Master Degree in Artificial Intelligence for Science and Technology

Cluster Analysis: K-means Clustering

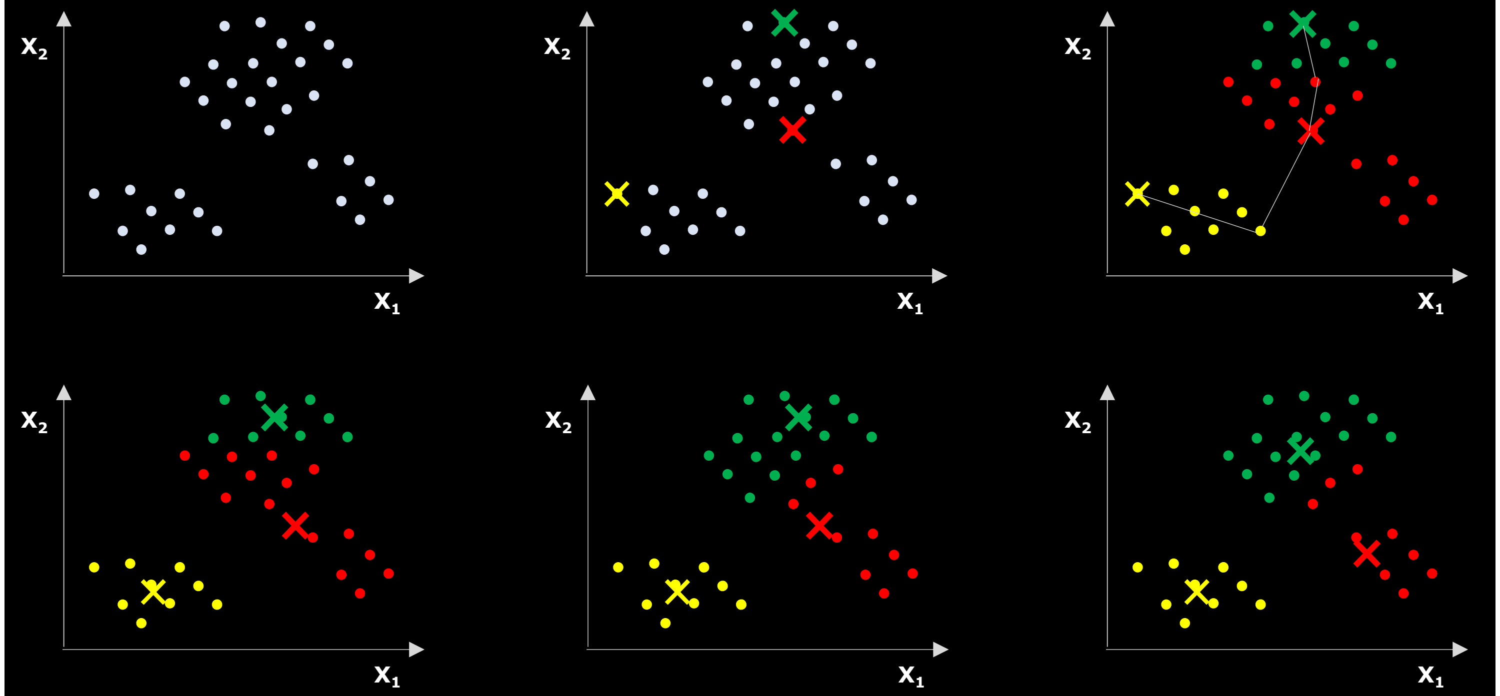
Fabio Stella
Department of Informatics, Systems and Communication
University of Milan-Bicocca
fabio.stella@unimib.it

OUTLOOK

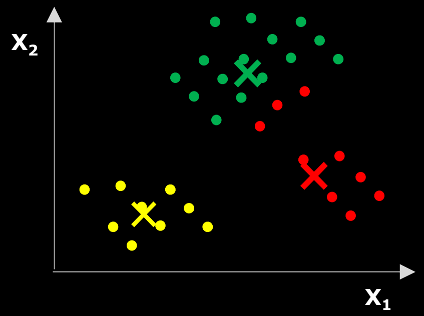
- K-means learning algorithm
- Examples
- Details, complexity, …
- Objective function
- Choosing initial centroids
- Limitations and how to overcome them

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

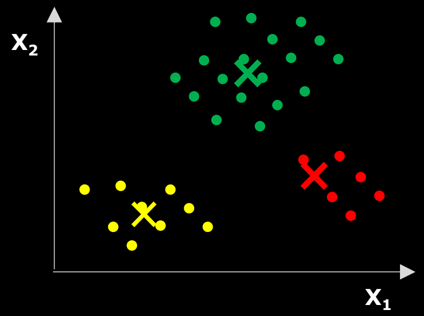
1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change



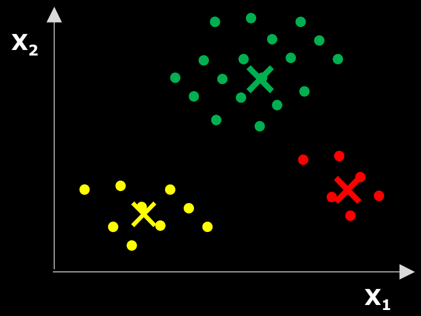
x_2
x_1
x_2
x_1
x_2
x_1
x_2
x_1
x_2
x_1
x_2
x_1



x_2
x_1



x_2
x_1



x_2
x_1

- Simple iterative algorithm.
  - — choose initial centroids;
  - — repeat {assign each point to a nearest centroid; re-compute cluster centroids}
  - — until centroids stop changing.
- 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, but other definitions are possible, i.e., medoid, ...
- K-means converges for common proximity measures with appropriately defined centroid
- 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

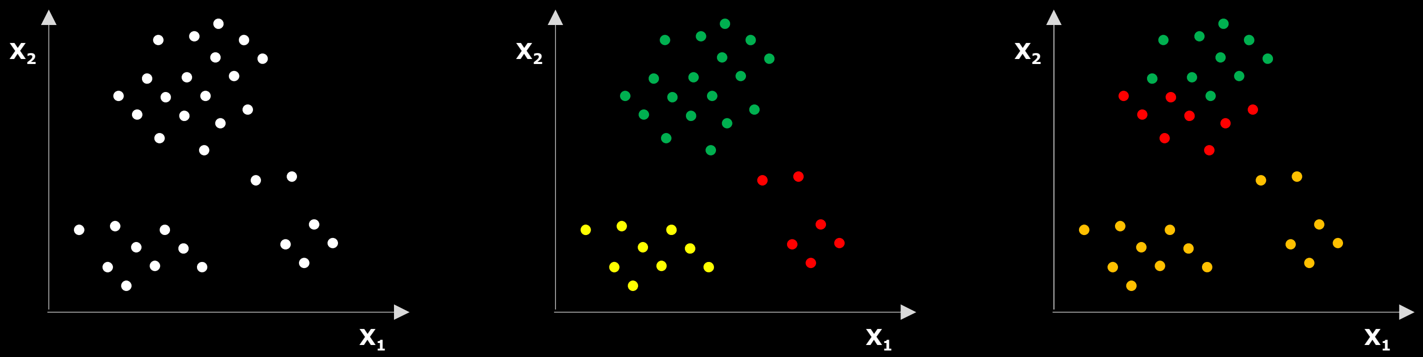
A common **OBJECTIVE FUNCTION** (used with Euclidean distance measure) is the **SUM OF SQUARED ERROR** (SSE)

- for each point, the error is the distance to the nearest cluster center
- to get SSE, we square these errors and sum them

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x) = \sum_{i=1}^{K} \sum_{x \in C_i} \sum_{j=1}^{n} \left(m_{ij} - x_{ij}\right)^2$$

- x is a data point in cluster C_i and m_i is the centroid (mean) for cluster C_i
- SSE improves in each iteration of K-means until it reaches a local or global minima.

