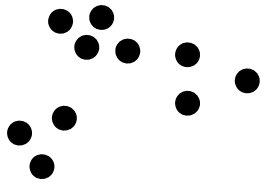
# What is not clustering?

---

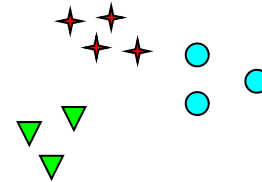
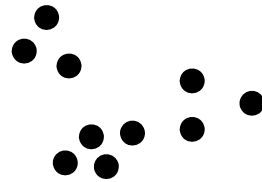
- Supervised classification or regression
  - Have class label or response information
- Simple segmentation
  - Dividing students into different registration groups alphabetically, by last name
- Results of a query
  - Groupings are a result of an external specification

# Notion of a cluster can be ambiguous

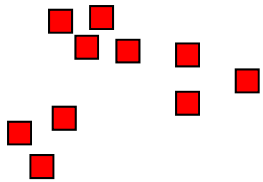
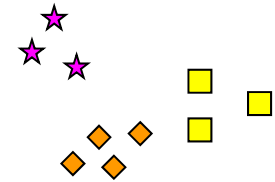
---



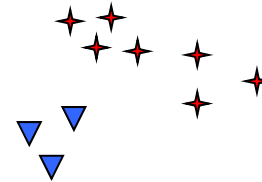
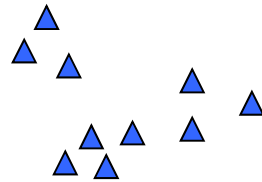
How many clusters?



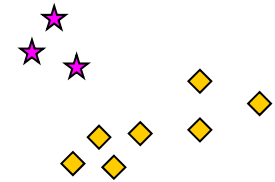
6 clusters



2 clusters



4 clusters



# Other approaches to clustering

---

- 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
  - Clusters of widely different sizes, shapes, and densities

# Types of clusters

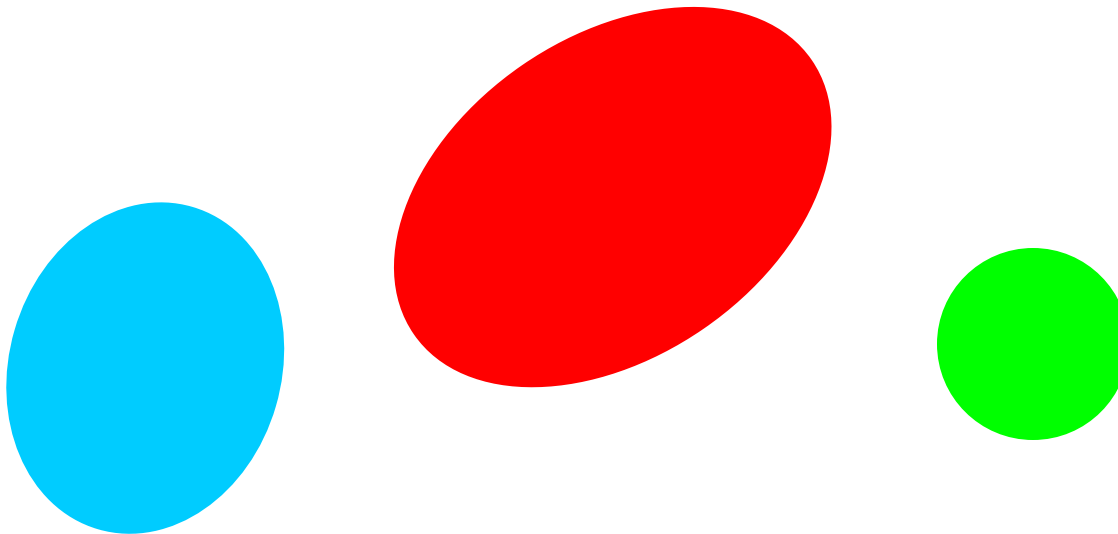
---

- 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.

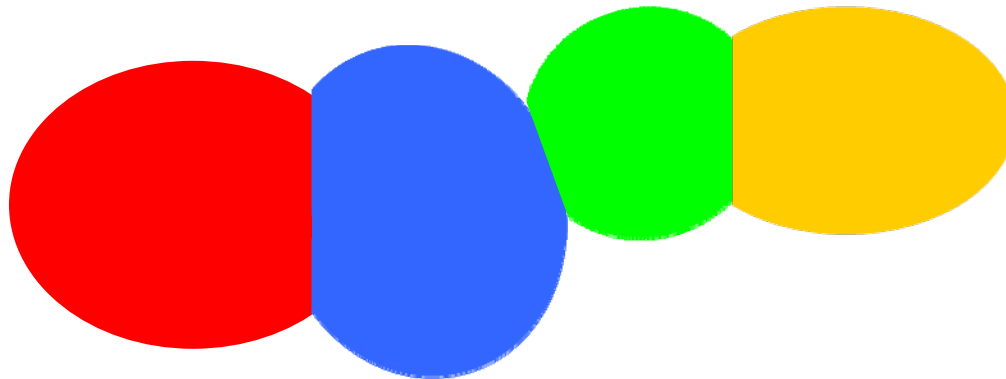


**3 well-separated clusters**

# Types of clusters: center-based

---

- Center-based clusters
  - A cluster is a set of points such that a point in a cluster is closer (more similar) to the “center” of that cluster than to the center of any other cluster.
  - The center of a cluster can be:
    - ◆ the **centroid**, the average position of all the points in the cluster
    - ◆ a **medoid**, the most “representative” point of a cluster

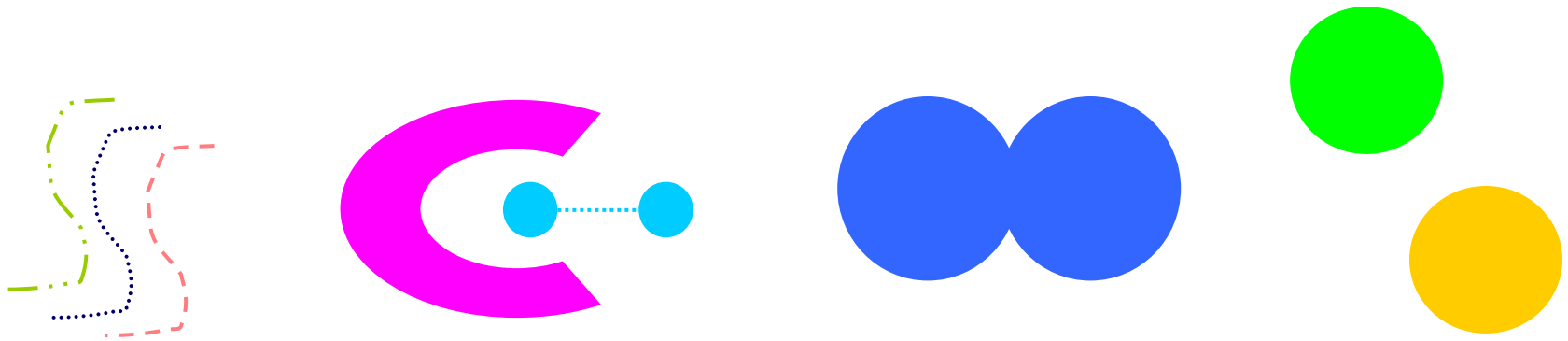


**4 center-based clusters**

# Types of clusters: contiguity-based

---

- Contiguous clusters (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.

