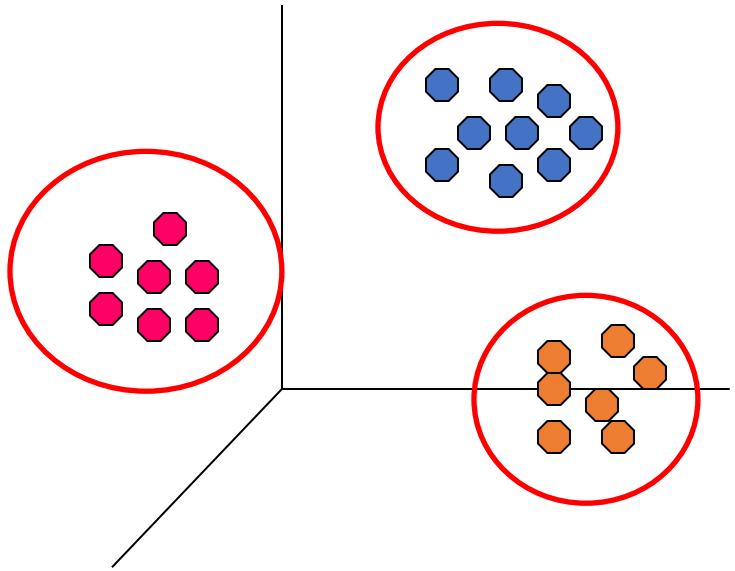
Introduction to Cluster Analysis

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



Examples of Clustering Task

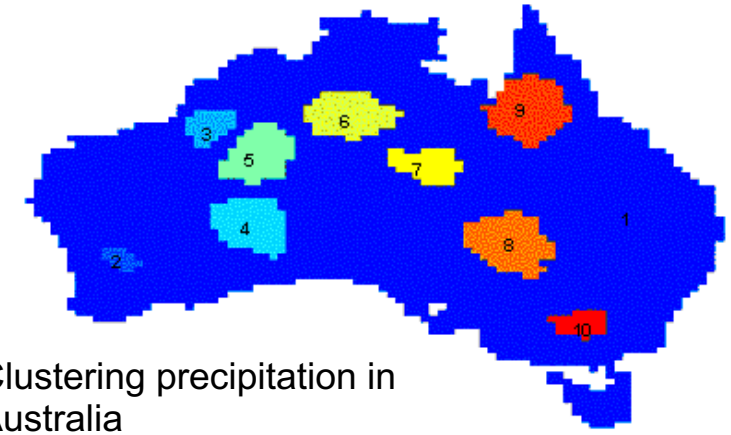
Object	Attribute set, x	Clustering Task
Document	Words in documents	Group documents based on their similar topics
Customer	Demographic and purchase information	Group together similar customers
Location	GPS trajectories of mobile phone users	Finding hot spots frequently visited by users

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

	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

Types of Clustering

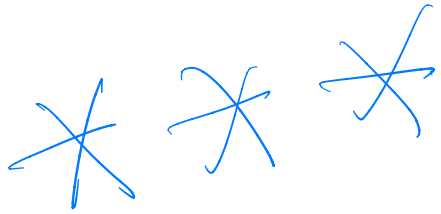
- A clustering is a set of clusters
- Types of clustering
  - Exclusive versus Overlapping versus Fuzzy
  - Exclusive: each object is assigned to only one cluster
  - Overlapping: an object may belong to more than one cluster
  - Fuzzy: every object belongs to every cluster, with a weight between 0 (absolutely doesn’t belong) and 1 (absolutely belongs)
  - Partial versus complete
  - Complete clustering: assigns every object to a cluster
  - Partial clustering: does not have to assign every object to some clusters

Prototype-based Clustering

- Each cluster is represented by a ``prototype", which is a representative point
  - All other points are assigned to clusters based on their distance to the cluster prototypes
  - User must define number of clusters/prototypes
  - Distance measure for clustering must also be specified

