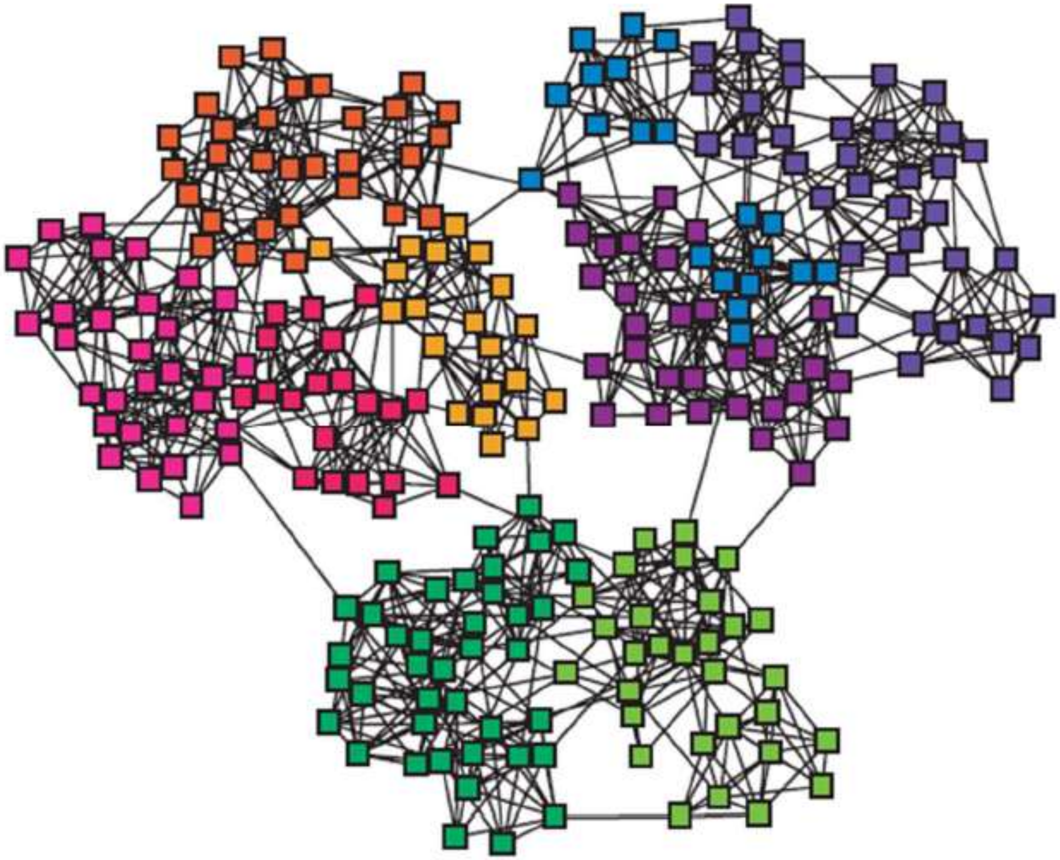
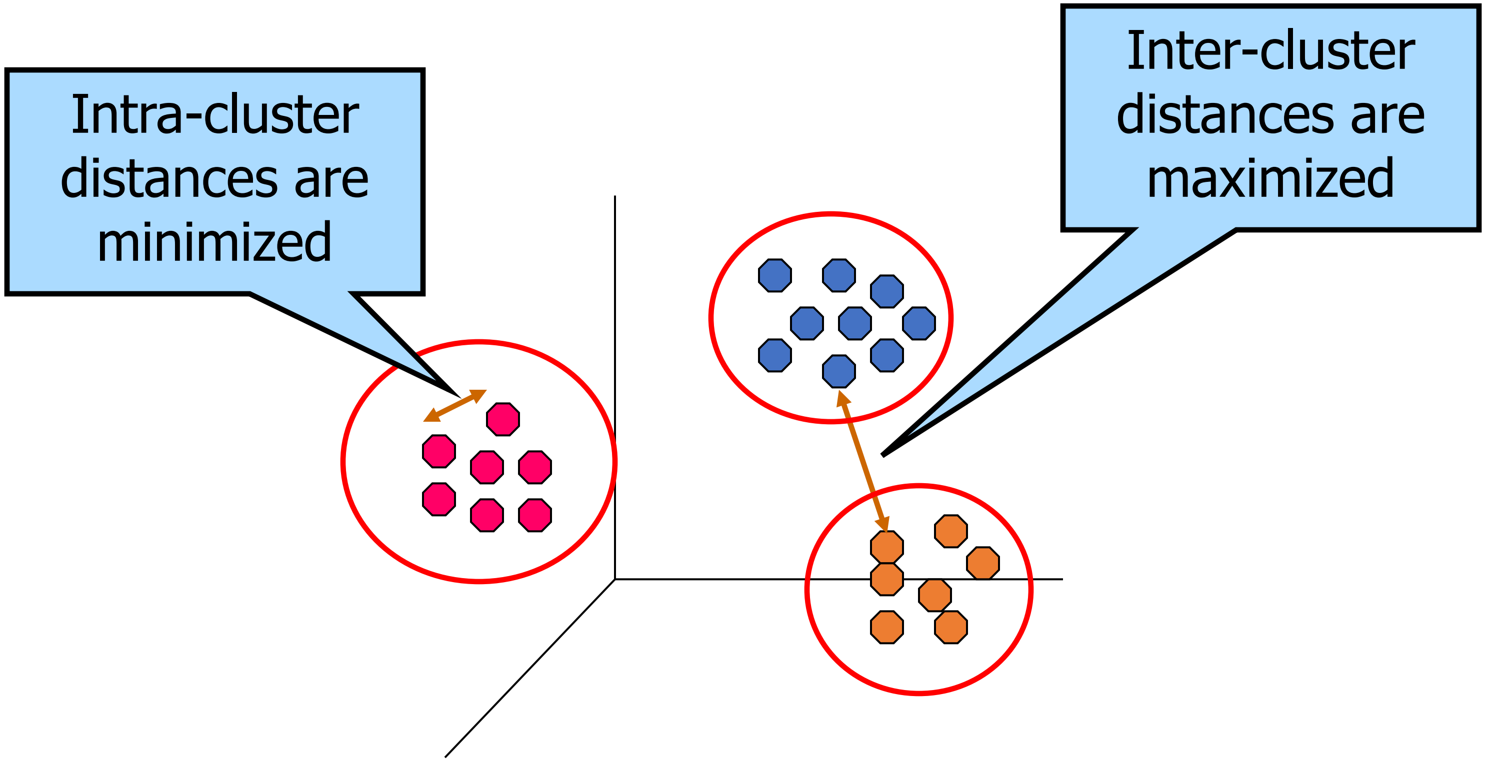


Cluster Analysis

Basic Concepts and Algorithms

What is Cluster Analysis?