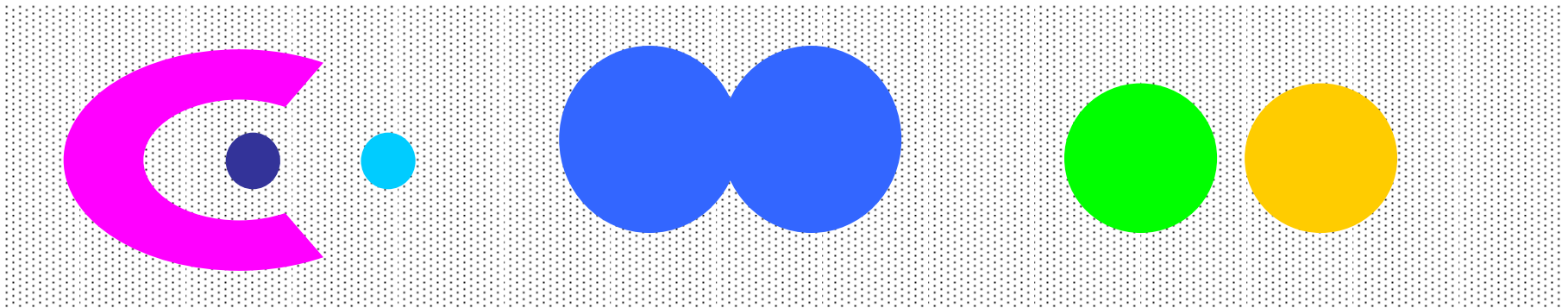


**8 contiguous clusters**

# Types of clusters: density-based

---

- Density-based clusters
  - 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.



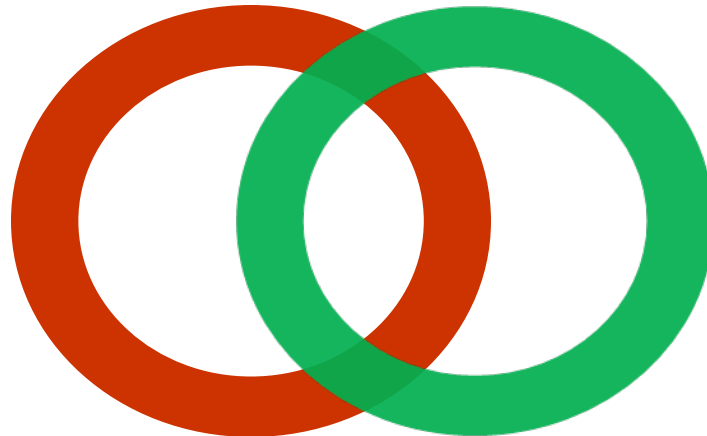
**6 density-based clusters**



# Types of clusters: conceptual clusters

---

- Shared property or conceptual clusters
  - A cluster is a set of objects that share some common property or represent a particular concept.
  - The most general notion of a cluster; in some ways includes all other types.



**2 overlapping concept clusters**

# Types of clusters: objective function

---

- Clusters defined by an objective function
  - Set of clusters minimizes or maximizes some 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 objective function.
    - ◆ Hierarchical clustering algorithms typically have local objective function.
    - ◆ Partitional algorithms typically have global objective function.
  - 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.
    - ◆ Example: Gaussian mixture models (GMM) assume the data is a ‘mixture’ of a fixed number of Gaussian distributions.

# Characteristics of input data are important

---

- Type of similarity 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
- Domain 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 whose centroid it is closest to
- 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 can vary from one run to another.
- The centroid is (typically) the mean of the points in the cluster.
- Similarity is measured by Euclidean distance, cosine similarity, correlation, etc.
- k-Means will converge for common similarity measures mentioned above.
- Most of the convergence happens in the first few iterations.
  - Often the stopping condition is changed 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

# Demos: k-means clustering

---

[..\videos\k-means\\_k2.mp4](#)

[..\videos\k-means\\_k3.mp4](#) (3:05)

on web:

<http://www.youtube.com/watch?v=zaKjh2N8jN4>

<http://www.youtube.com/watch?v=luRb3y8qKX4>

# Evaluating k-means clusterings

---

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

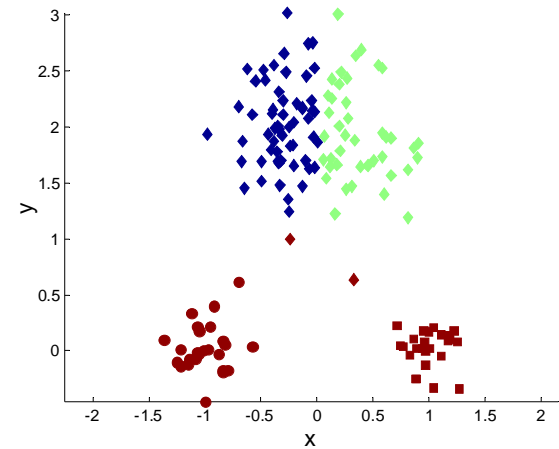
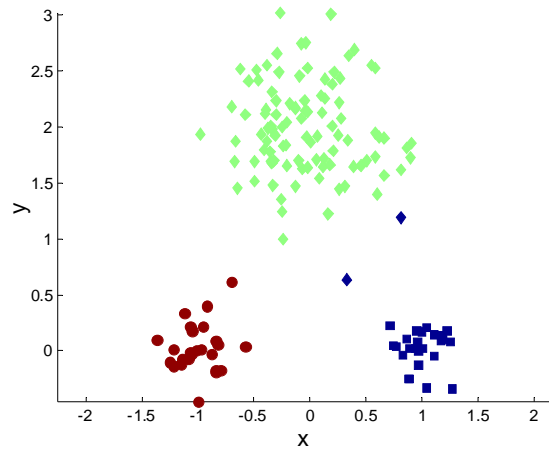
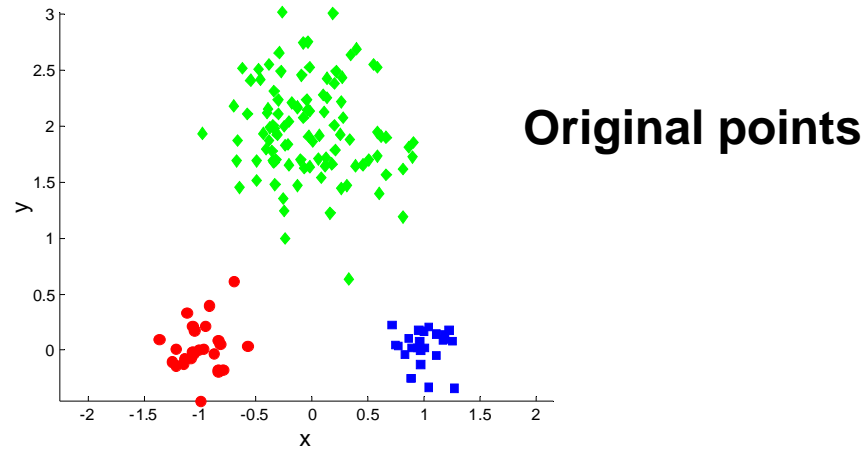
$$SSE = \sum_{i=1}^K \sum_{x \in C_i} \text{dist}^2(m_i, x)$$

where  $x$  is a data point in cluster  $C_i$  and  $m_i$  is the centroid of  $C_i$ .