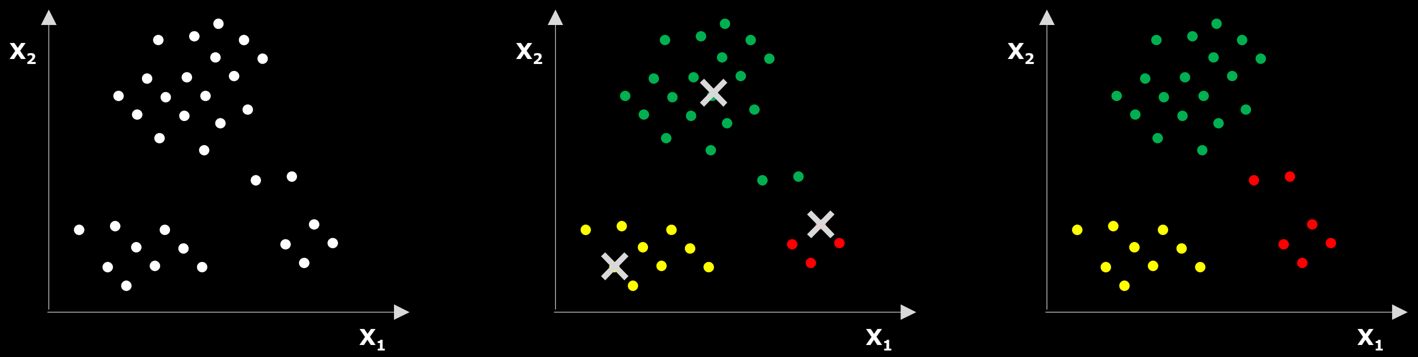
Two different k-means clusterings



original data | optimal clustering | sub-optimal clustering

The selection of initial centers (centroids) can lead to different clusterings!!!

Choosing initial centers (centroids).



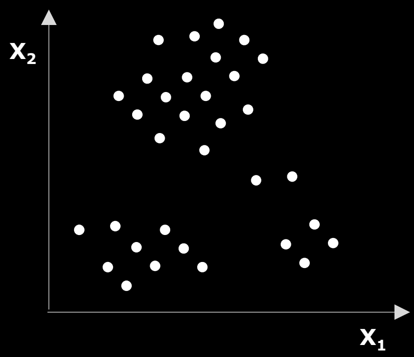
original data

one centroid per cluster

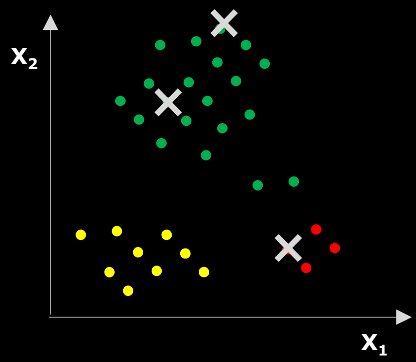
optimal clustering

An lucky selection!!!

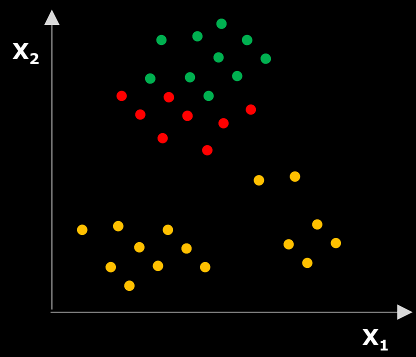
Choosing initial centers (centroids).



original data



one centroid per cluster



sub-optimal clustering

An unlucky selection!!!

PROBLEMS WITH SELECTING INITIAL CENTERS

- 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

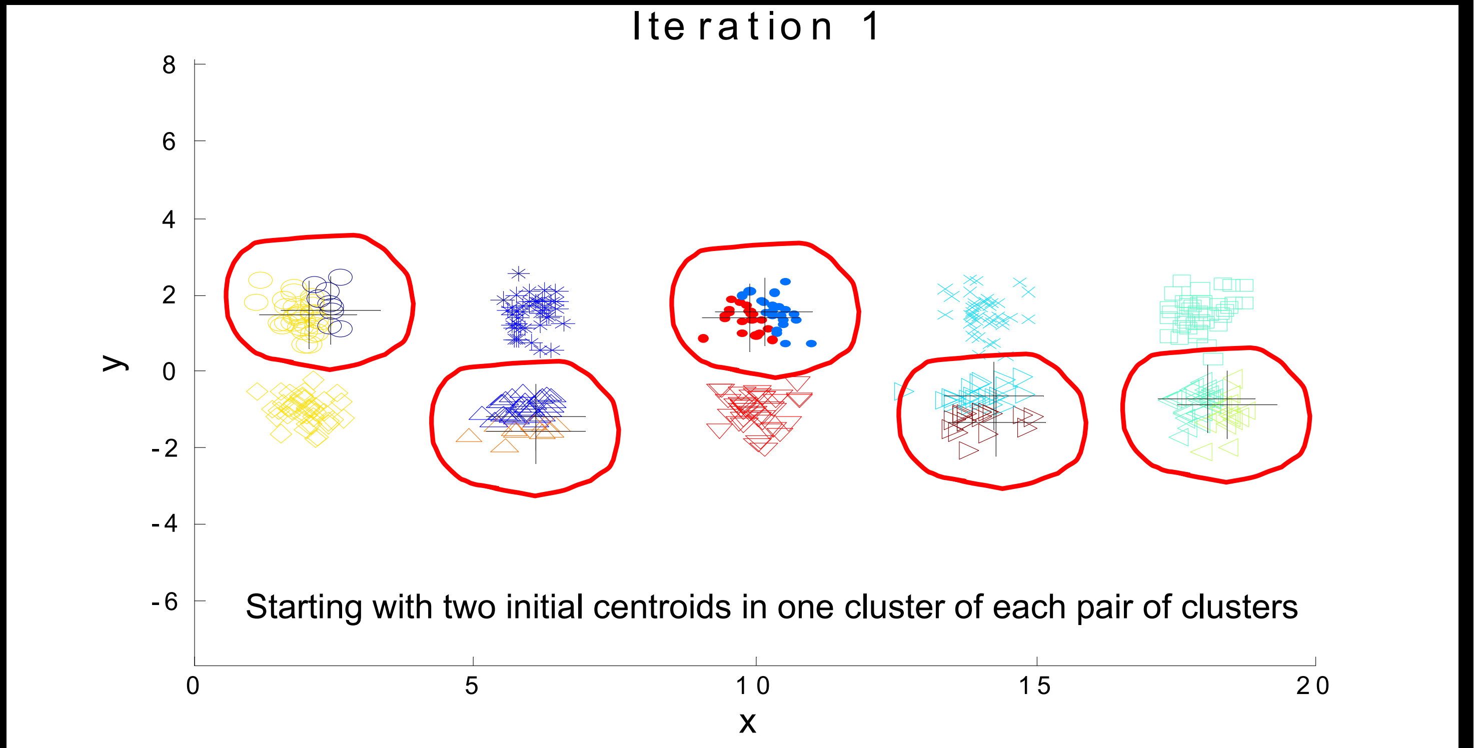
— if clusters are the same size, n, then