- Examples: k-means and its variants, k-medoid, self-organized map (SOM), etc

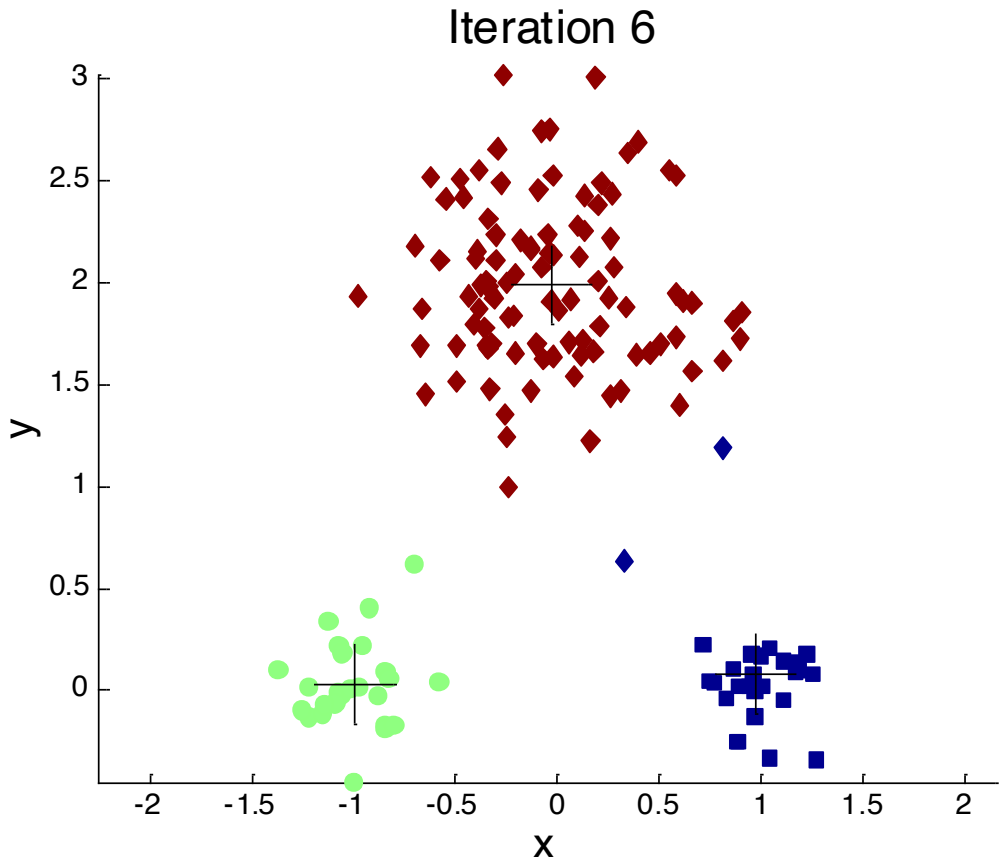
K-means Clustering



- Each cluster is associated with a centroid (center)
- Number of clusters, K, must be specified

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

Illustrating K-means



K-means as Optimization Problem

- k-means is an iterative approach for minimizing sum of squared error

$$SSE = \sum_{i=1}^{N}\sum_{j=1}^{K} w_{ij}\left\|x_i - \mu_j\right\|_2^2$$

$$\text{s.t. } \forall \text{i} : w_{ij} \in \{0,1\},\ \sum_j w_{ij} = 1$$

  - w_{ij} = 1 if x_i in cluster j; otherwise w_{ij} = 0
  - μ_j is the representative point (prototype) for cluster j

K-means as Optimization Problem

$$SSE = \sum_{i=1}^{N}\sum_{j=1}^{K} w_{ij}\left\|x_i - \mu_j\right\|_2^2$$

$$\text{s.t. } \forall \text{i} : w_{ij} \in \{0,1\}, \sum_j w_{ij} = 1$$

- When μ_j is fixed:

$$w_{ij} = \begin{cases} 1 & \text{if } j = \underset{k}{\text{argmin}} \| x_i - \mu_j \|^2 \\ 0 & \text{otherwise} \end{cases}$$

- When w_{ij} is fixed:

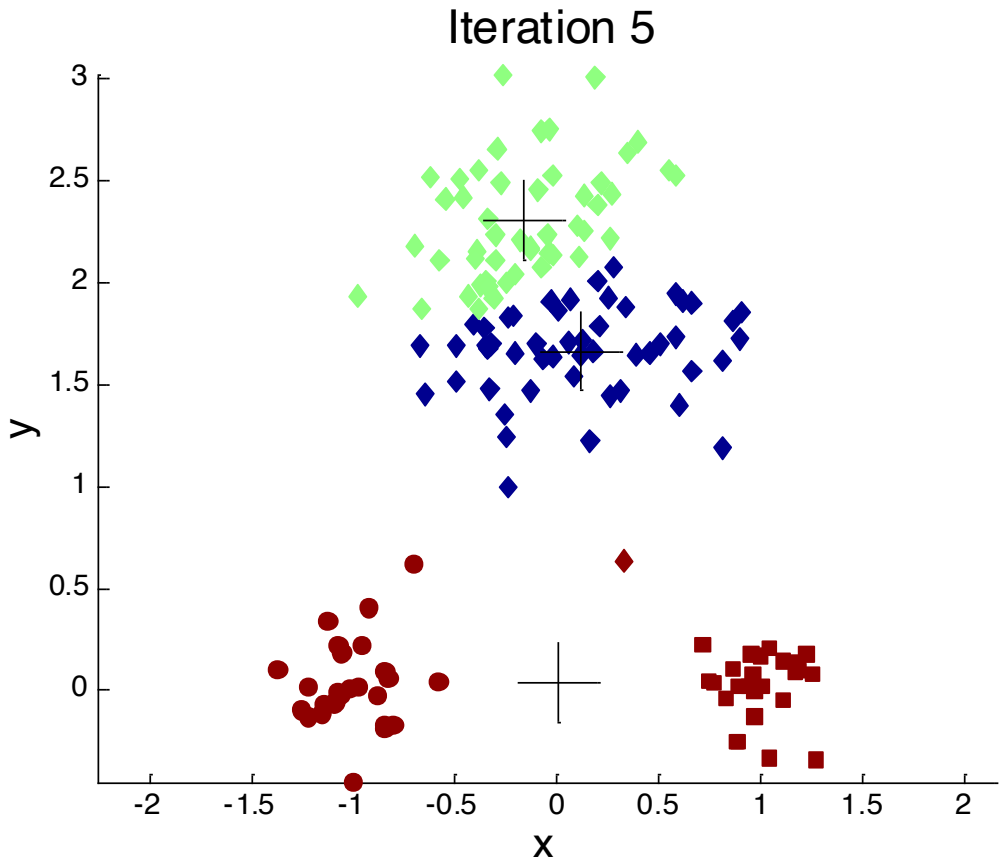
$$L = \sum_{i=1}^{N}\sum_{j=1}^{K} w_{ij}\left\|x_i - \mu_j\right\|_2^2 + \sum_{i=1}^{N}\lambda_i\left(\sum_{j=1}^{K} w_{ij} - 1\right)$$

$$\frac{\partial L}{\partial \mu_k} = -2\sum_{i=1}^{N} w_{ik}\left(x_i - \mu_k\right) = 0 \Rightarrow \mu_k = \frac{\sum_{i=1}^{N} w_{ik} x_i}{\sum_{i=1}^{N} w_{ik}}$$

K-means Clustering – Details

- Time complexity
  - Complexity is O(n * K * I * d)
  - n = number of points, K = number of clusters,
    I = number of iterations, d = number of attributes
- Initial centroids are often chosen randomly
  - Clusters produced may vary from one run to another
- Data should be normalized/standardized to ensure each attribute contributes equally to the distance function