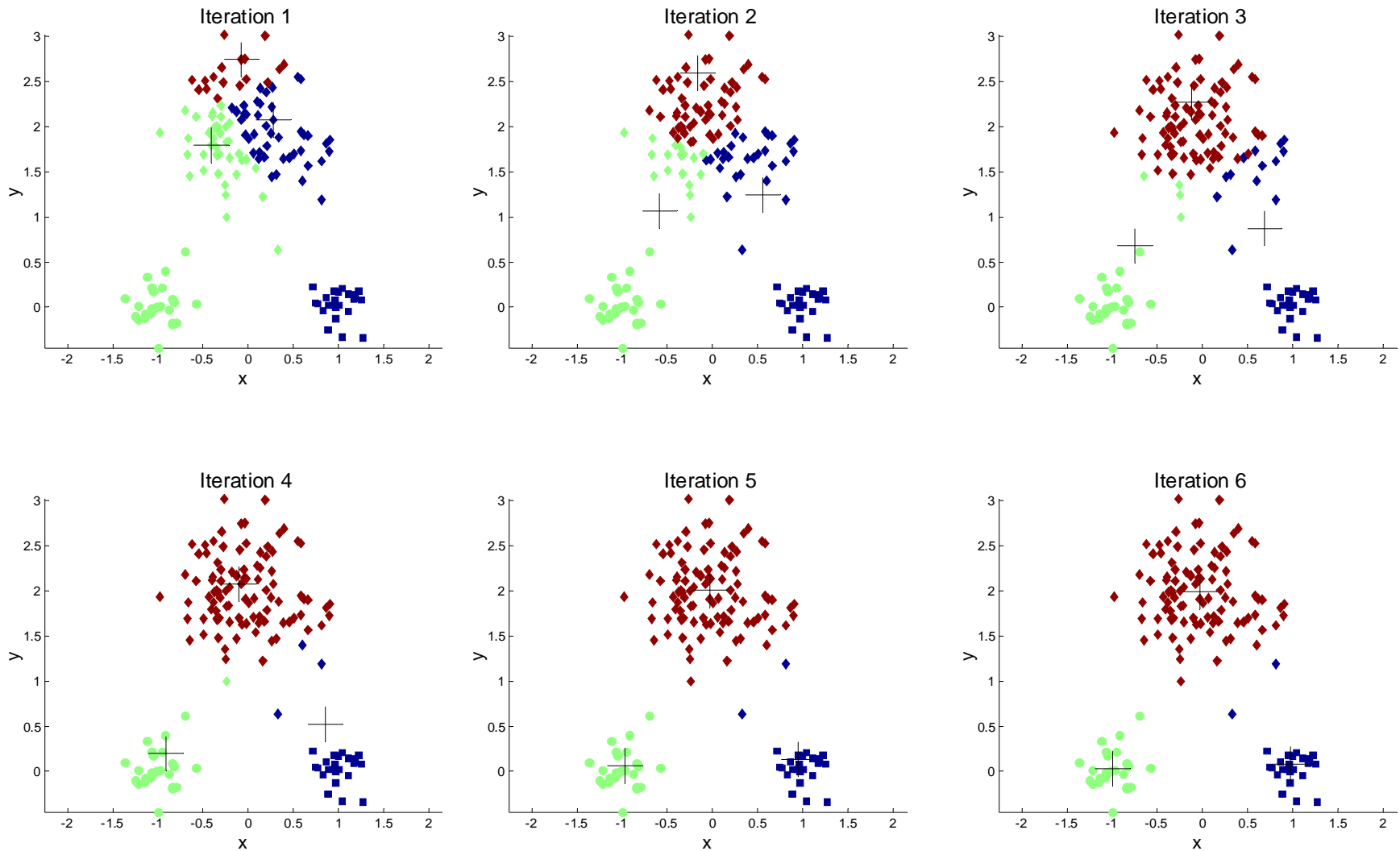
- Given two clusterings, we choose the one with the smallest SSE
- One easy way to reduce SSE is to increase  $k$ , the number of clusters
  - ◆ But a good clustering with smaller  $k$  can have a lower SSE than a poor clustering with higher  $k$



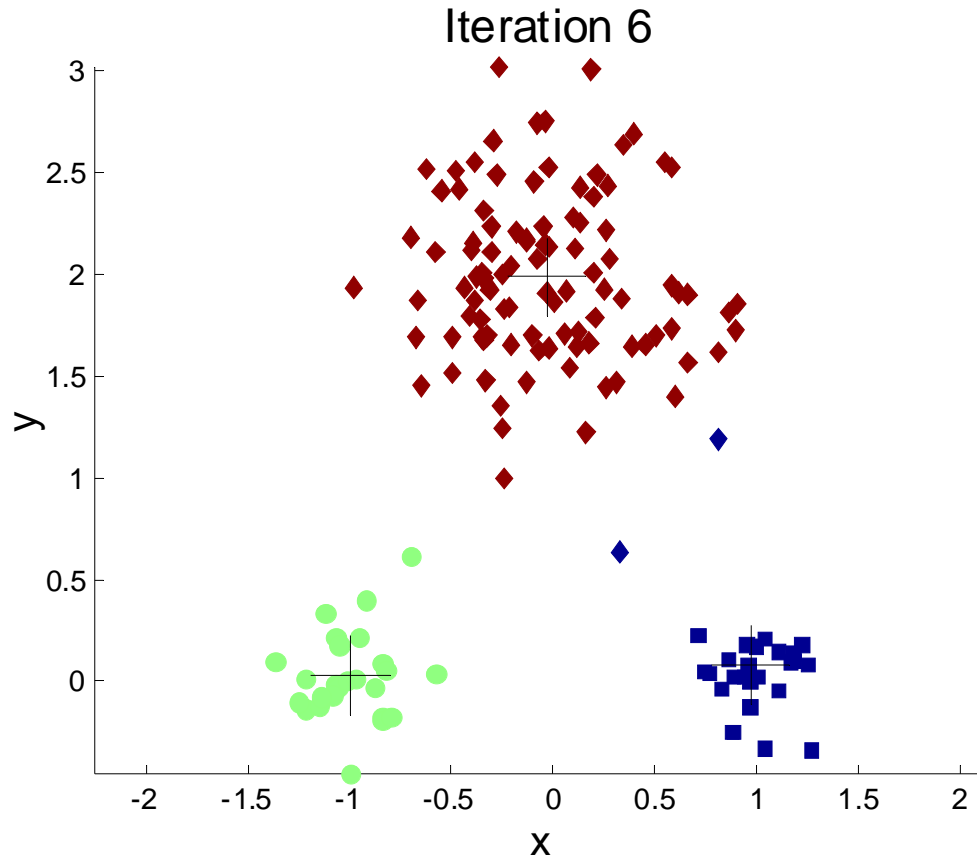
# Two different k-means clusterings



# Impact of initial choice of centroids

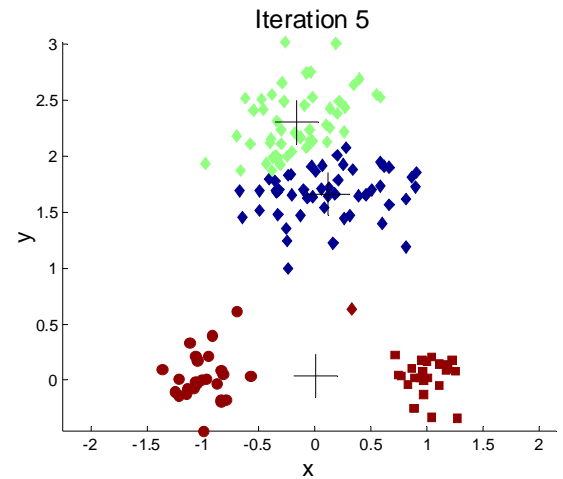
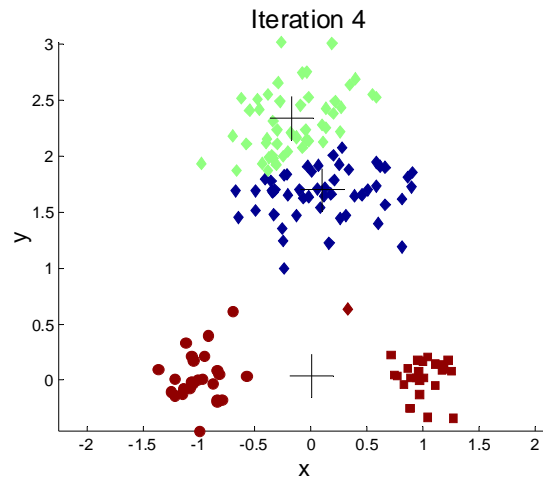
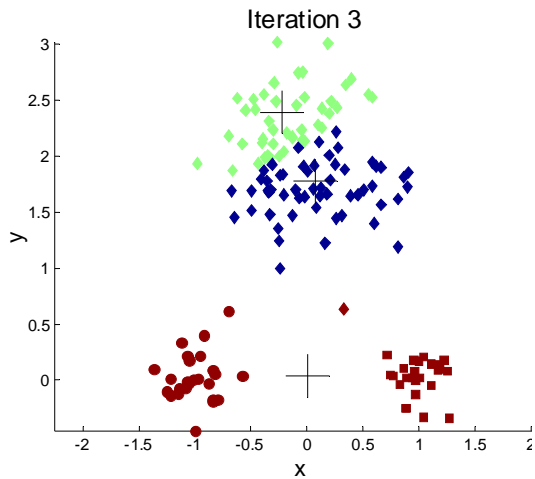
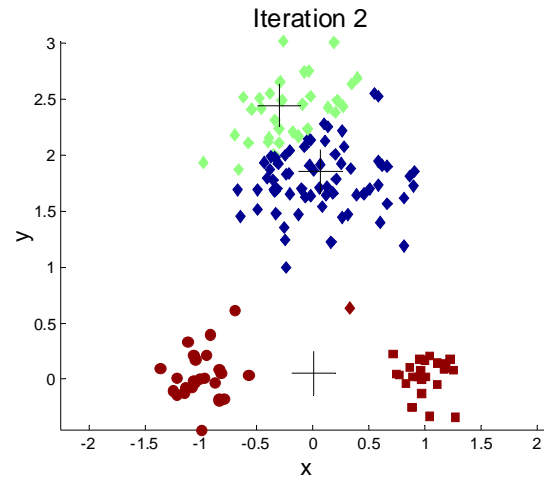
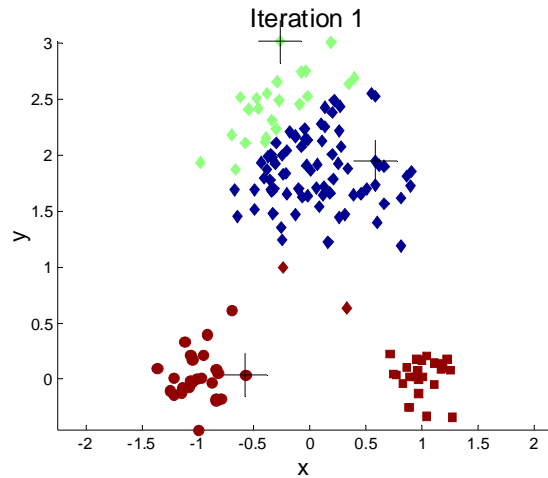