- **Clustering**
  - Finding groups of objects such that the objects in a group will be similar to one another and different from the objects in other groups

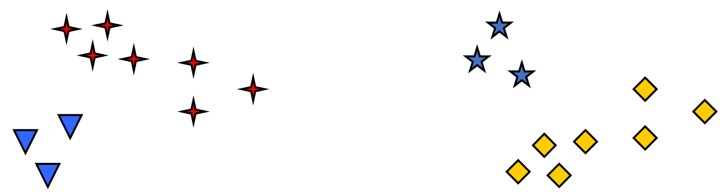


Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

Two Clusters



Four Clusters

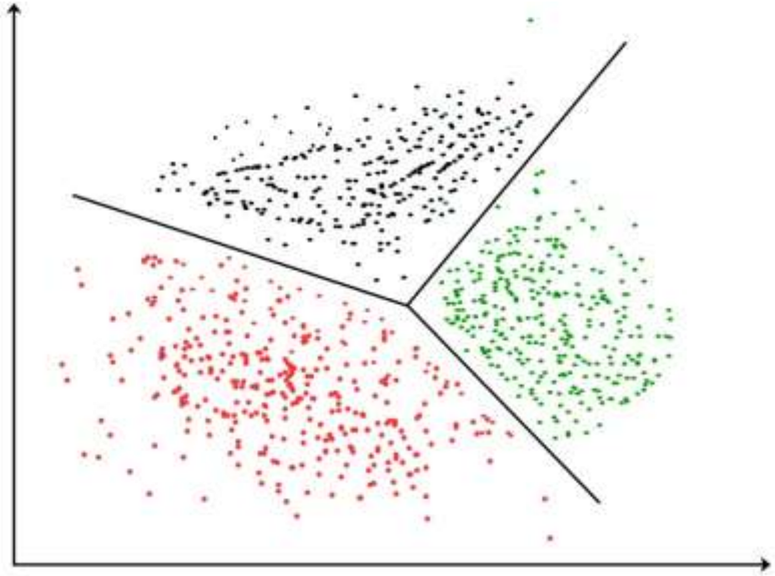
Clustering results are crucially dependent on the measure of similarity (or distance) between “points” to be clustered

Measure the Quality of Clustering

There is usually a separate “quality” function that measures the “goodness” of a cluster.

It is hard to define “similar enough” or “good enough”

- The answer is typically highly subjective



Similarity and Dissimilarity

Similarity

- Numerical measure of how alike two data objects are.
- Is higher when objects are more alike.
- Often falls in the range [0,1]

Dissimilarity

- Numerical measure of how different are two data objects
- Lower when objects are more alike
- Minimum dissimilarity is often 0
- Upper limit varies

Proximity refers to a similarity or dissimilarity

Euclidean Distance

- Euclidean Distance

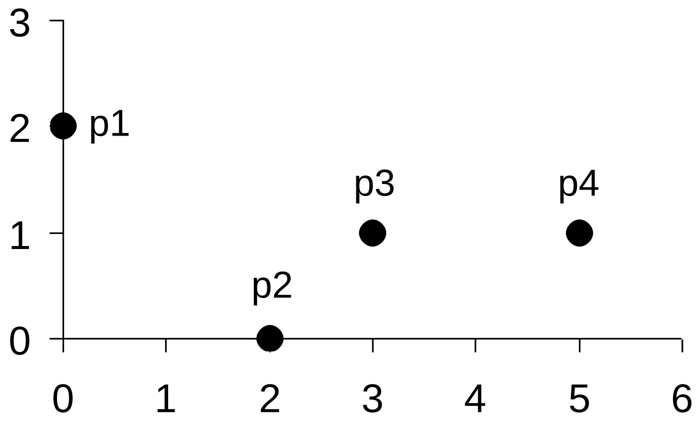
$$dist = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2}$$

Where *n* is the number of dimensions (attributes)

p_k and q_k are the k^{th} attributes (components) or data objects *p* and *q*.

- Standardization is necessary, if scales differ.

Euclidean Distance



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Distance Matrix

$$dist = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2}$$

Minkowski Distance: Examples

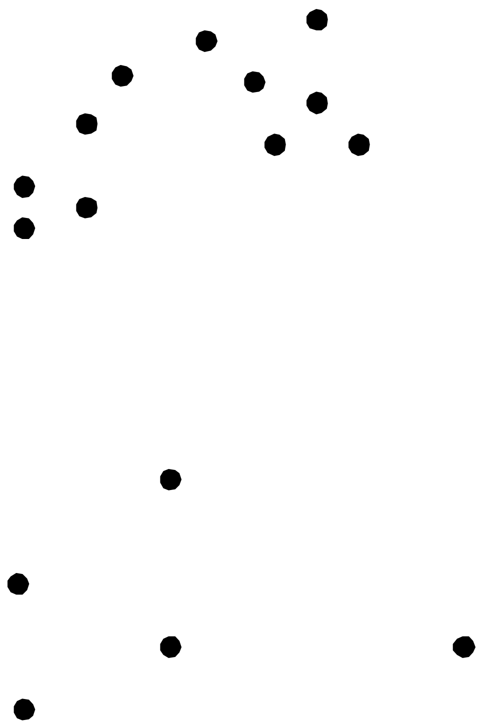
- $r = 1$. Manhattan distance
  - A common example of this is the Hamming distance, which is just the number of bits that are different between two binary vectors

- $r = 2$. Euclidean distance

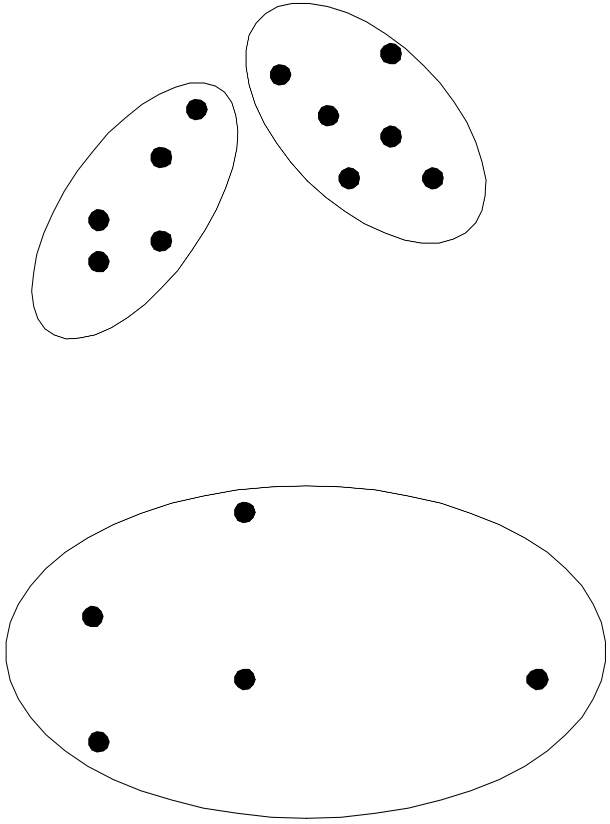
- **Do not confuse *r* with *n*, i.e., all these distances are defined for all numbers of dimensions.**