Importance of Choosing Initial Centroids …



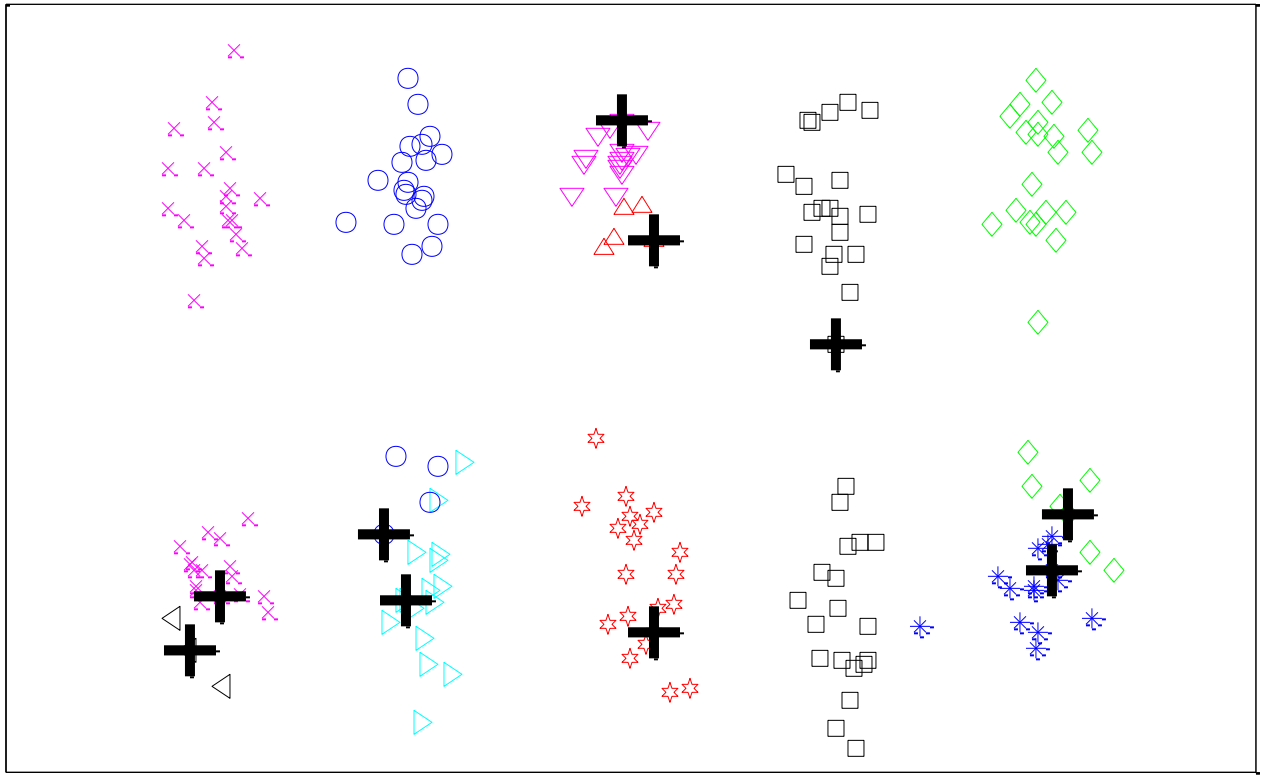
Problems with Selecting Initial Points

- If there are K ‘real’ clusters then the chance of selecting one centroid from each cluster is small.
  - If clusters are the same size, n, then

$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{\binom{n}{1}^k}{\binom{nk}{k}} = \frac{n^k k!\,(nk-k)!}{(nk)!}$$