# Impact of initial choice of centroids

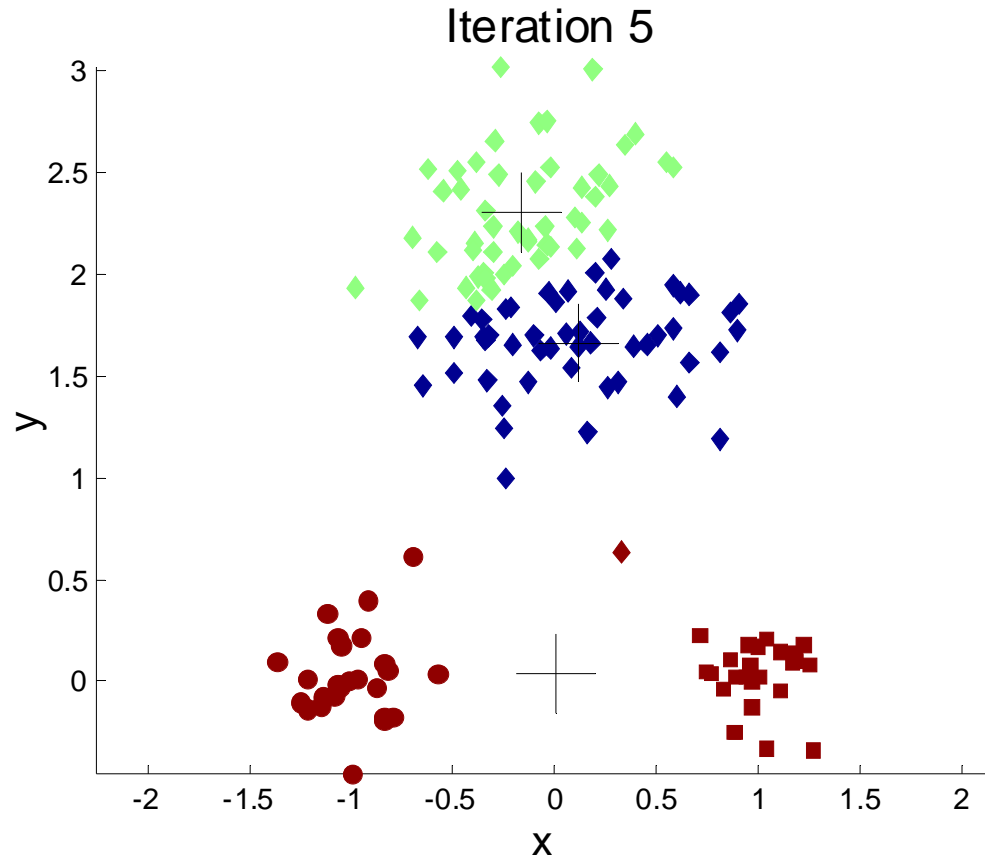


**Good outcome:**  
clusters found by algorithm correspond to natural clusters in data

# Impact of initial choice of centroids



# Impact of initial choice of centroids



**Bad outcome:  
clusters found by algorithm do not correspond to natural clusters in data**

# Problems with selecting initial centroids

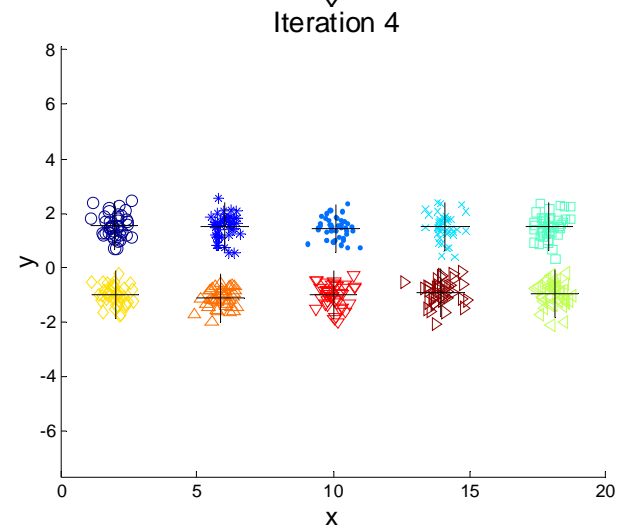
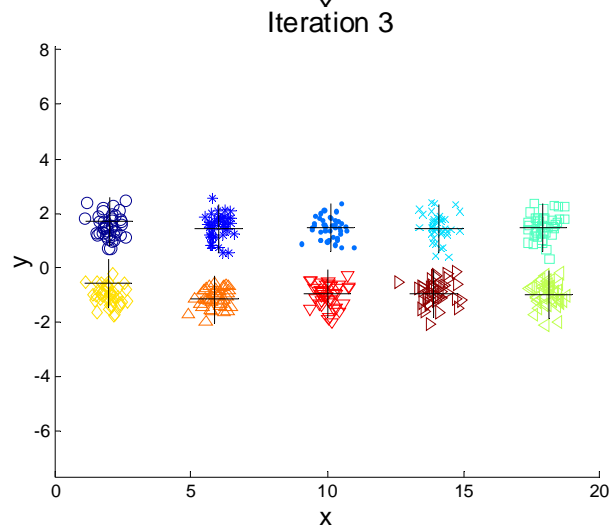
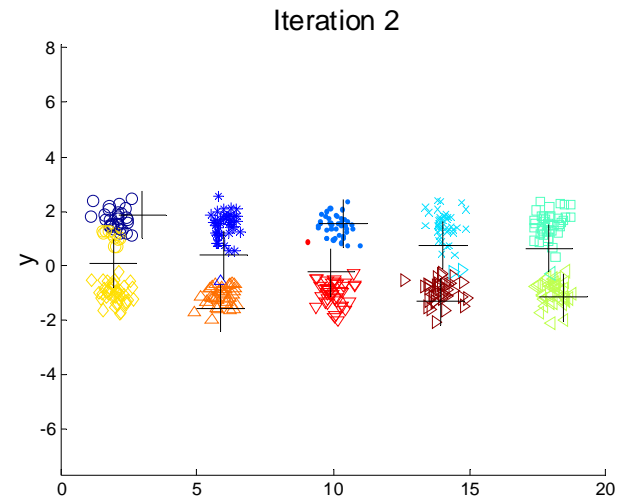
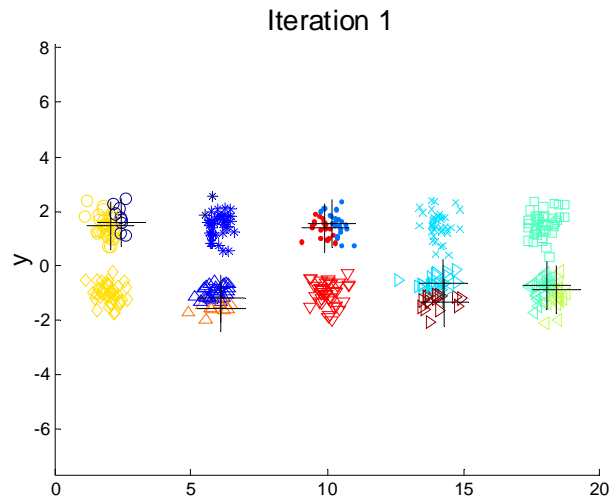
---