Partitional Clustering

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



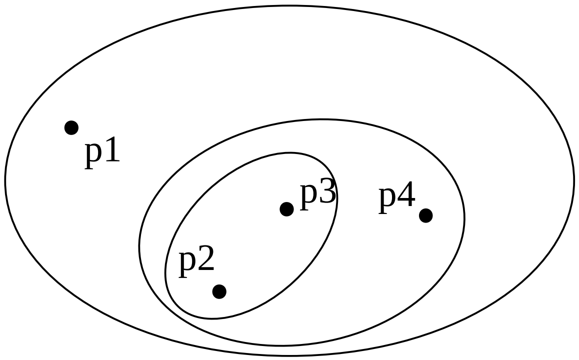
Original Points



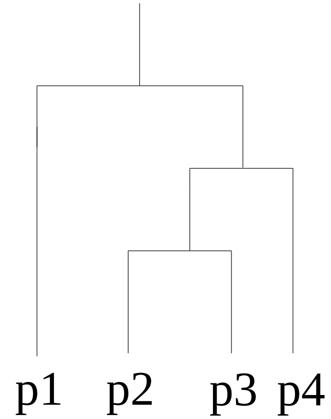
A Partitional Clustering

Hierarchical Clustering

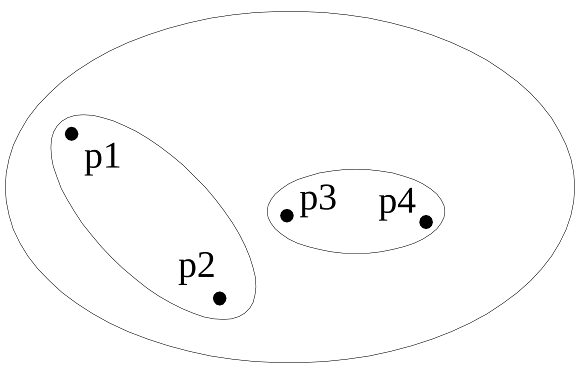
A set of nested clusters organized as a hierarchical tree



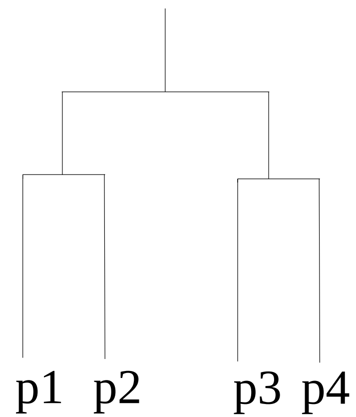
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



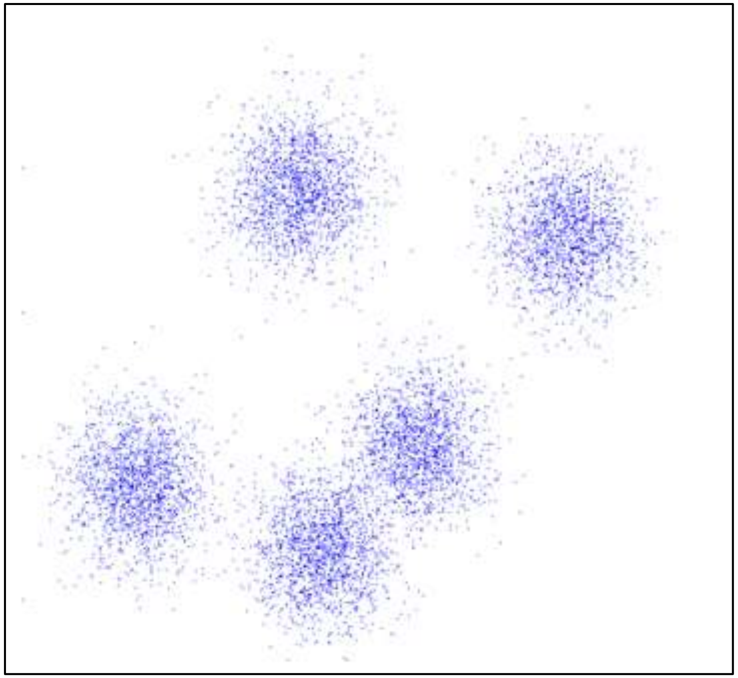
Non-traditional Dendrogram

K-Means: Partitioning approach

- An iterative clustering algorithm
- 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

Initialize: Pick *K* random points as cluster centers
Repeat:
1. Assign data points to the closest cluster center
2. Change the cluster center to the average of its assigned points

Until The centroids don't change

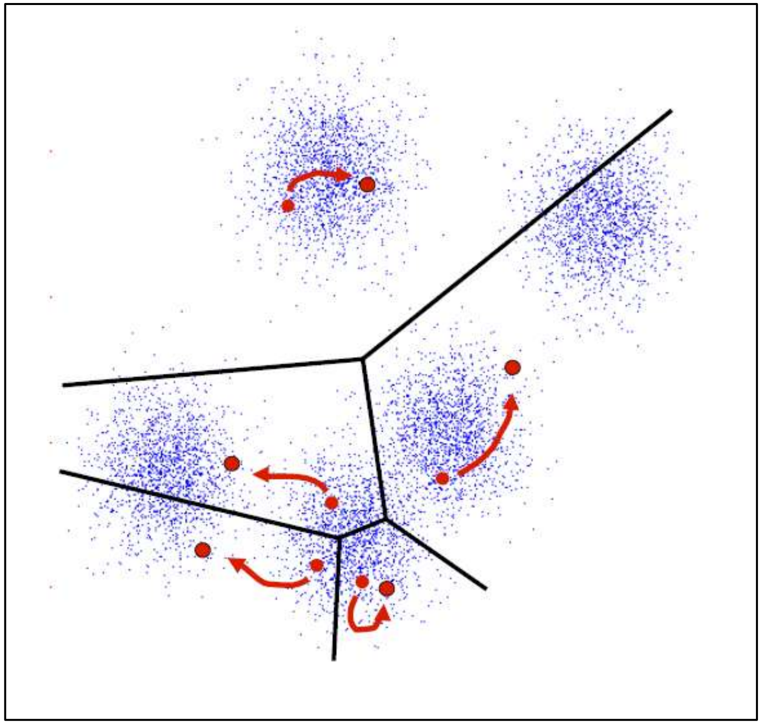


K-Means : Partitioning approach

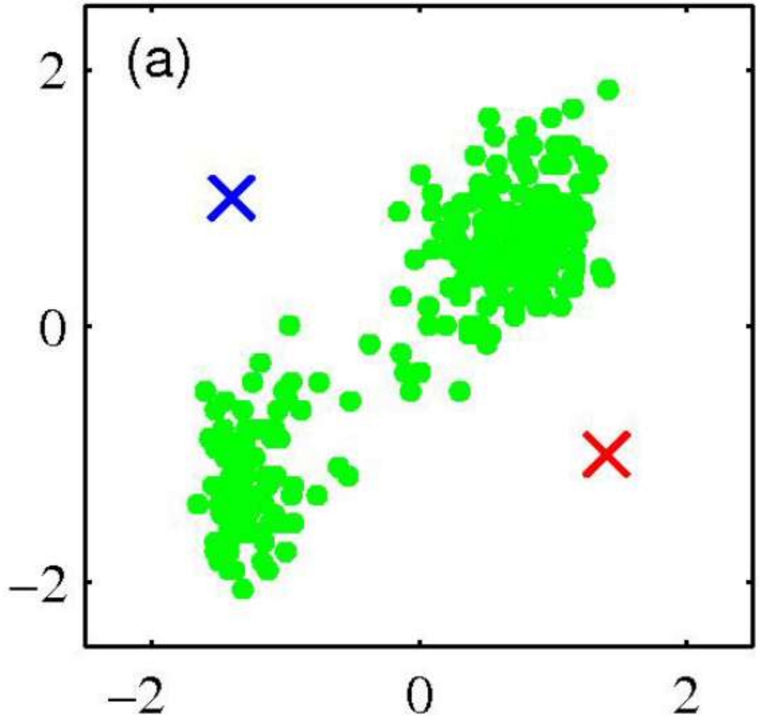
- An iterative clustering algorithm
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified

Initialize: Pick *K* random points as cluster centers
Repeat:
1. Assign data points to closest cluster center
2. Change the cluster center to the average of its assigned points

Until The centroids don't change



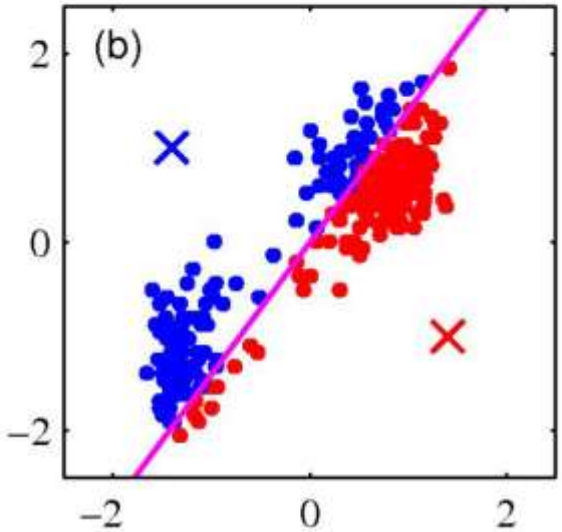
K-means clustering Example



- Pick K random points as cluster centers (means)

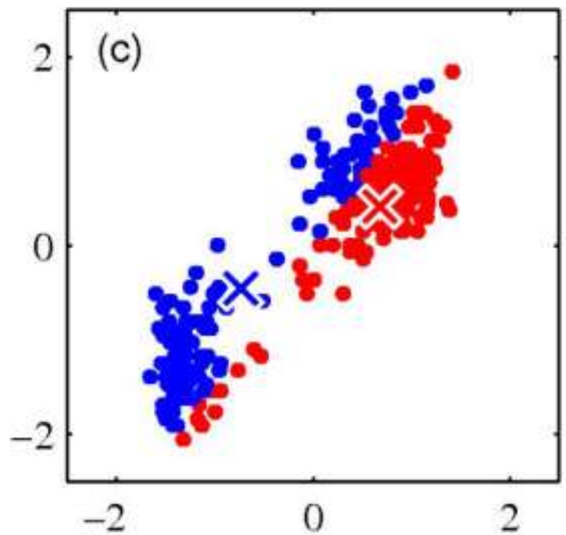
Shown here for K=2

K-means clustering Example



Iteration 1

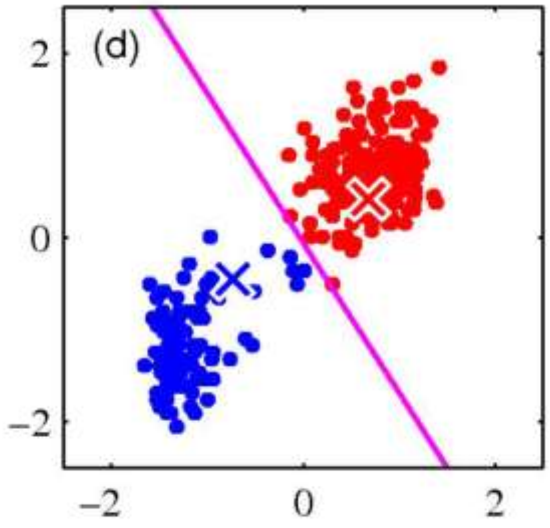
Iterative Step 1: Assign data points to closest cluster center



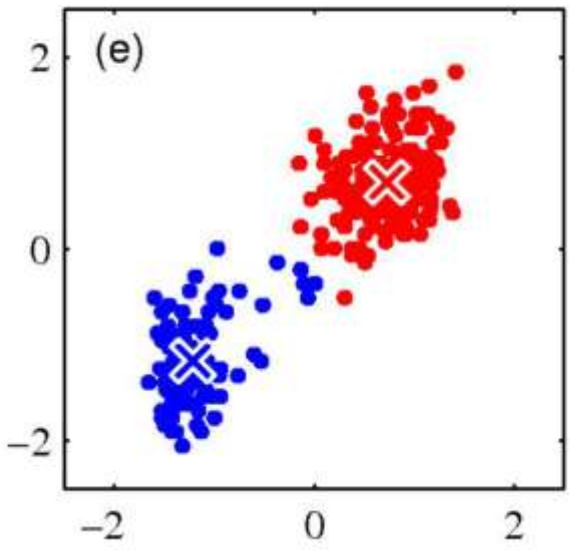
Iterative Step 2: Change the cluster center to average of the assigned points

K-means clustering Example

Iteration 2



Iterative Step 1: Assign data points to closest cluster center

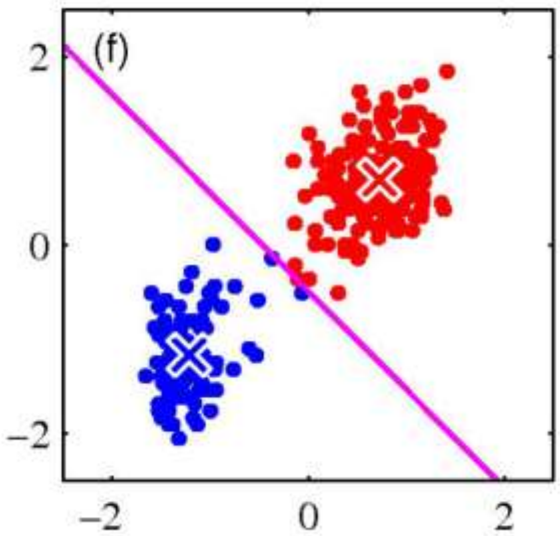


Repeat until convergence

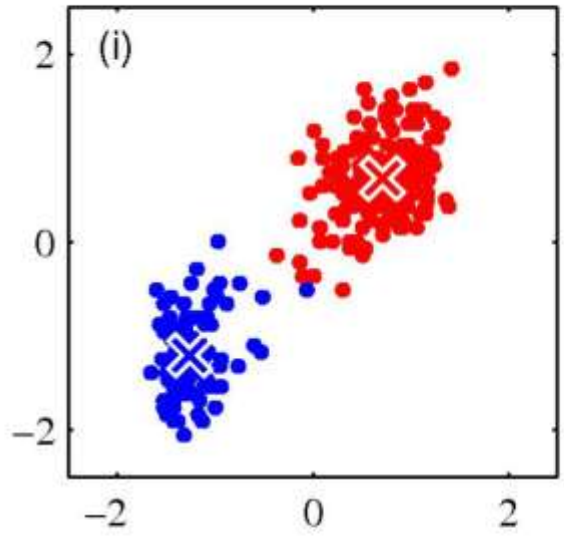
Iterative Step 2: Change the cluster center to average of the assigned points

K-means clustering Example

Iteration 3



Iterative Step 1: Assign data points to closest cluster center



Iterative Step 2: Change the cluster center to average of the assigned points

Repeat until convergence

K-means Clustering – Details

The centroid is (typically) the mean of the points in the cluster.

Initial centroids are often chosen randomly.

- Clusters produced vary from one run to another.