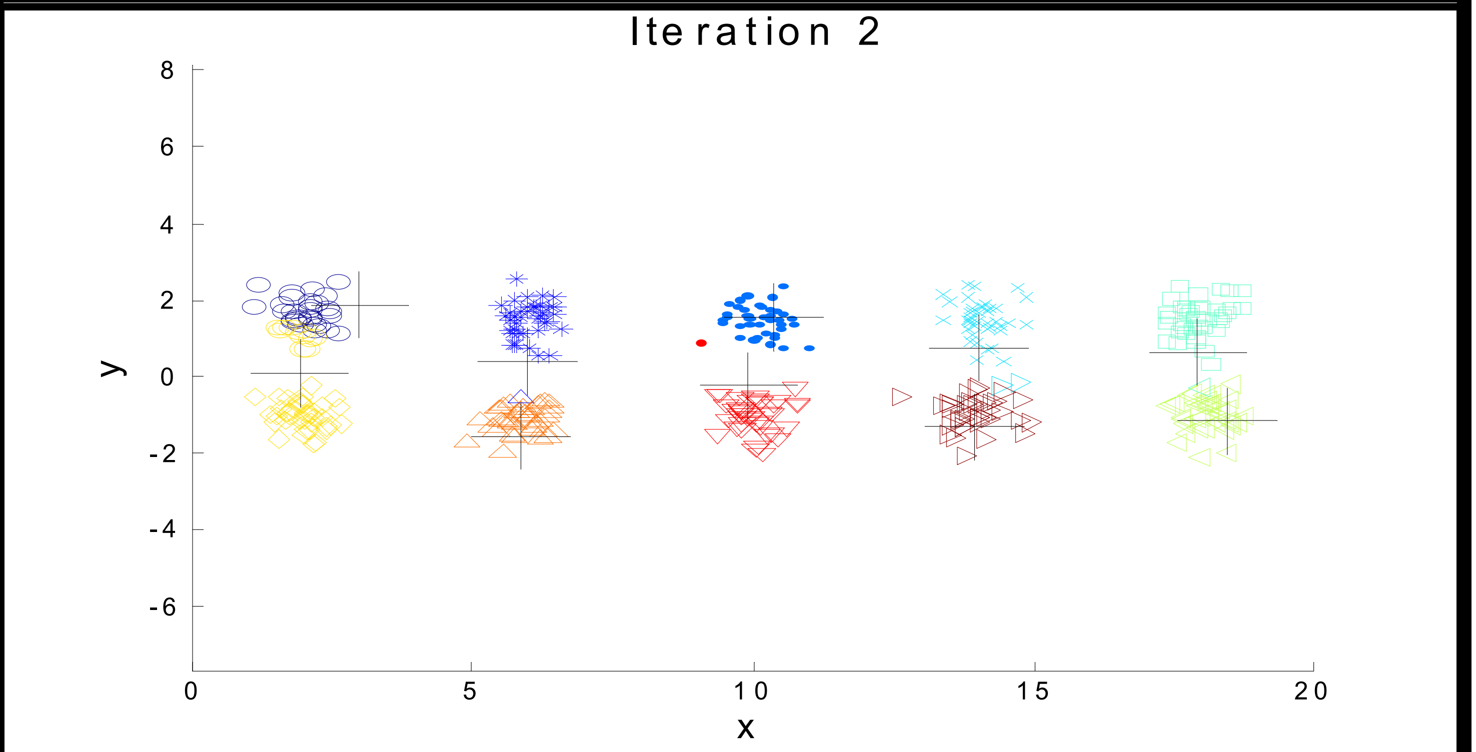
$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K! n^K}{(Kn)^K} = \frac{K!}{K^K}$$

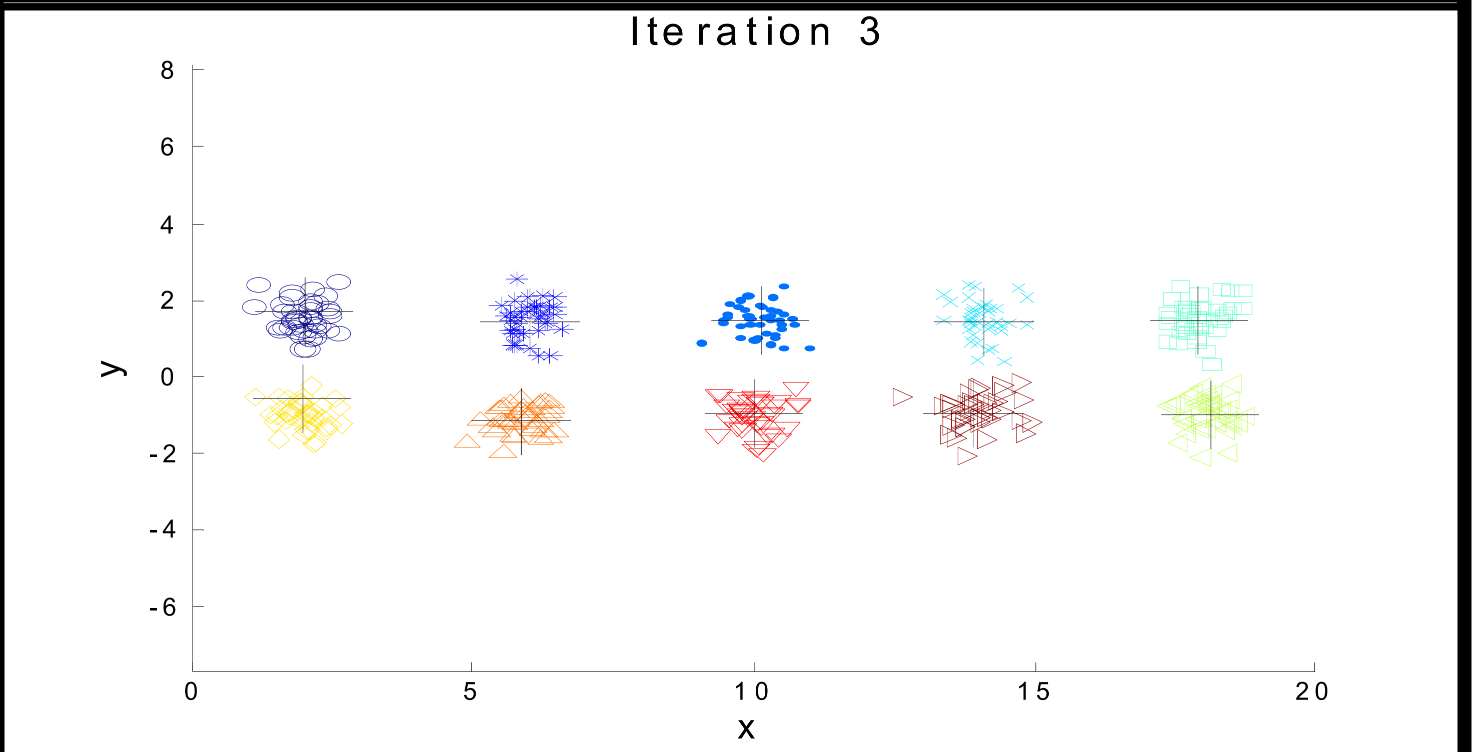
— for example, if $K = 10$, then probability = $10!/10^{10}$ = 0.00036