- If there are  $k$  'real' clusters then the chance of selecting one centroid from each cluster is small.
  - Chance is really 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

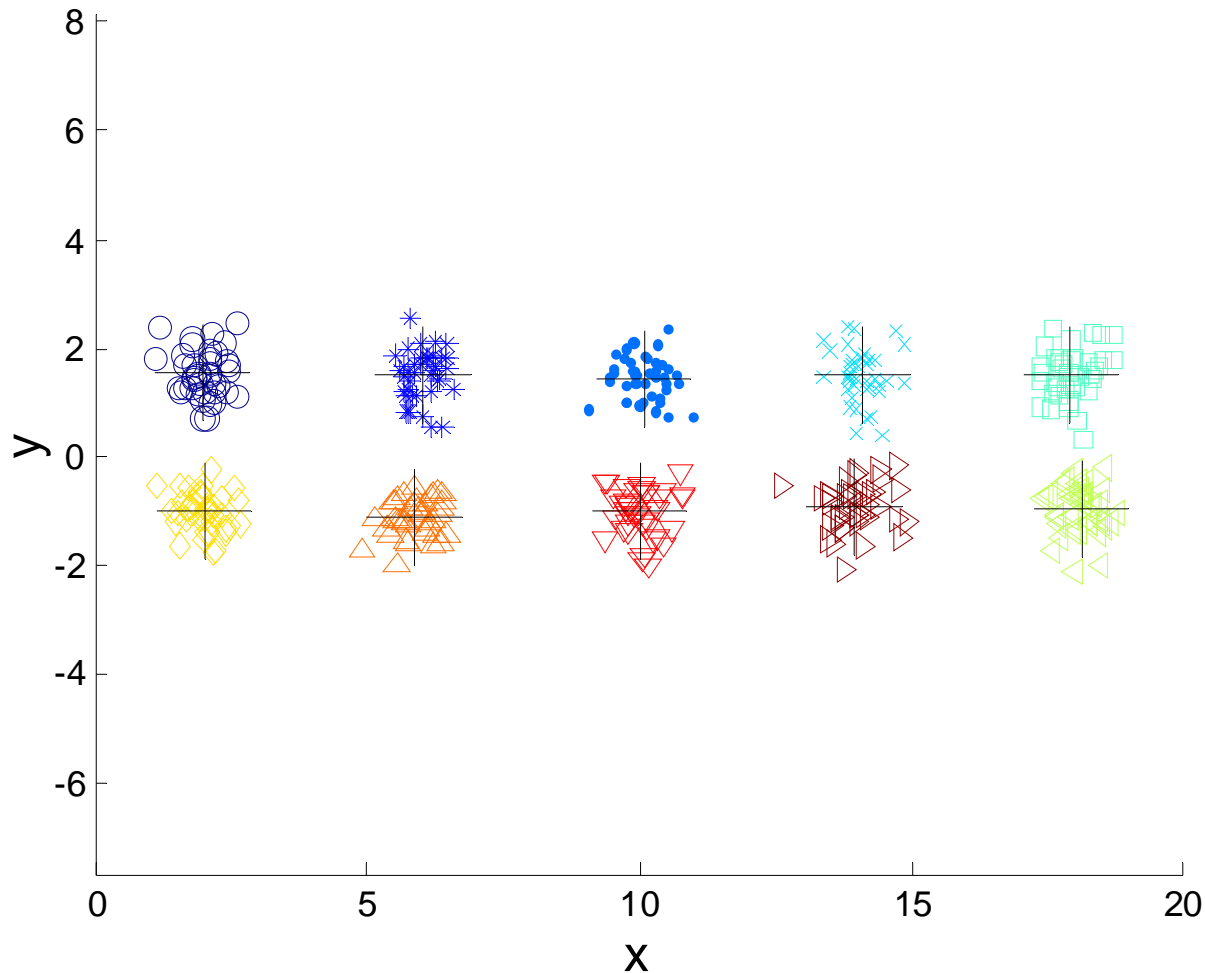
# Ten clusters example



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

# Ten clusters example

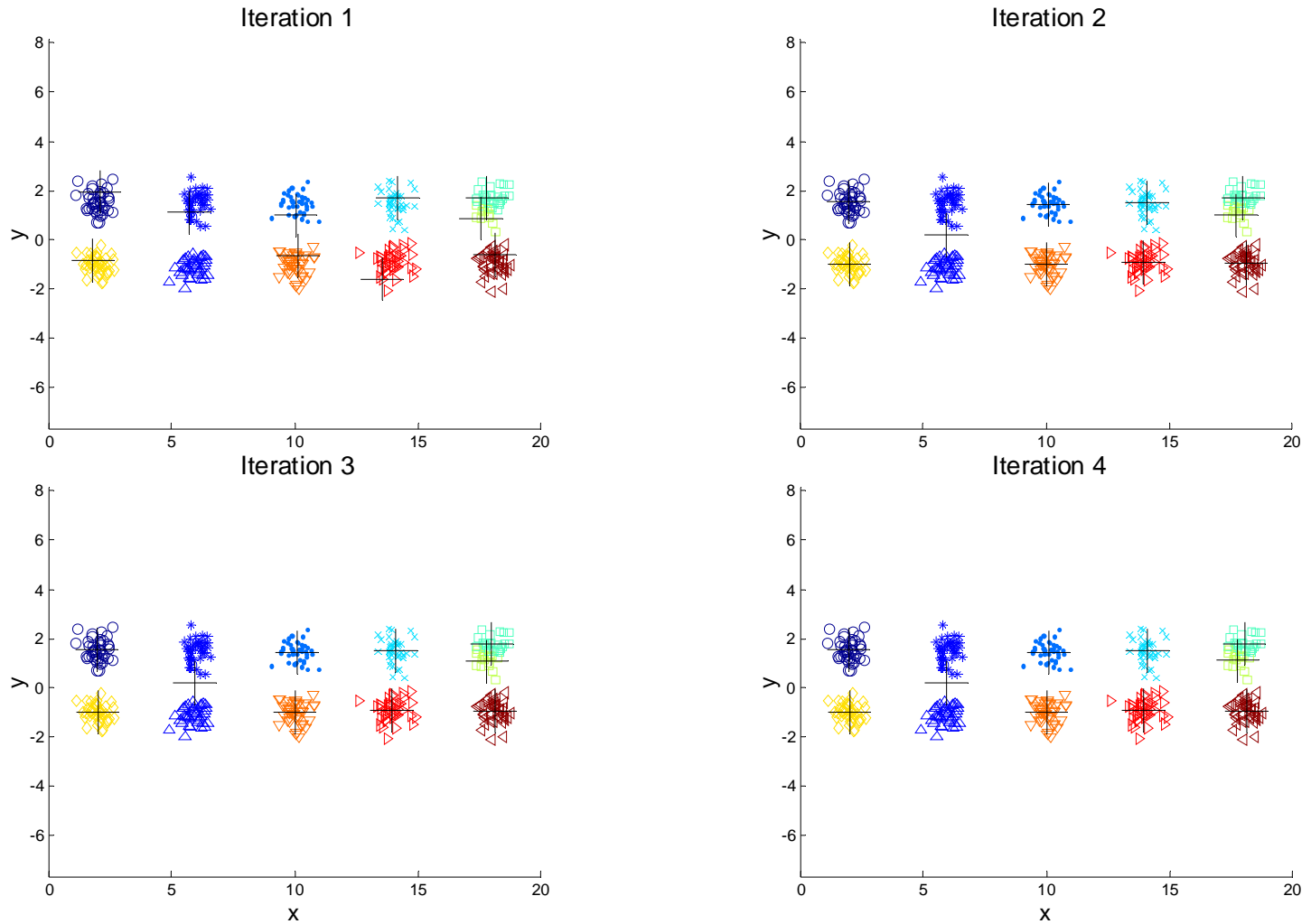
Iteration 4



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



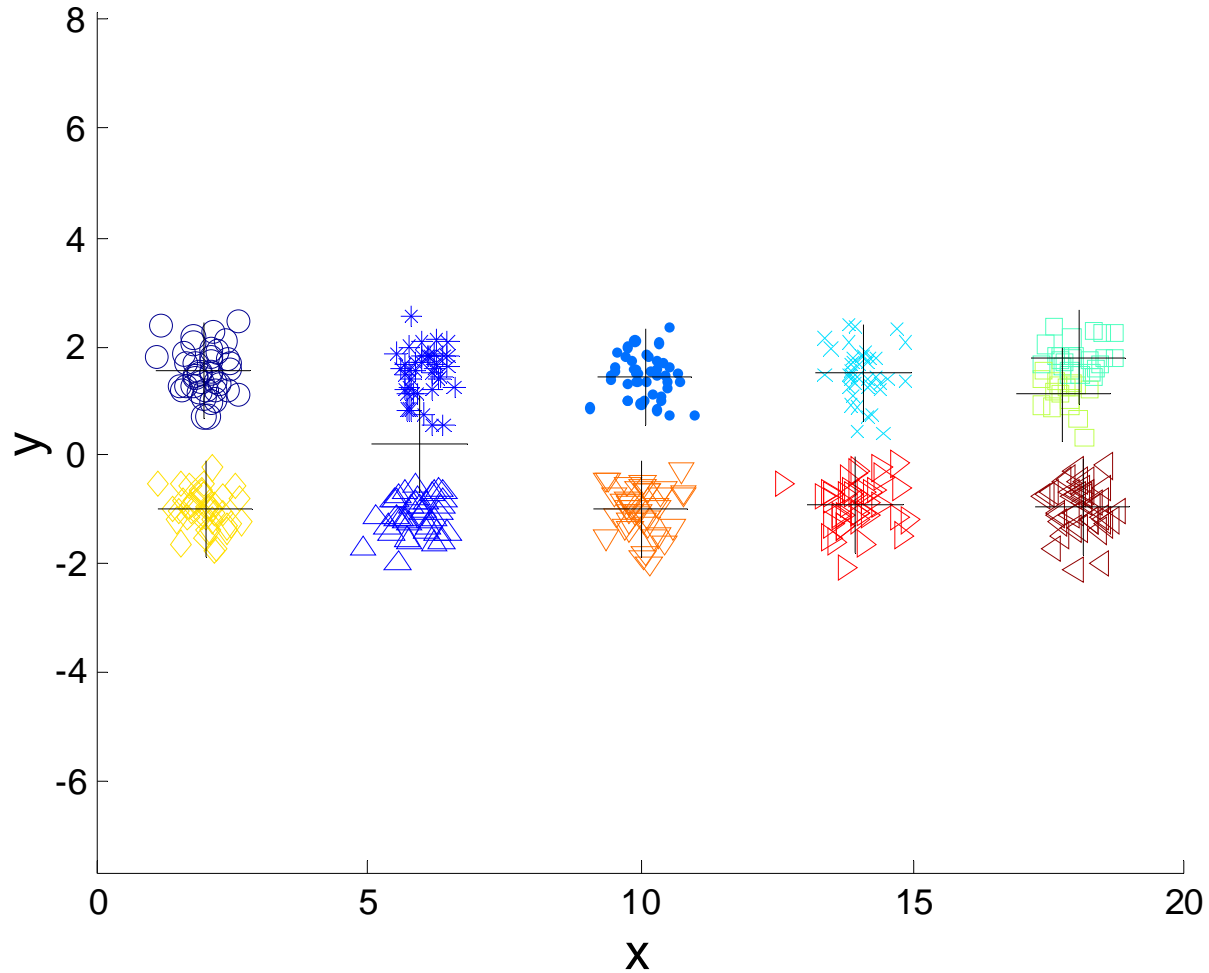
# Ten clusters example



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

# Ten clusters example

Iteration 4



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

# Solutions to initial centroids problem

---