'Closeness' is measured by Euclidean distance, 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'

Complexity is O(n * K * I)

n = number of points,

K = number of clusters,

I = number of iterations

K-Means: Step-By-Step Example

Subject	A	B
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

	Individual	Mean Vector (centroid)
Group 1	1	(1.0, 1.0)
Group 2	4	(5.0, 7.0)

Choose two points as centroid

The remaining points are allocated to the cluster with which they are closest in term of Euclidean distance to the cluster mean.

	Cluster 1	Cluster 2
Step	Individual	Individual
1	1	4
2	1, 2	4
3	1, 2, 3	4
4	1, 2, 3	4, 5
5	1, 2, 3	4, 5, 6
6	1, 2, 3	4, 5, 6, 7

http://mnemstudio.org/clustering-k-means-example-1.htm

k-Means: Step-By-Step Example

	Individual	Mean Vector (centroid)
Cluster 1	1, 2, 3	(1.8, 2.3)
Cluster 2	4, 5, 6, 7	(4.1, 5.4)

Mean of new clusters

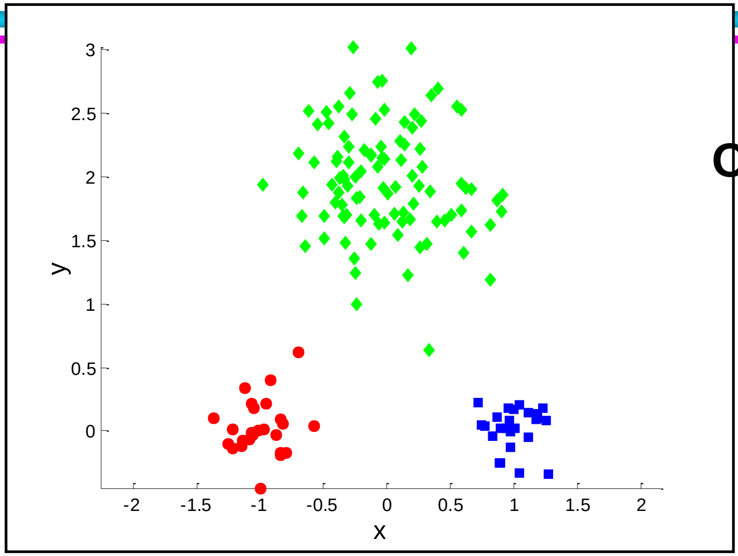
Recalculate the cluster of each point.

Individual	Distance to mean (centroid) of Cluster 1	Distance to mean (centroid) of Cluster 2
1	1.5	5.4
2	0.4	4.3
3	2.1	1.8
4	5.7	1.8
5	3.2	0.7
6	3.8	0.6
7	2.8	1.1

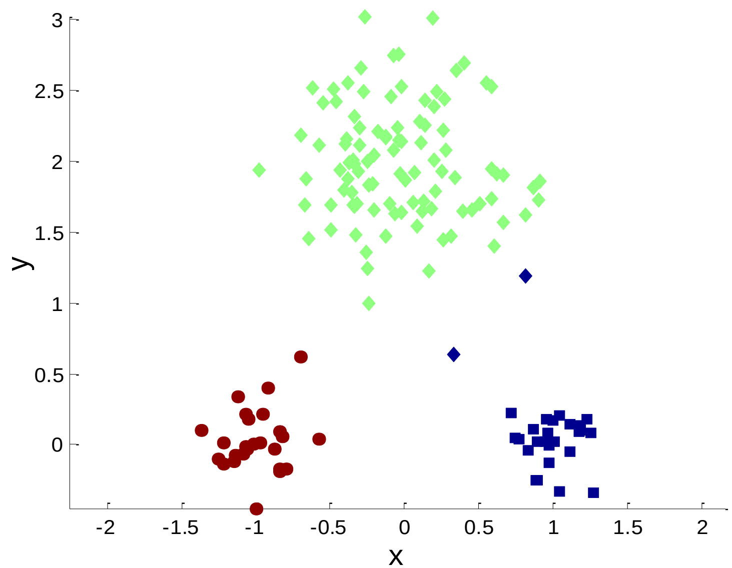
	Individual	Mean Vector (centroid)
Cluster 1	1, 2	(1.3, 1.5)
Cluster 2	3, 4, 5, 6, 7	(3.9, 5.1)