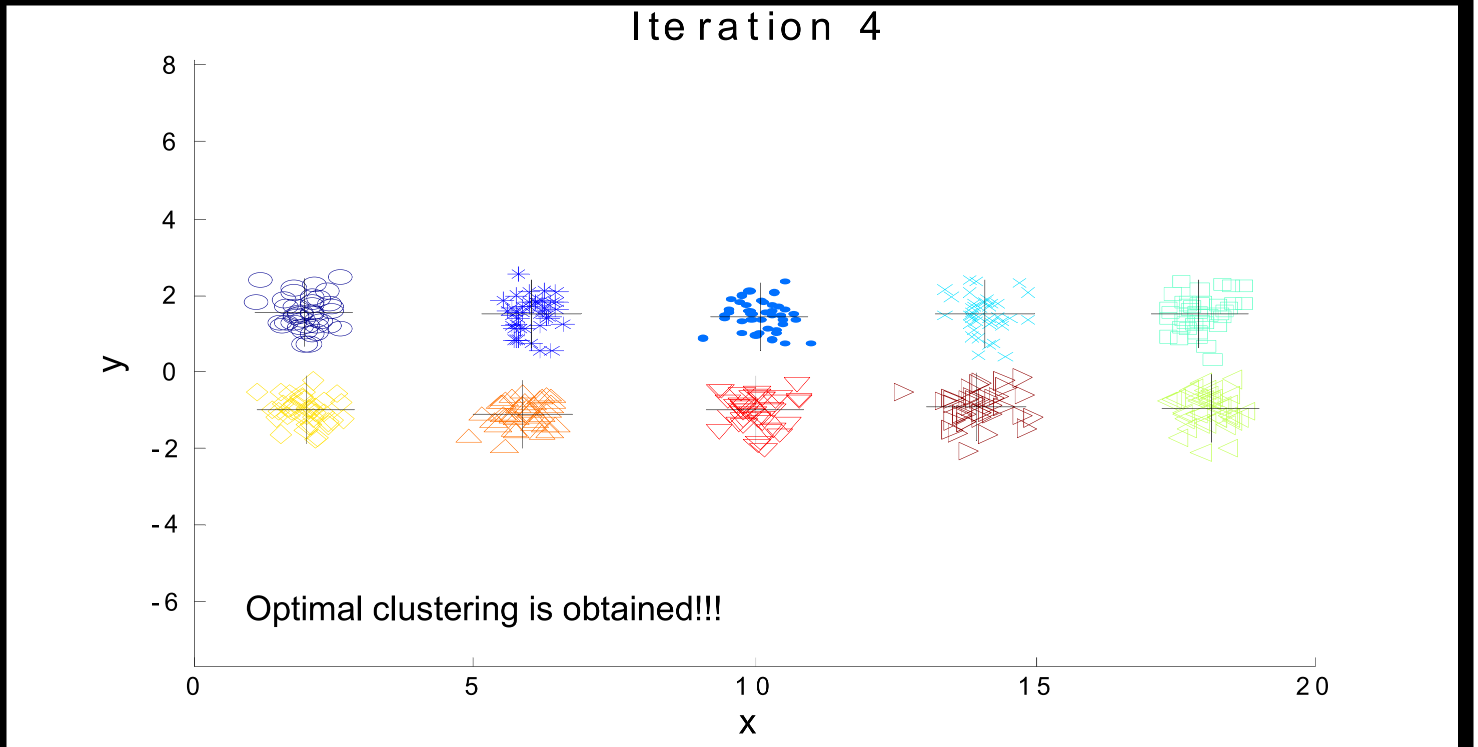
— sometimes the initial centroids will readjust themselves in 'right' way, and sometimes they don't

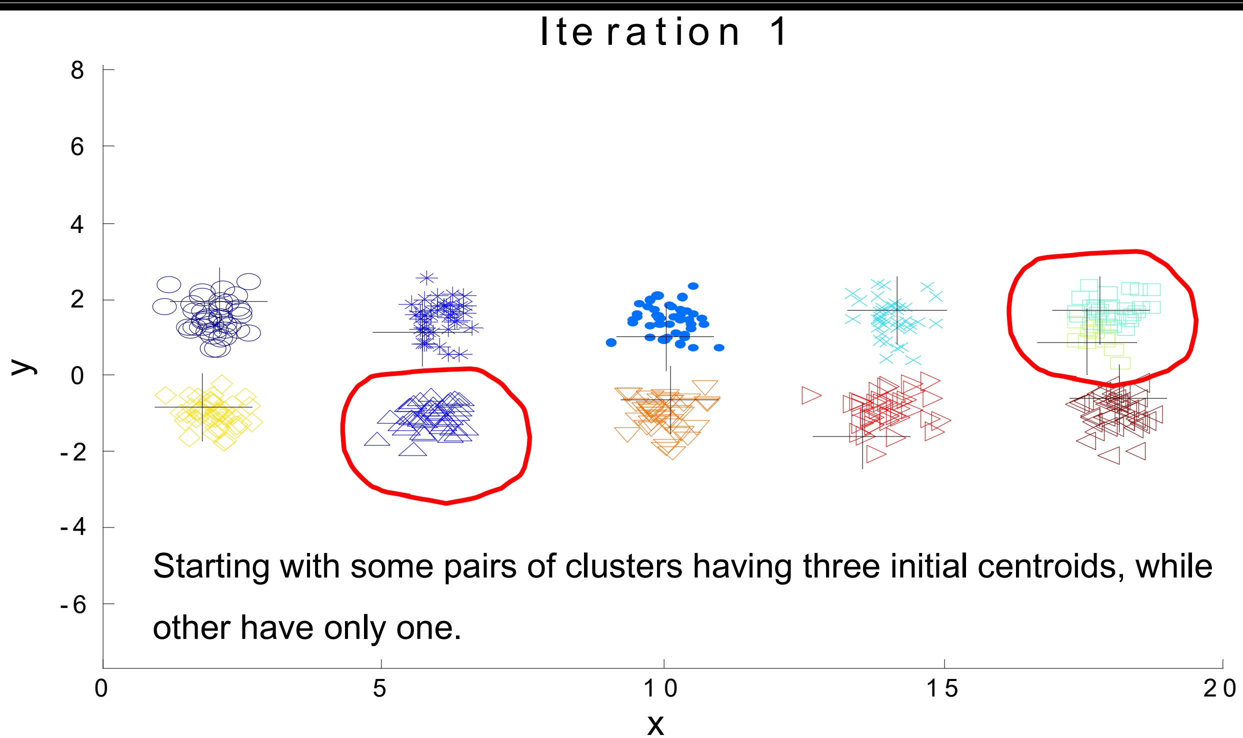
— consider an example of five pairs of clusters