- Multiple runs with different random initializations
- Use k-means++
  - ‘Smart’ random initialization
- 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.

# 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

---

```
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
```

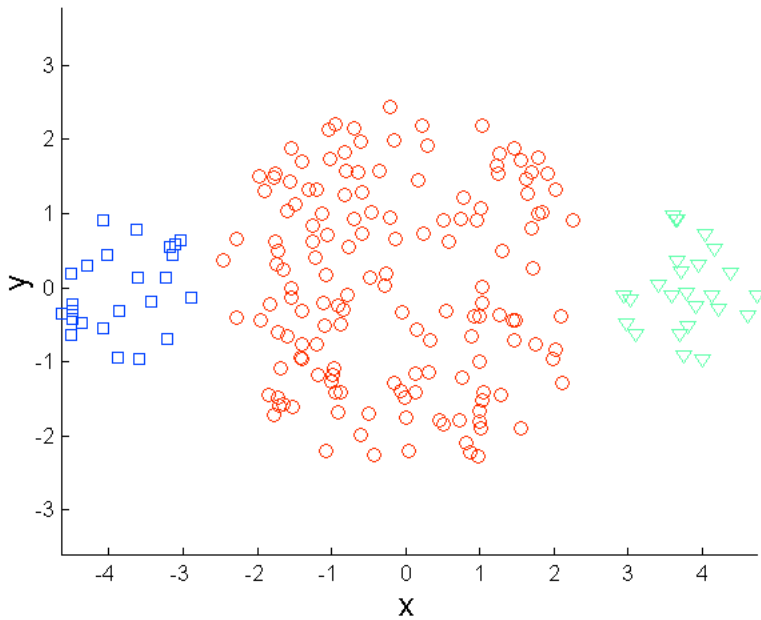
---

# Limitations of k-means

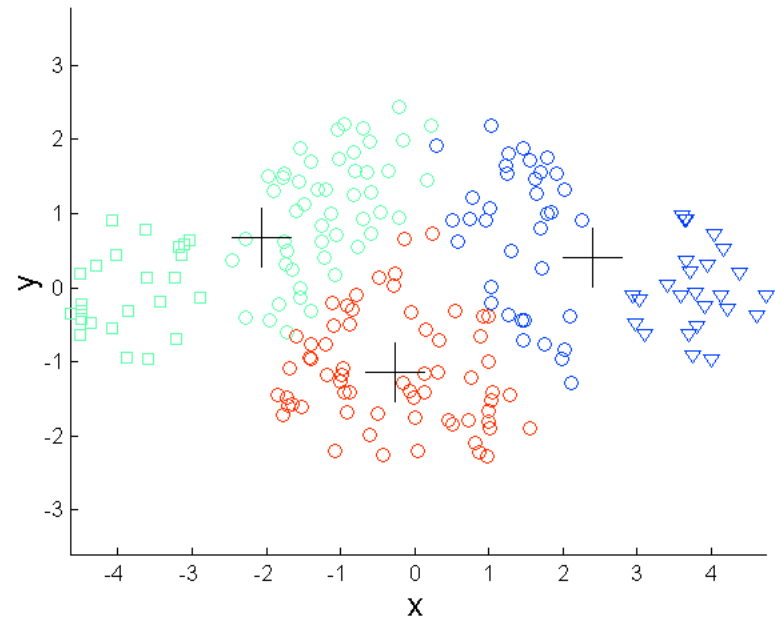
---

- k-Means optimization finds a local, not a global minimum.
- k-Means can have problems when clusters have:
  - Variable sizes
  - Variable densities
  - Non-globular shapes
- k-Means does not deal with outliers gracefully.
- k-Means will always find k clusters, no matter what actual structure of data is (even randomly distributed).