http://mnemstudio.org/clustering-k-means-example-1.htm

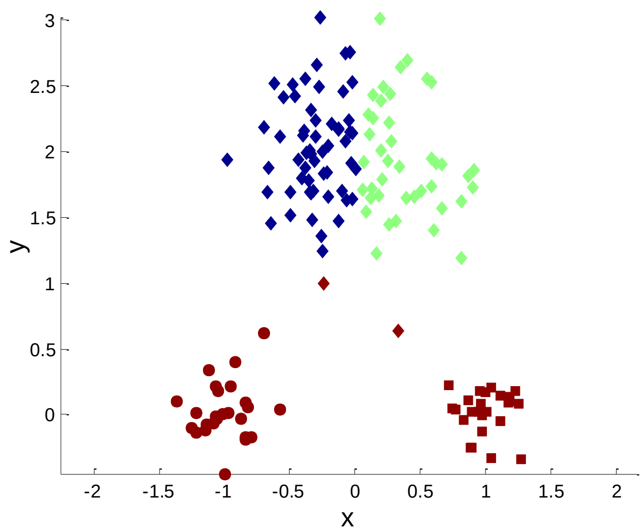
Two different K-means Clusterings



Original Points

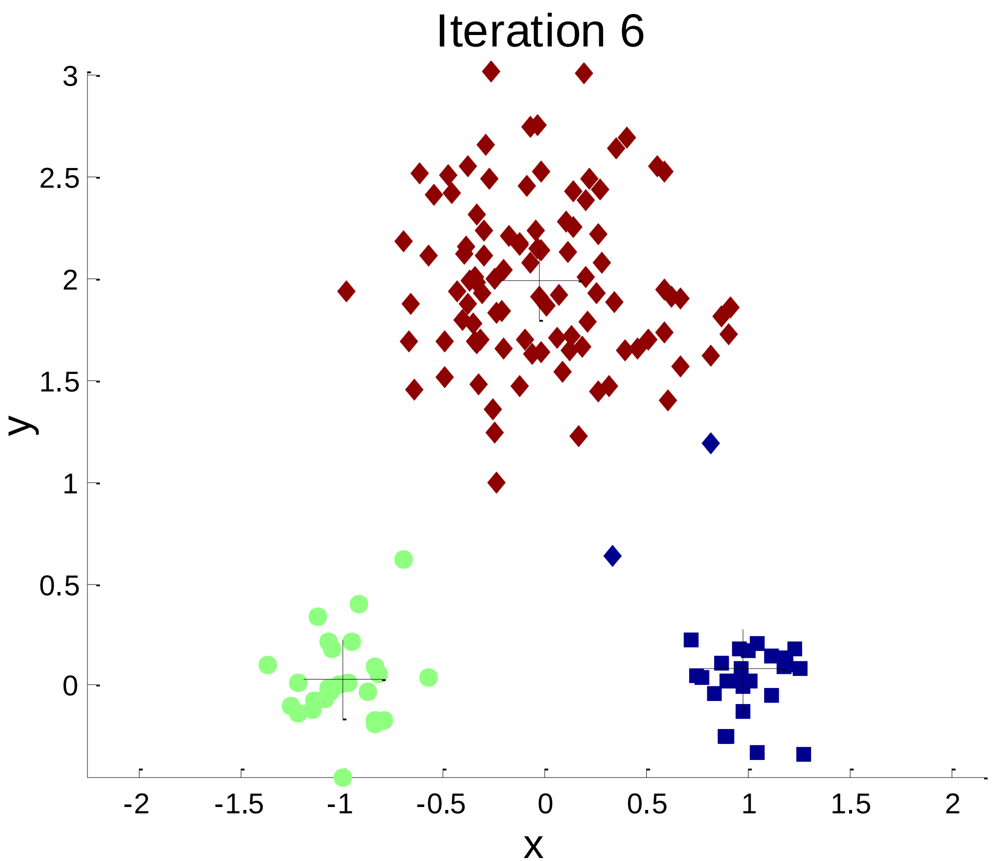


Optimal Clustering

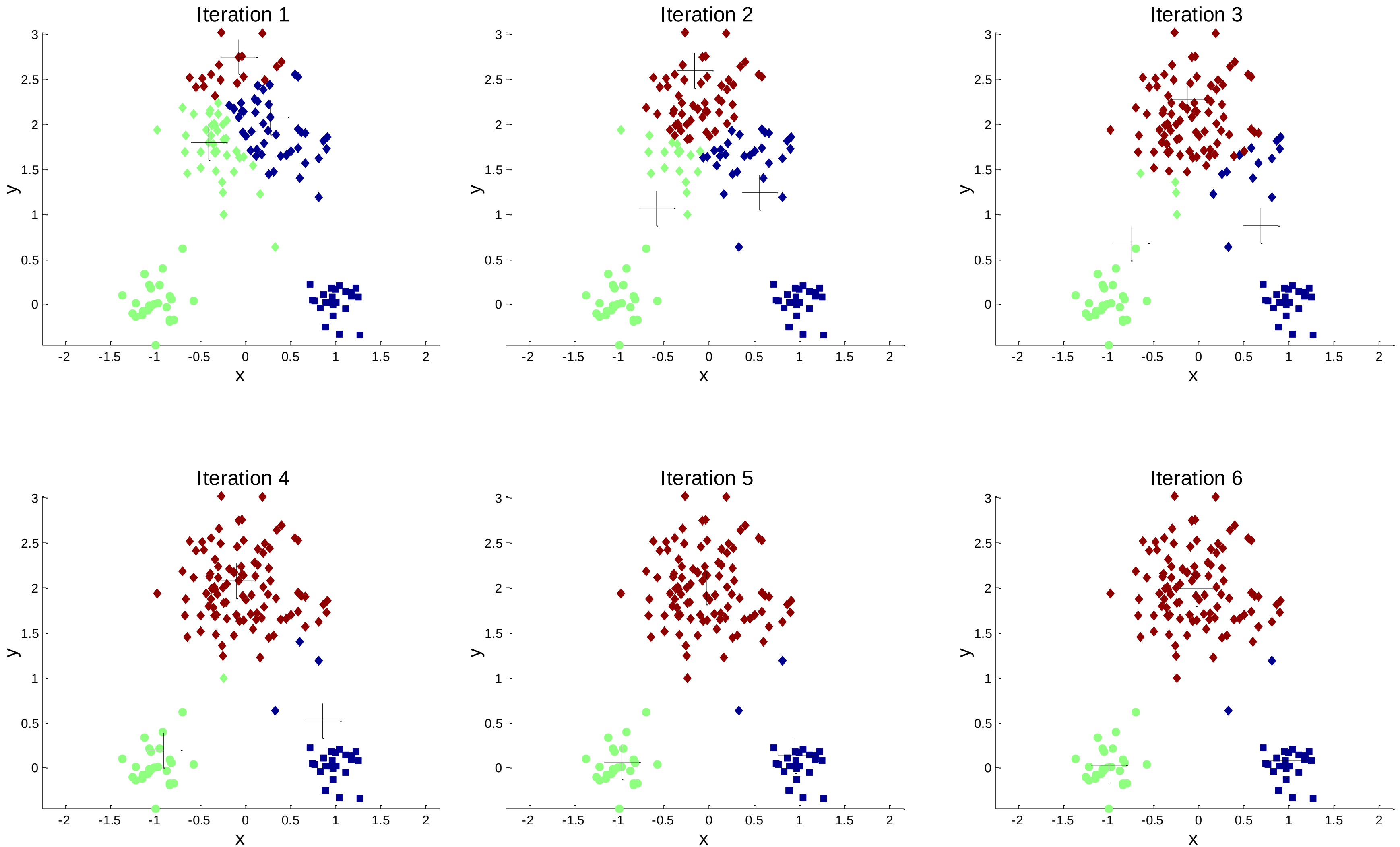


Sub-optimal Clustering

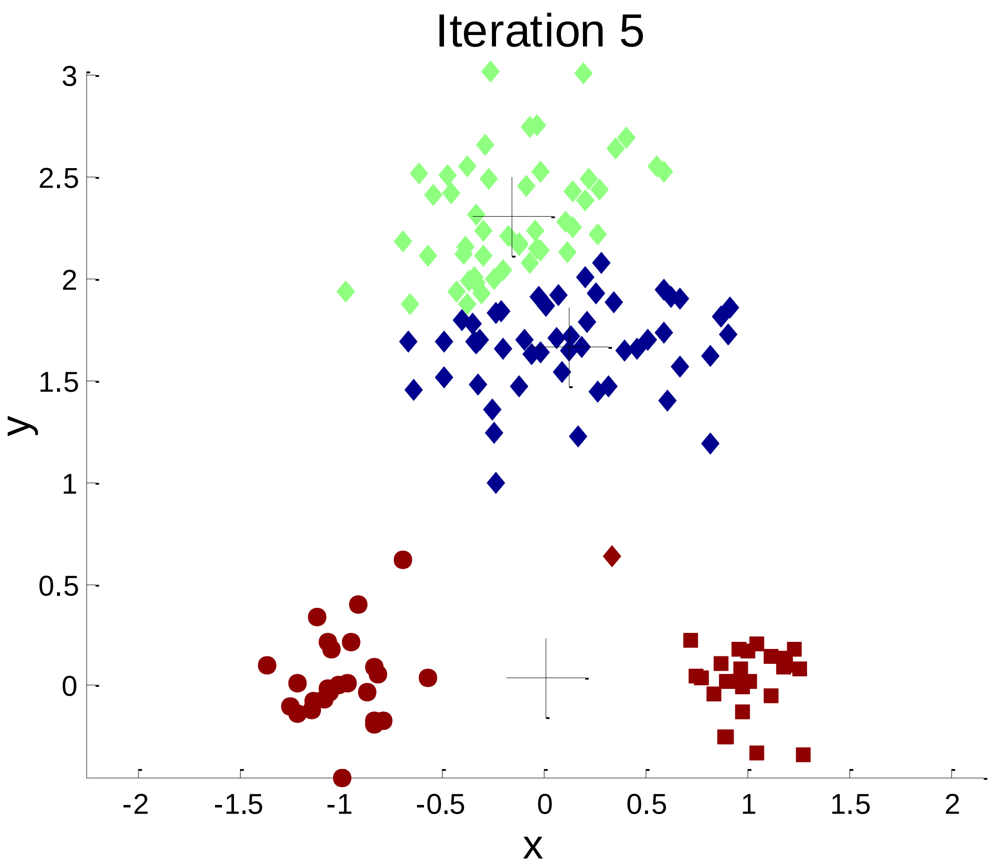
Importance of Choosing Initial Centroids