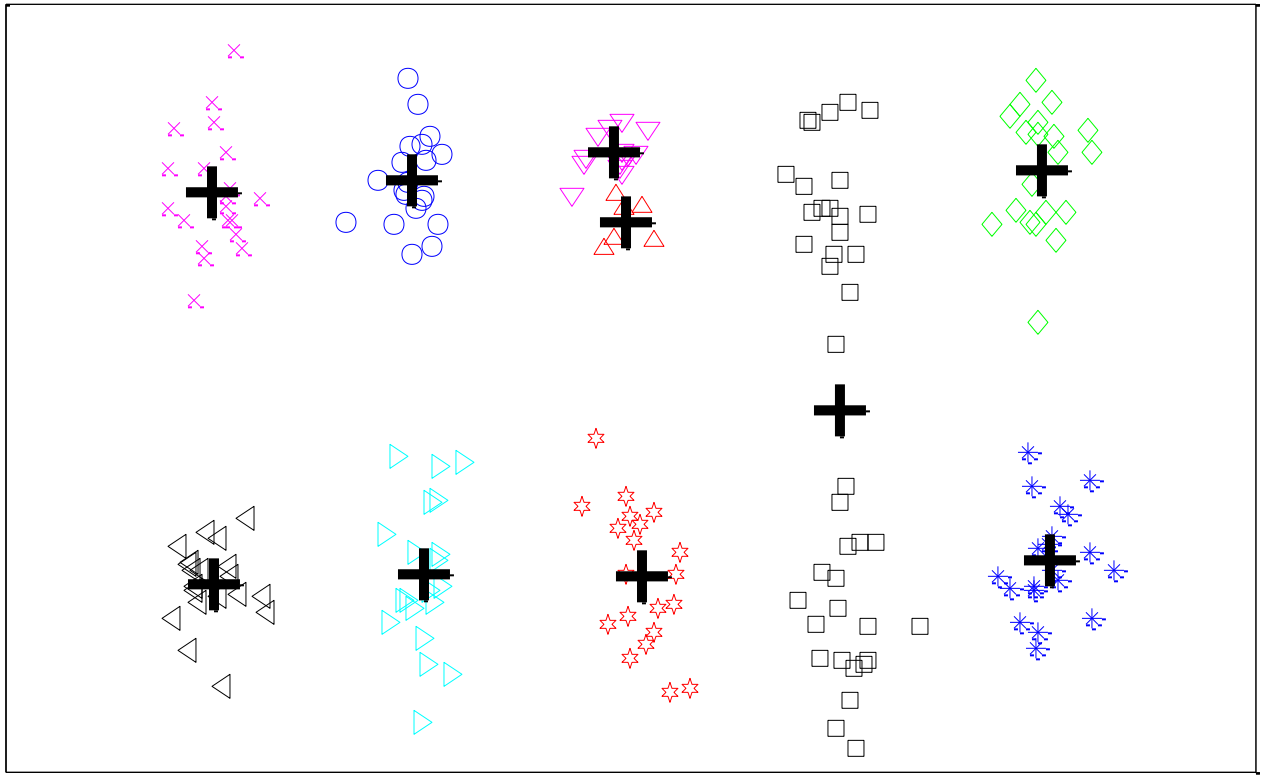
  - For example, if K = 10, n = 10 then probability = 5.8×10^{-12}
  - Sometimes the initial centroids will readjust themselves in ‘right’ way, and sometimes they don’t
  - Consider an example of five pairs of clusters