Iteration 1
8
6
4
2
0
-2
-4
-6
y
0
5
10
15
20
x
Starting with two initial centroids in one cluster of each pair of clusters

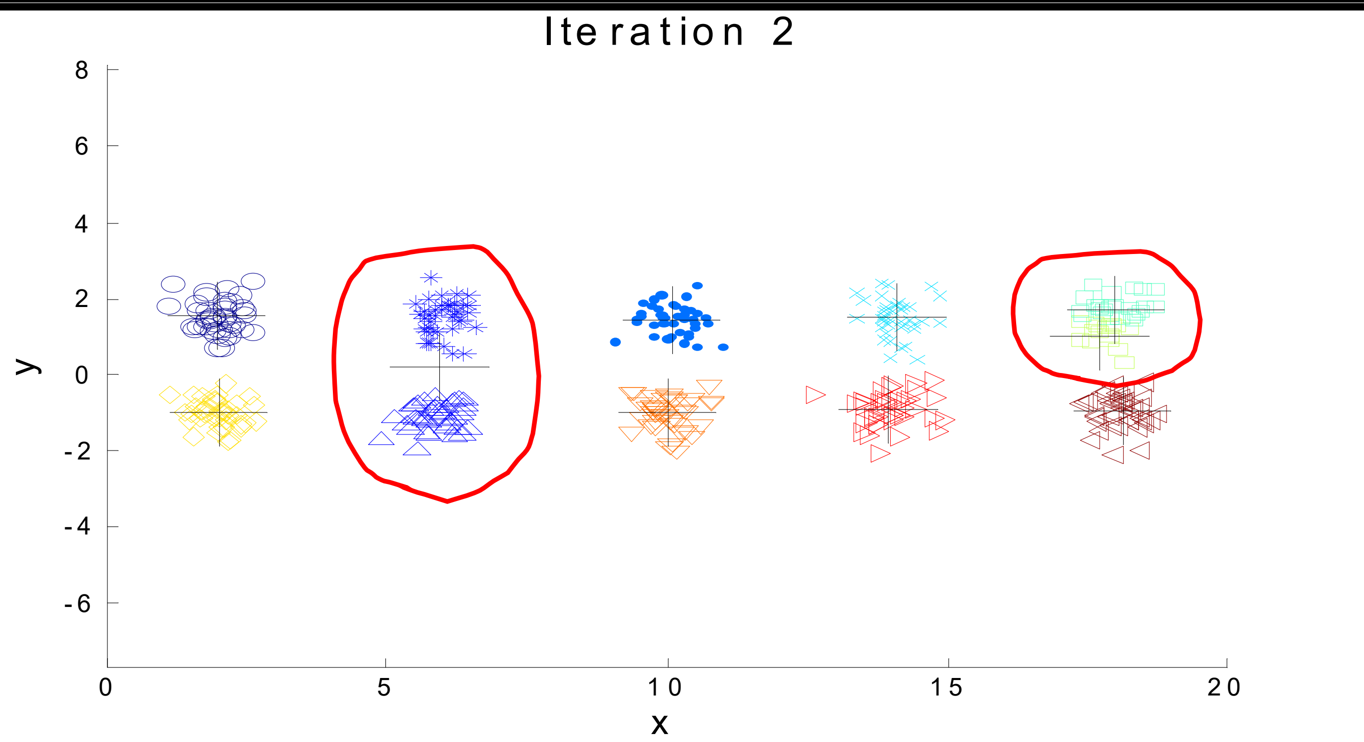

Iteration 2
8
6
4
2
0
-2
-4
-6
y
0
5
10
15
20
x

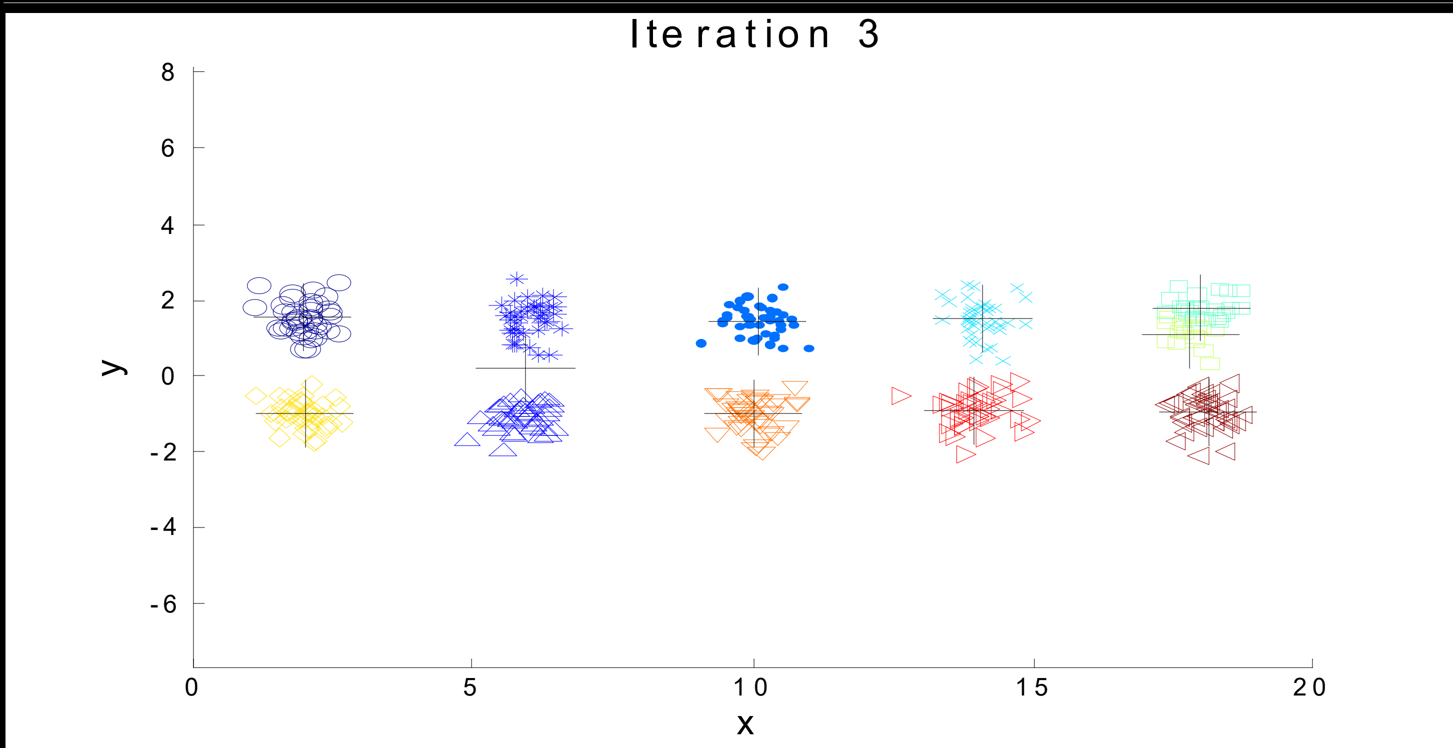

Iteration 3
8
6
4
2
0
-2
-4
-6
y
0
5
10
15
20
x

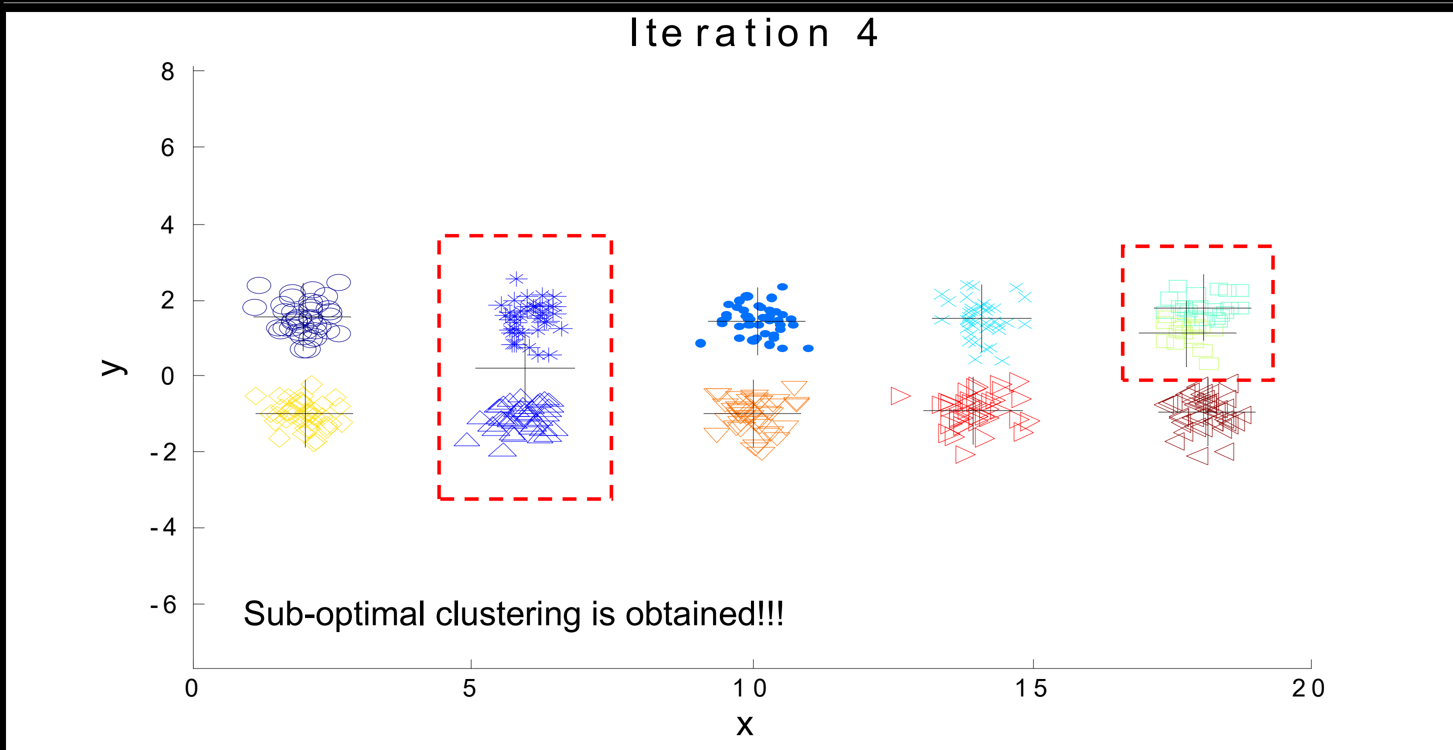

Iteration 4
8
6
4
2
0
-2
-4
-6
y
0
5
10
15
20
x
Optimal clustering is obtained!!!



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


Iteration 2
8
6
4
2
0
-2
-4
-6
y
0
5
10
15
20
x


Iteration 3
8
6
4
2
0
-2
-4
-6
y
0
5
10
15
20
x


Iteration 4
8
6
4
2
0
-2
-4
-6
y
0
5
10
15
20
x
Sub-optimal clustering is obtained!!!

SOLUTIONS TO INITIAL CENTROIDS PROBLEM

- Multiple runs
  - — helps, but probability is not on your side
- Use some strategy to select the K initial centroids and then select among these initial centroids
  - — select most widely separated
  - — K-means++ is a robust way of doing this selection
  - — use hierarchical clustering to determine initial centroids
- Bisecting K-means
  - — not as susceptible to initialization issues

K-MEANS++

- This approach can be slower than random initialization, but very consistently produces better results in terms of SSE