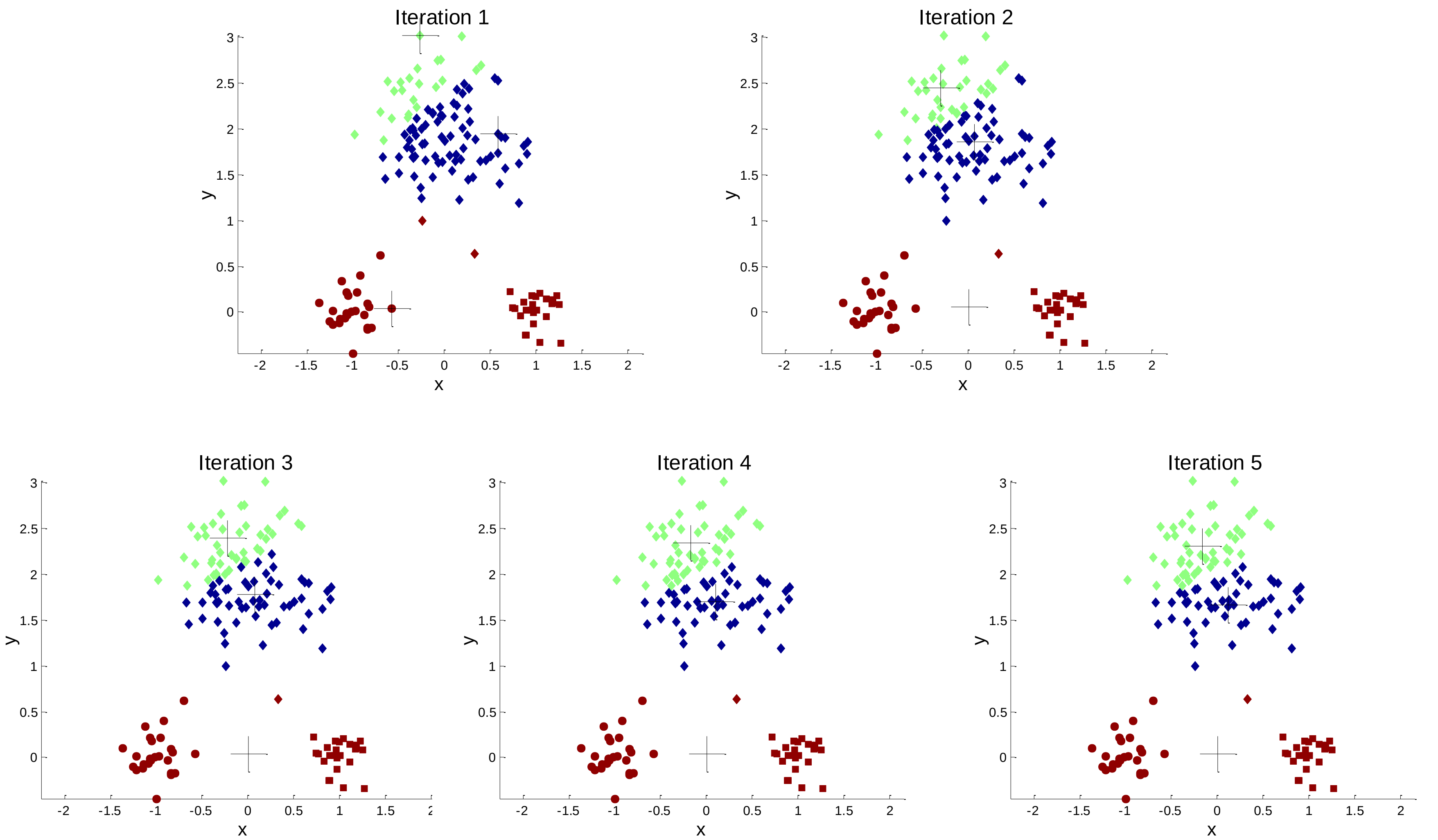
Importance of Choosing Initial Centroids



Importance of Choosing Initial Centroids …



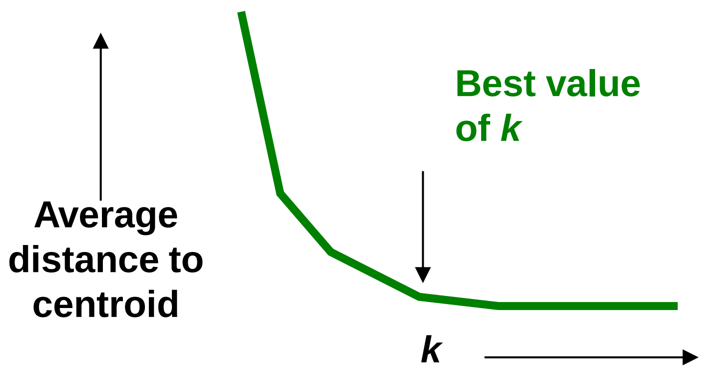
Importance of Choosing Initial Centroids …



Getting the *k* right

How to select *k*?

- Try different **k**, looking at the change in the average distance to centroid as **k** increases
- Average falls rapidly until right **k**, then changes little

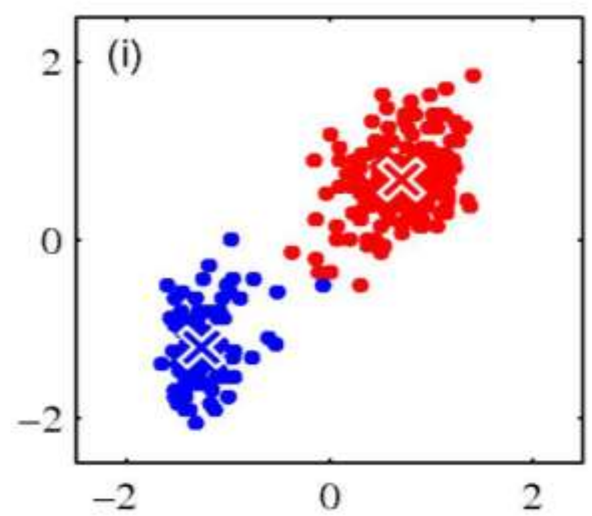


Evaluating K-means Clusters

- 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.