# Limitations of k-means: variable sizes



**original points**

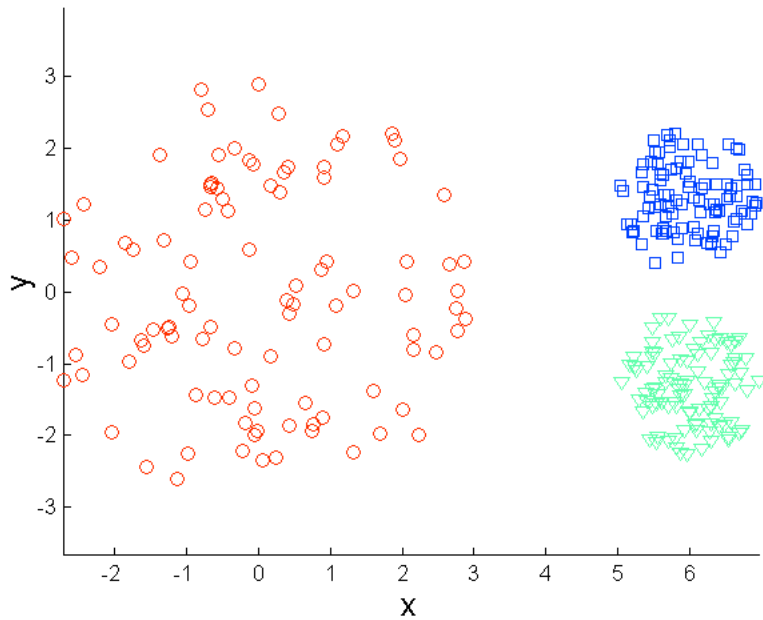


**k-means (3 clusters)**

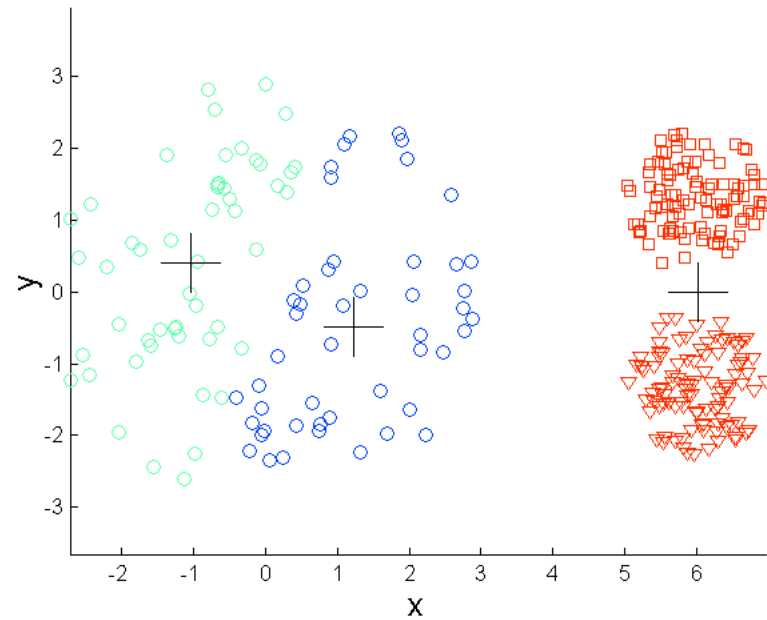


# Limitations of k-means: variable densities

---

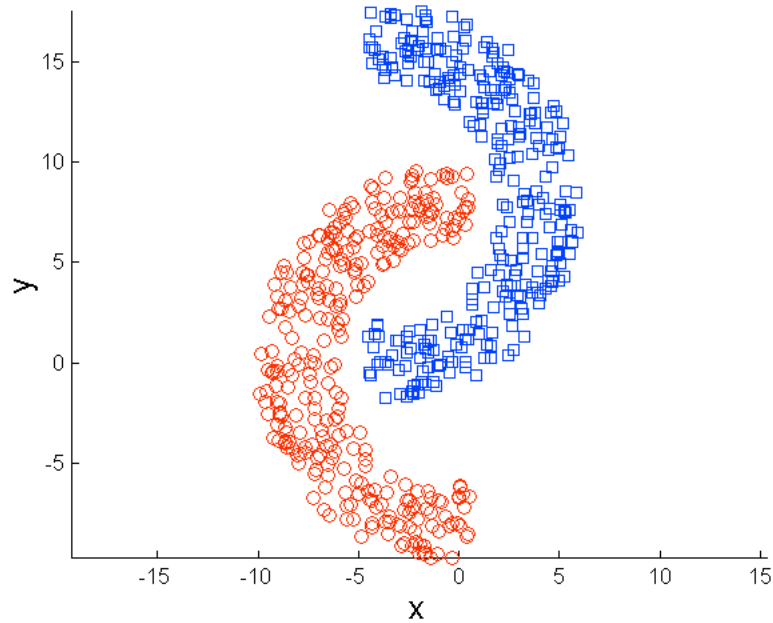


**original points**

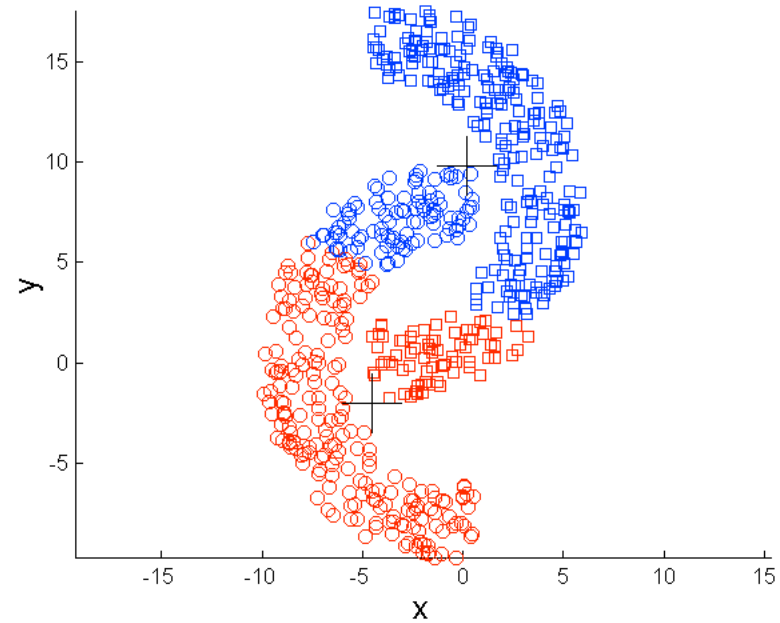


**k-means (3 clusters)**

# Limitations of k-means: non-globular shapes



**original points**



**k-means (2 clusters)**

# Demo: k-means clustering

---

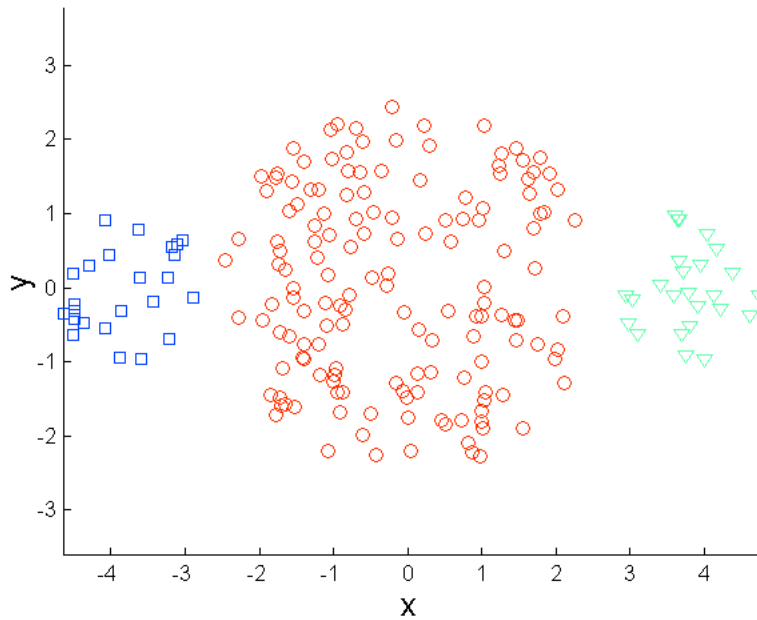
[..\videos\Visualizing k Means Algorithm.mp4](#)

on web:

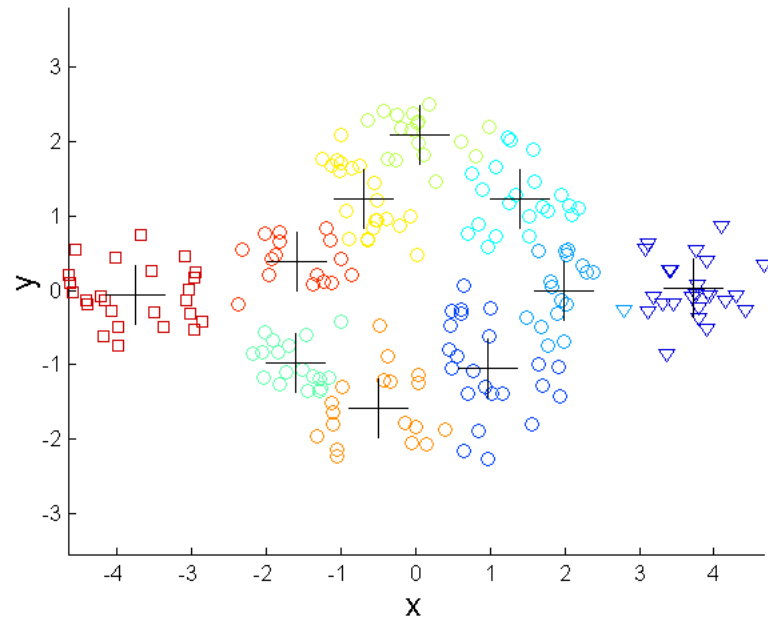
[http://www.youtube.com/watch?v=gSt4\\_kcZPxE](http://www.youtube.com/watch?v=gSt4_kcZPxE)

Note that well-defined clusters are formed even though data is uniformly and randomly distributed.

# Overcoming k-means limitations



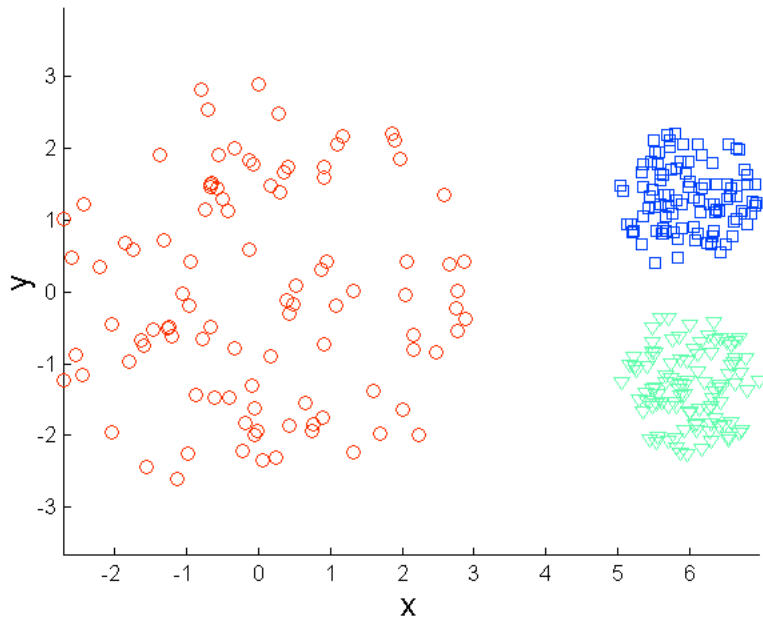
**original points**



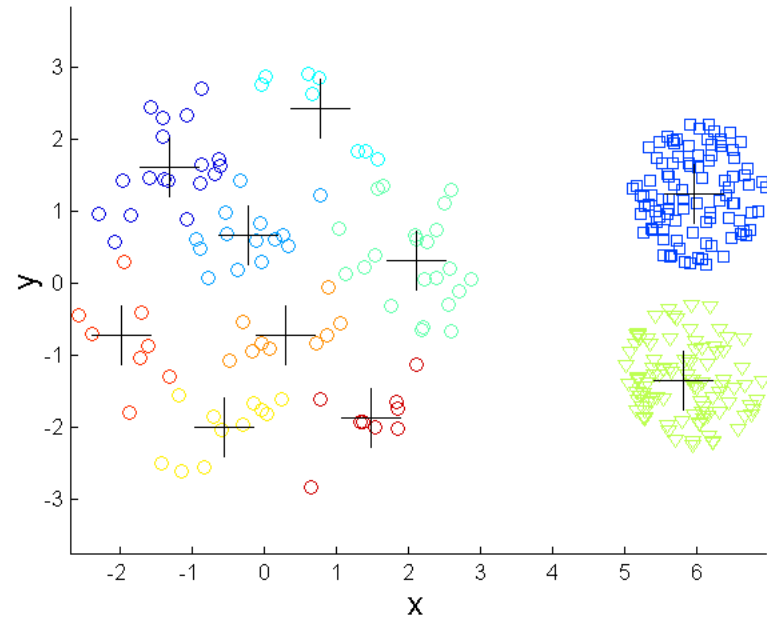
**k-means (10 clusters)**

One solution is to use many clusters. Finds pieces of natural clusters, but need to put together.

# Overcoming k-means limitations

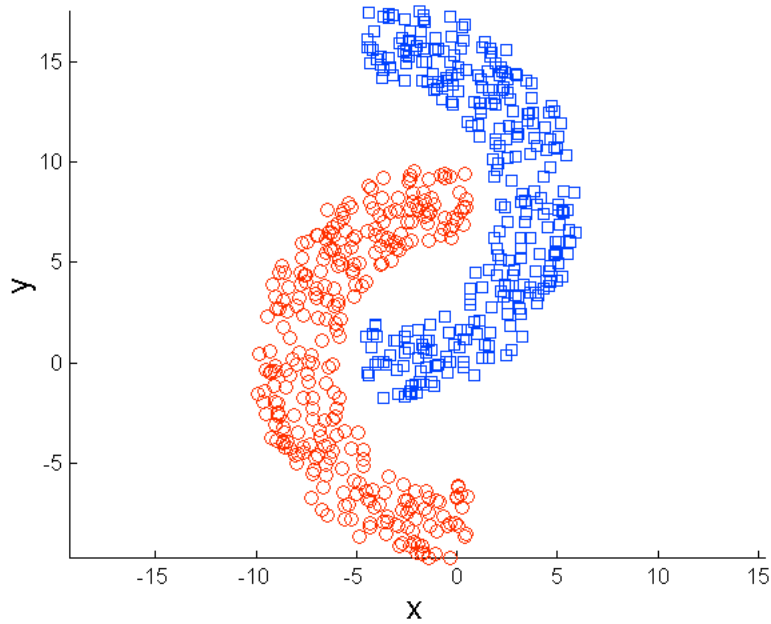


**original points**

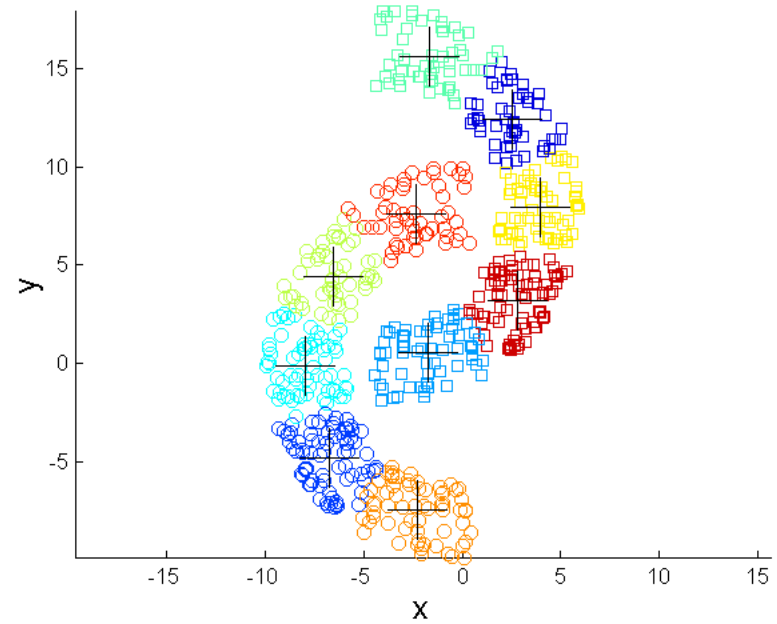


**k-means (10 clusters)**

# Overcoming k-means limitations



**original points**



**k-means (10 clusters)**

# MATLAB interlude

---

matlab\_demo\_11.m