  — the k-means++ algorithm guarantees an approximation ratio O(log K) in expectation, where K is the number of centers

- To select a set of initial centroids, $\mathbf{C}$, perform the following

1. Select an initial point at random to be the first centroid
2. For $K - 1$ steps
3. For each of the N points, x_i, $1 \le i \le N$, find the minimum squared distance to the currently selected centroids, $C_1, \dots, C_K$, $1 \le j < K$, i.e., $\min_j \mathrm{d}^2(C_j, x_i)$
4. Randomly select a new centroid by choosing a point with probability proportional to $\frac{\min_j \mathrm{d}^2(C_j, x_i)}{\sum_i \min_j \mathrm{d}^2(C_j, x_i)}$
5. End For

BISECTING K-MEANS

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

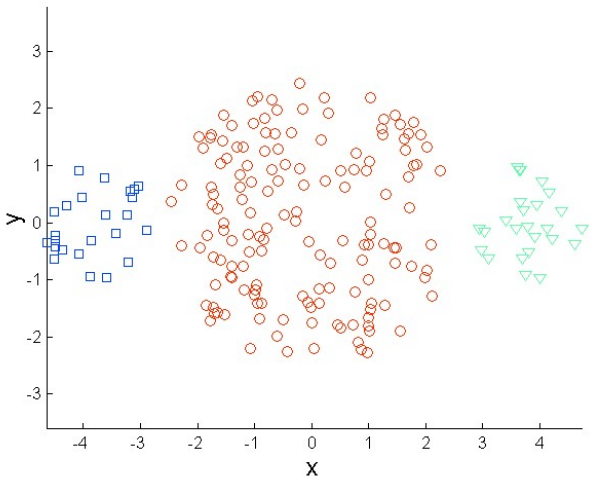
Initialize the list of clusters to contain the cluster containing all points.
repeat
  Select a cluster from the list of clusters
  for $i = 1$ to $number\_of\_iterations$ **do**
    Bisect the selected cluster using basic K-means
  end for
  Add the two clusters from the bisection with the lowest SSE to the list of clusters.
until Until the list of clusters contains K clusters