10 Clusters Example



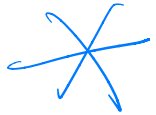
Initial Centroids

10 Clusters Example



Final Clustering

K-means++

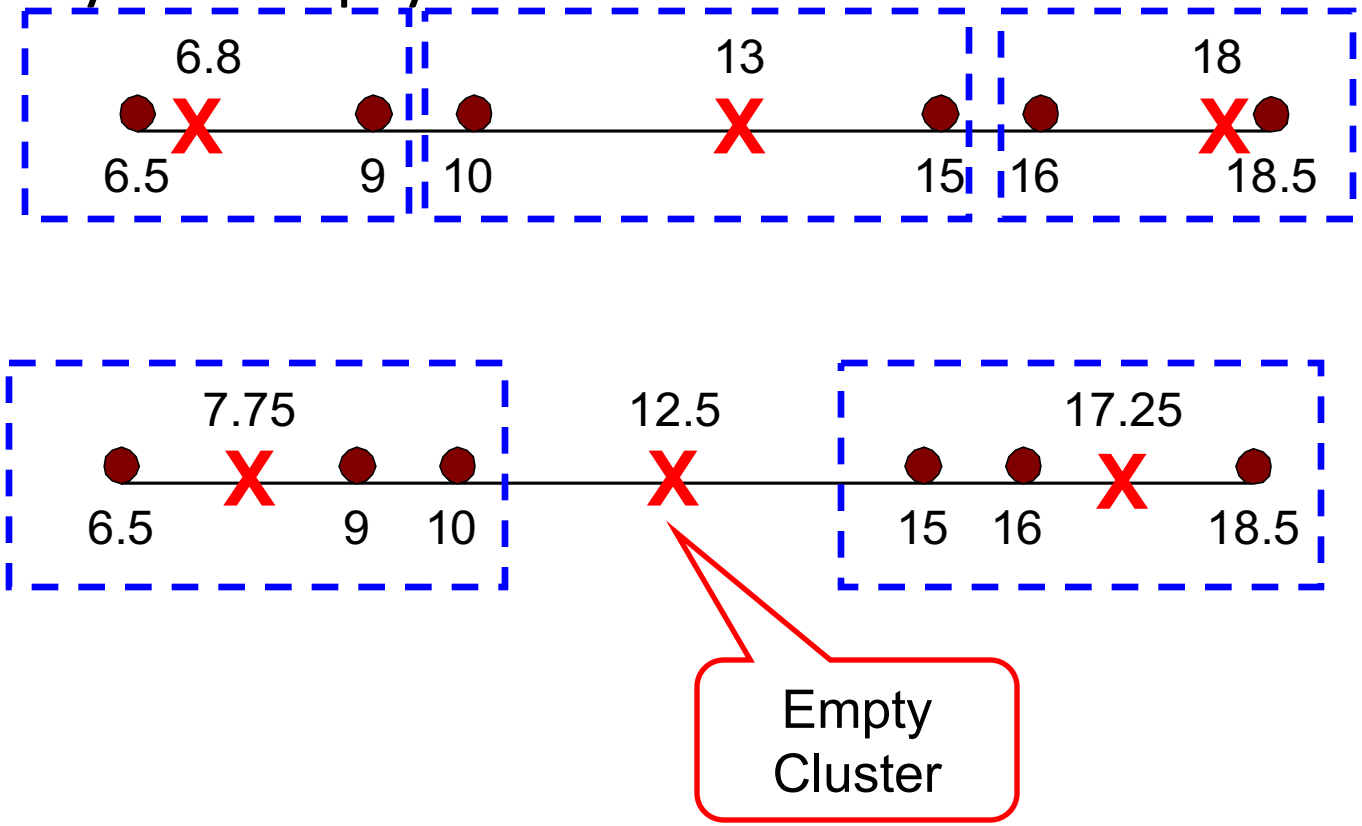


- This approach can be slower than random initialization but consistently produces better results in terms of SSE

1. Initialization: $C = \varnothing$
2. Randomly select a data point x_0 to be the first centroid $C = C \cup \{x_0\}$
3. Repeat until $|C| = k$
   - For each point, $x_i \notin C$, find the minimum squared distance to any currently selected centroid in C:
     $$d(x_i, C)^2 = \min_{x_j \in C} d(x_i, x_j)^2$$
   - Select a new centroid x_k by randomly choosing a point with probability proportional to $d(x_k, C)^2 / \sum_x d(x, C)^2$
   - $C = C \cup \{x_k\}$

Empty Clusters

- K-means can yield empty clusters



Handling Empty Clusters

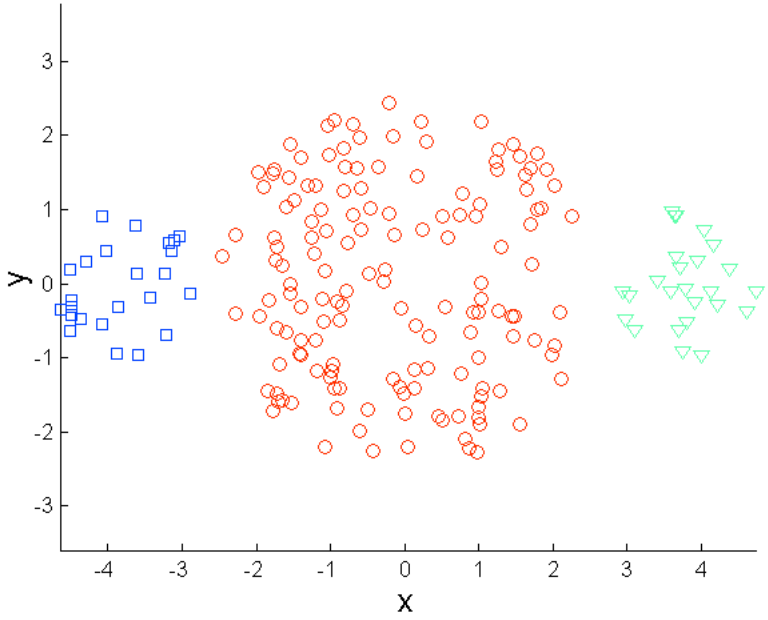
- Choose a new centroid to replace the empty cluster

- Several strategies
  - Choose the point that contributes most to SSE
    - this corresponds to the point that is farthest away from any of the current centroids
  - Choose a point from the cluster with the highest SSE
    - This will split the cluster and reduces the overall SSE