CLUTO: https://mybiosoftware.com/cluto-2-1-2a-gcluto-1-0-software-clustering-high-dimensional-datasets.html

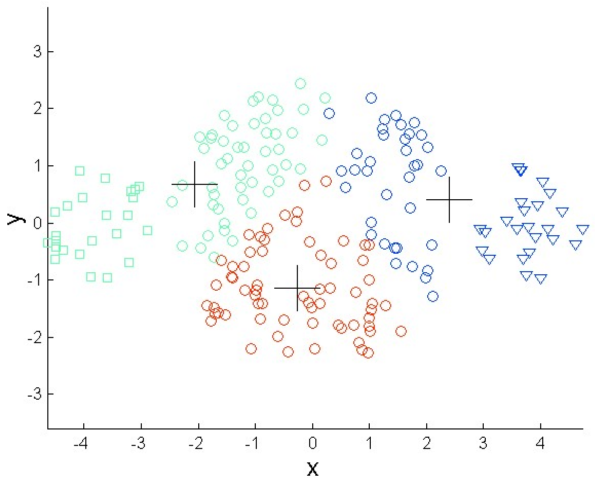
LIMITATIONS OF K-MEANS

- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes
- K-means has problems when the data contains outliers.
  - one possible solution is to remove outliers before clustering

LIMITATIONS OF K-MEANS (DIFFERENT SIZES)

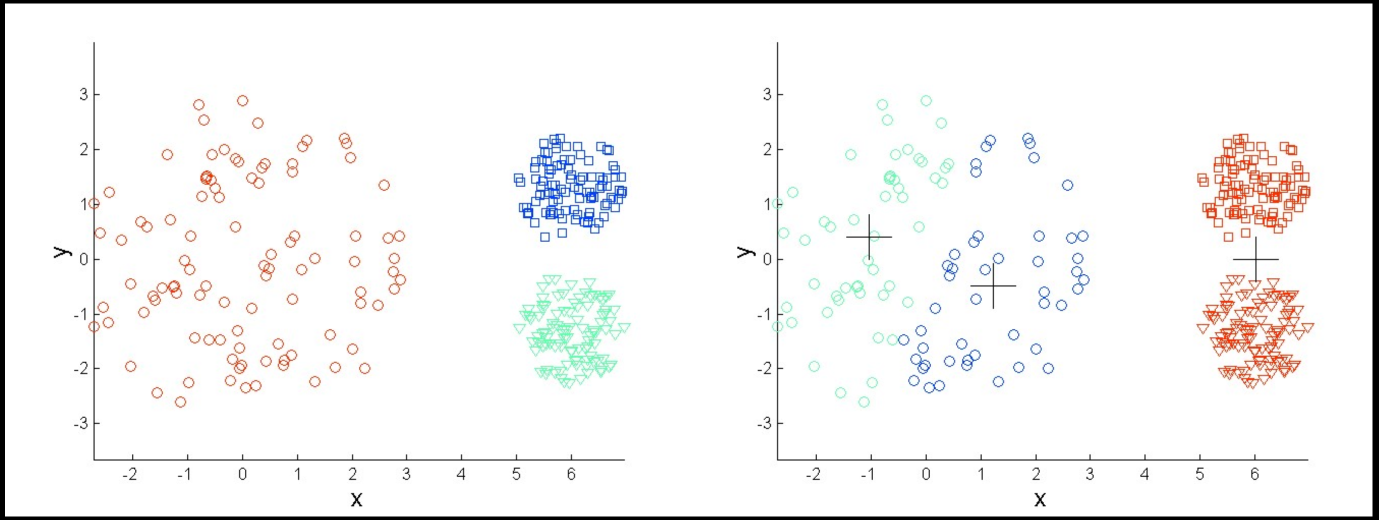


original data



K-means clustering

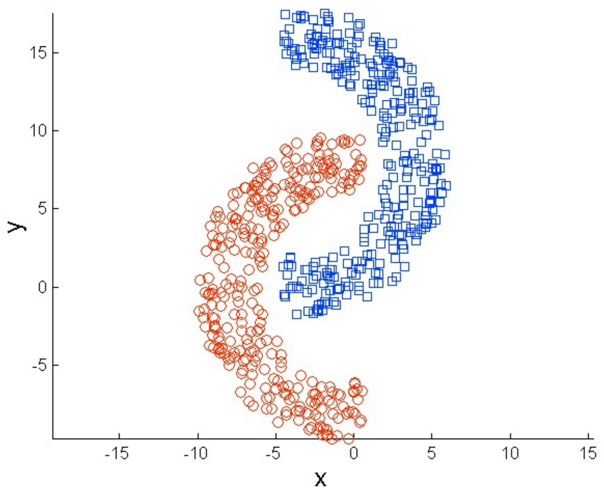
LIMITATIONS OF K-MEANS (DIFFERENT DENSITIES)



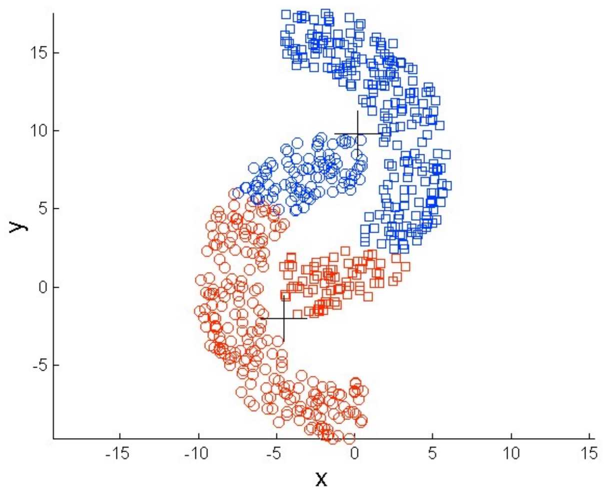
original data

K-means clustering

LIMITATIONS OF K-MEANS (NON-GLOBULAR SHAPES)