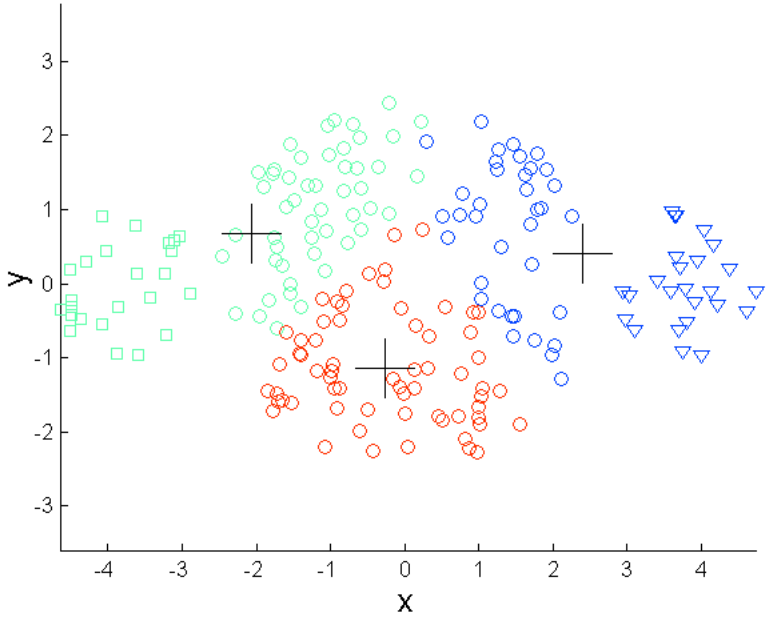
Limitations of K-means

- 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

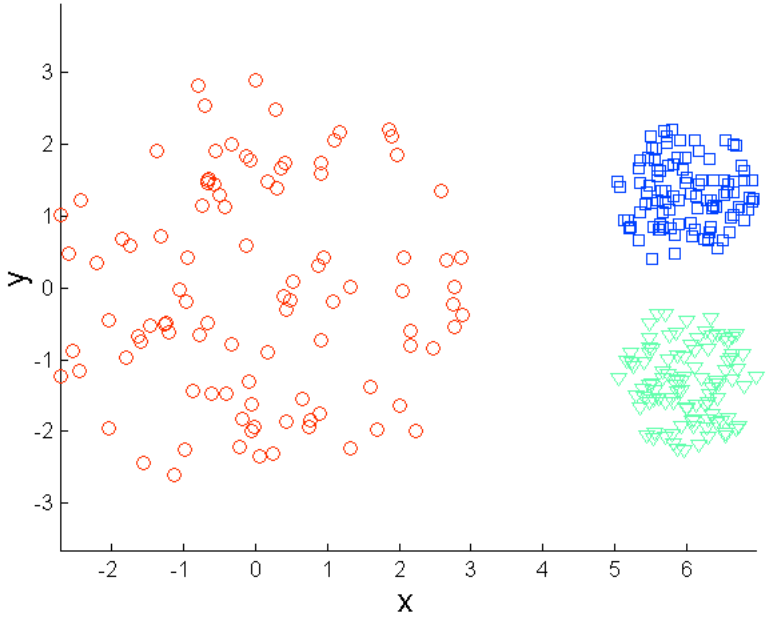


Original Points

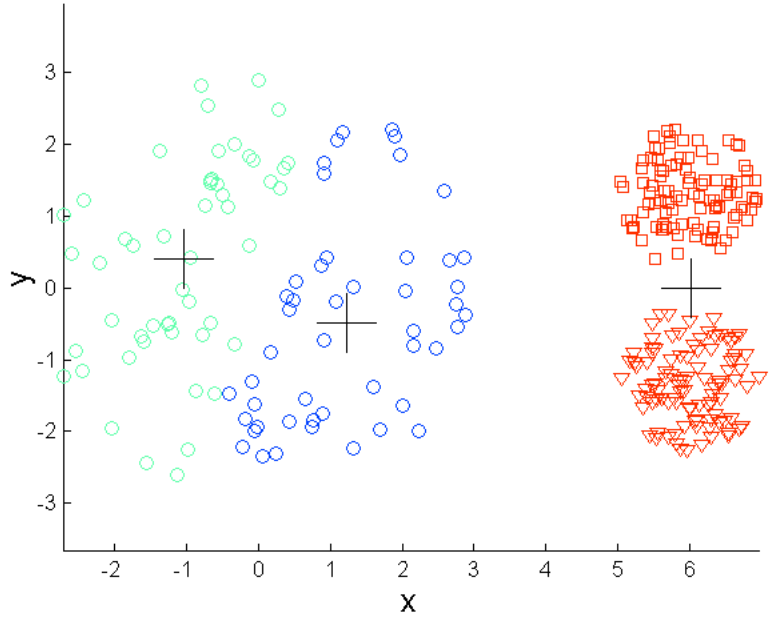


K-means (3 Clusters)

Limitations of K-means: Differing Density

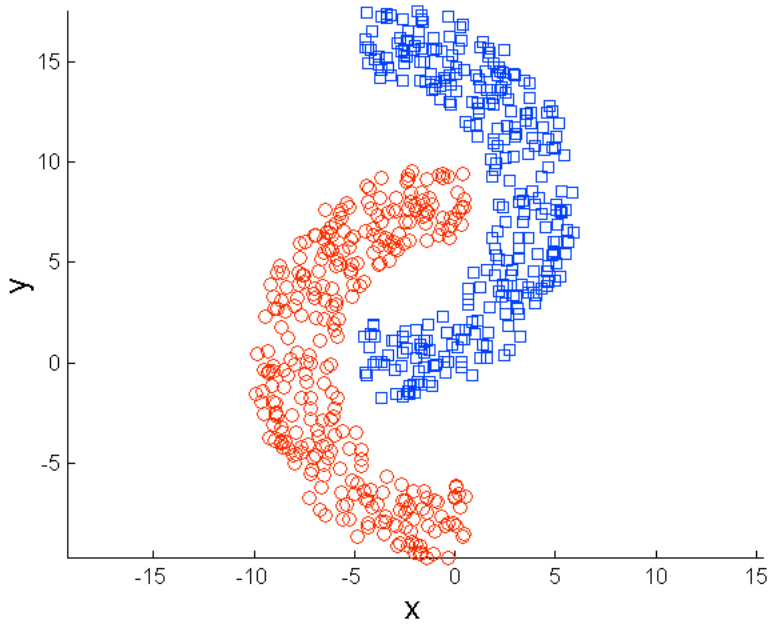


Original Points

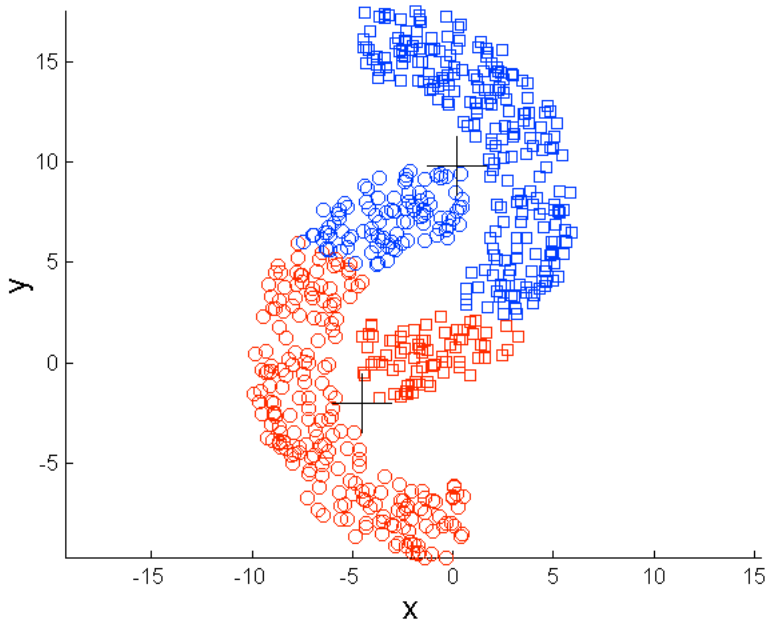


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes



Original Points



K-means (2 Clusters)

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

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

Bisecting K-means Example