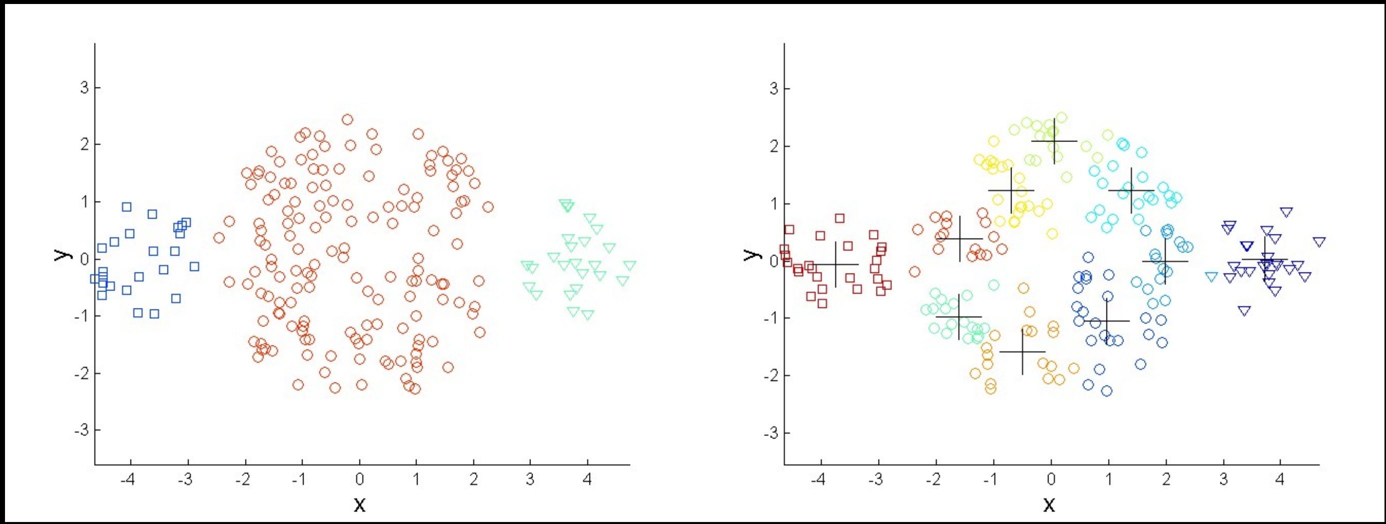


original data



K-means clustering

OVERCOMING K-MEANS LIMITATIONS (DIFFERENT SIZES)



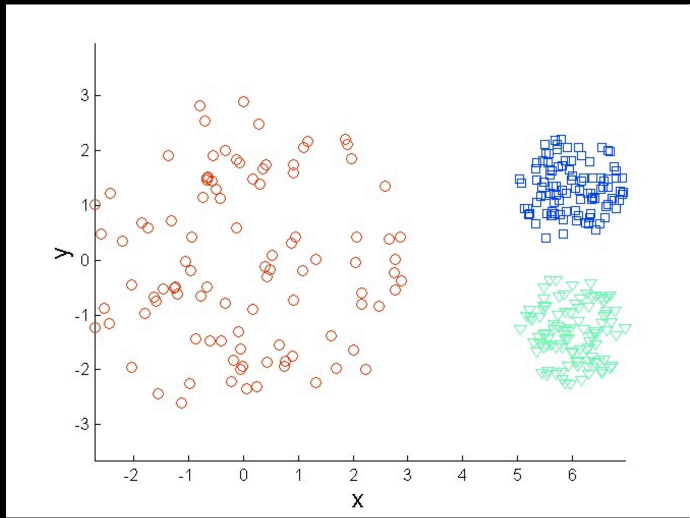
original data

K-means clustering

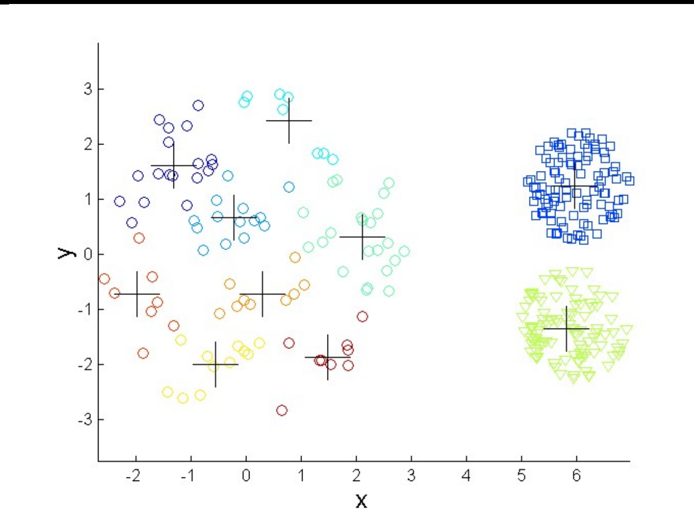
One solution is to find a large number of clusters such that each of them represents a part of a natural cluster.

But these small clusters need to be put together in a post-processing step.

OVERCOMING K-MEANS LIMITATIONS (DIFFERENT DENSITIES)



original data

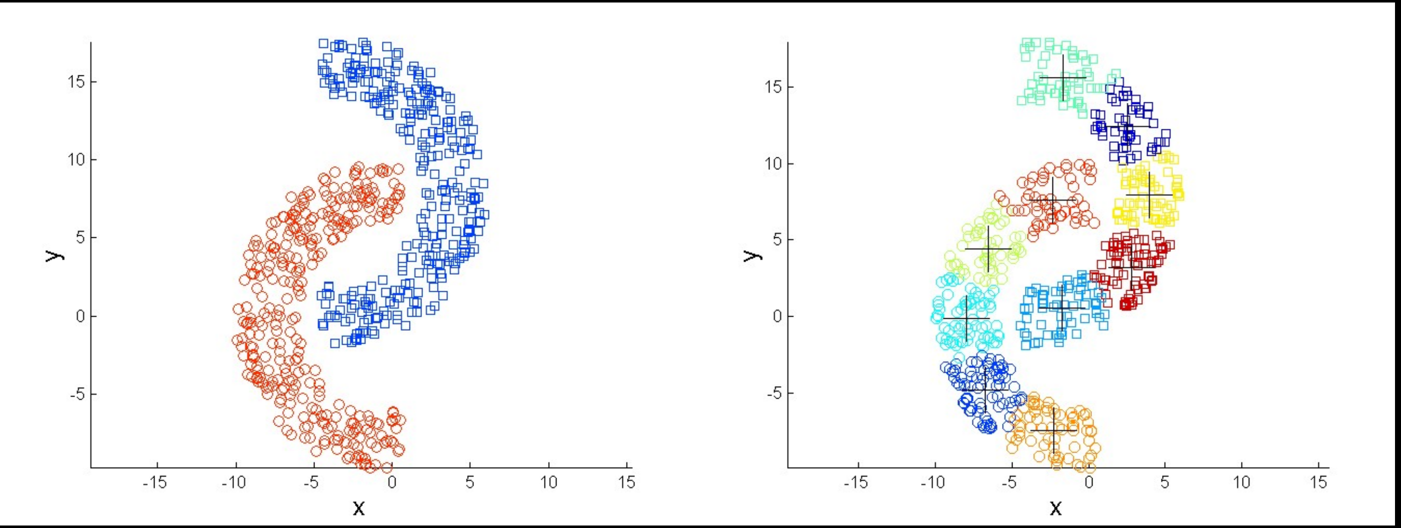


K-means clustering

One solution is to find a large number of clusters such that each of them represents a part of a natural cluster.

But these small clusters need to be put together in a post-processing step.

OVERCOMING K-MEANS LIMITATIONS (NON-GLOBULAR SHAPES)



original data K-means clustering

One solution is to find a large number of clusters such that each of them represents a part of a natural cluster.

But these small clusters need to be put together in a post-processing step.

RECAP

- K-means learning algorithm
- Examples
- Details, complexity, …
- Objective function
- Choosing initial centroids
- Limitations and how to overcome them