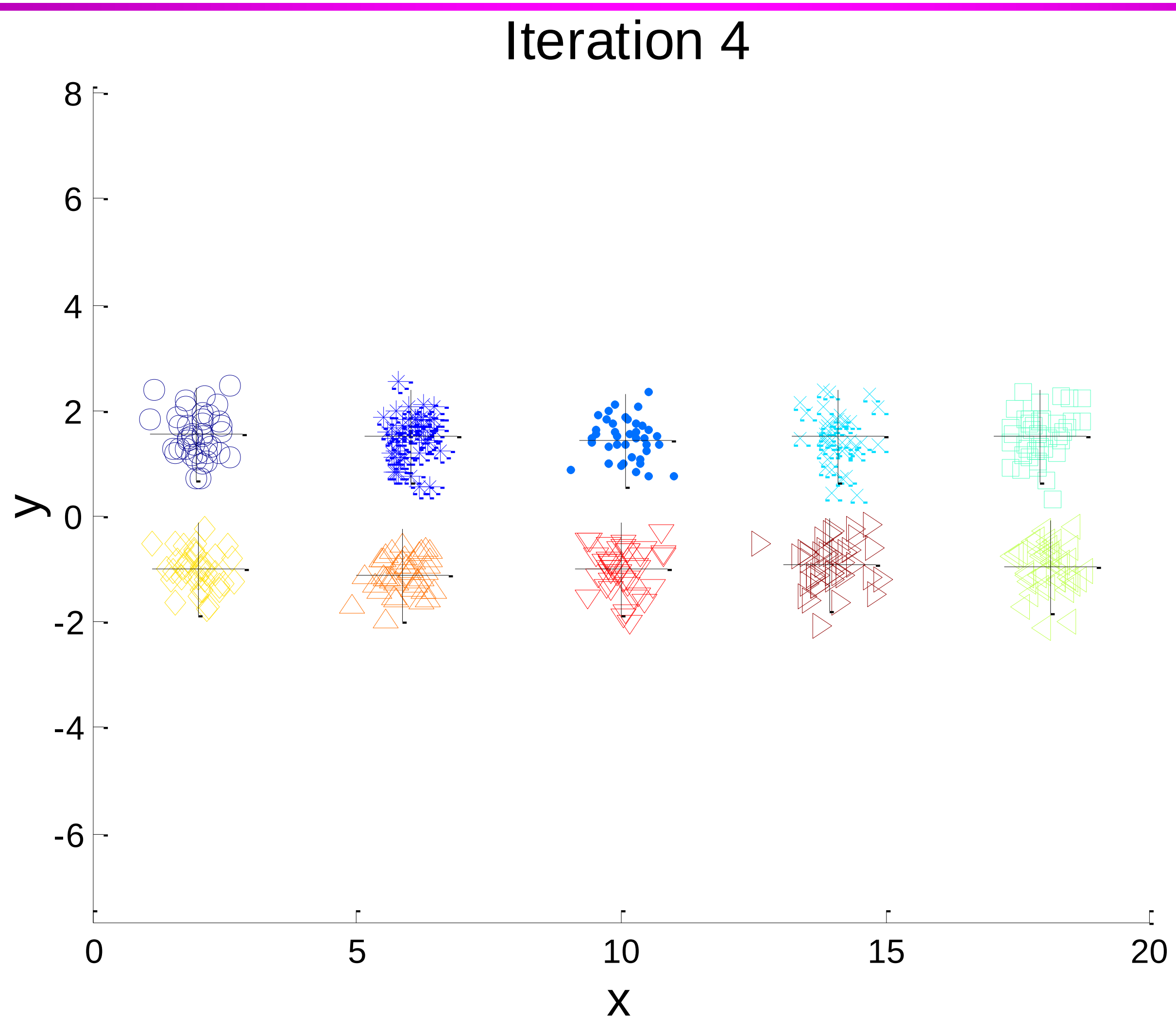
$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

- x is a data point in cluster C_i
- m_i is the center for cluster C_i



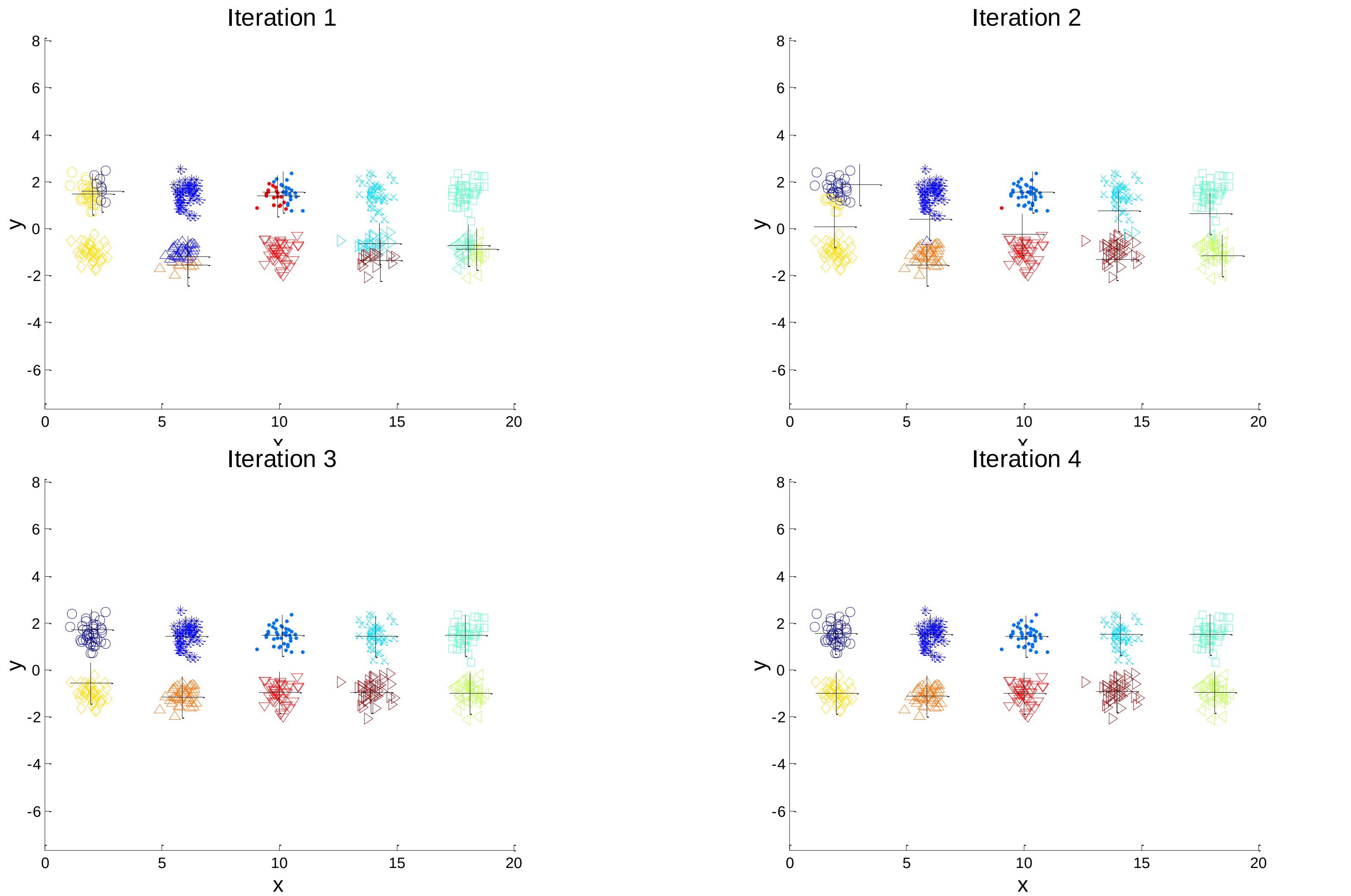
- Given two clusters, we choose the one with the smallest error
- One easy way to reduce SSE is to increase K, the number of clusters
  - A good clustering with smaller K can have a lower SSE than a poor clustering with higher K

10 Clusters Example



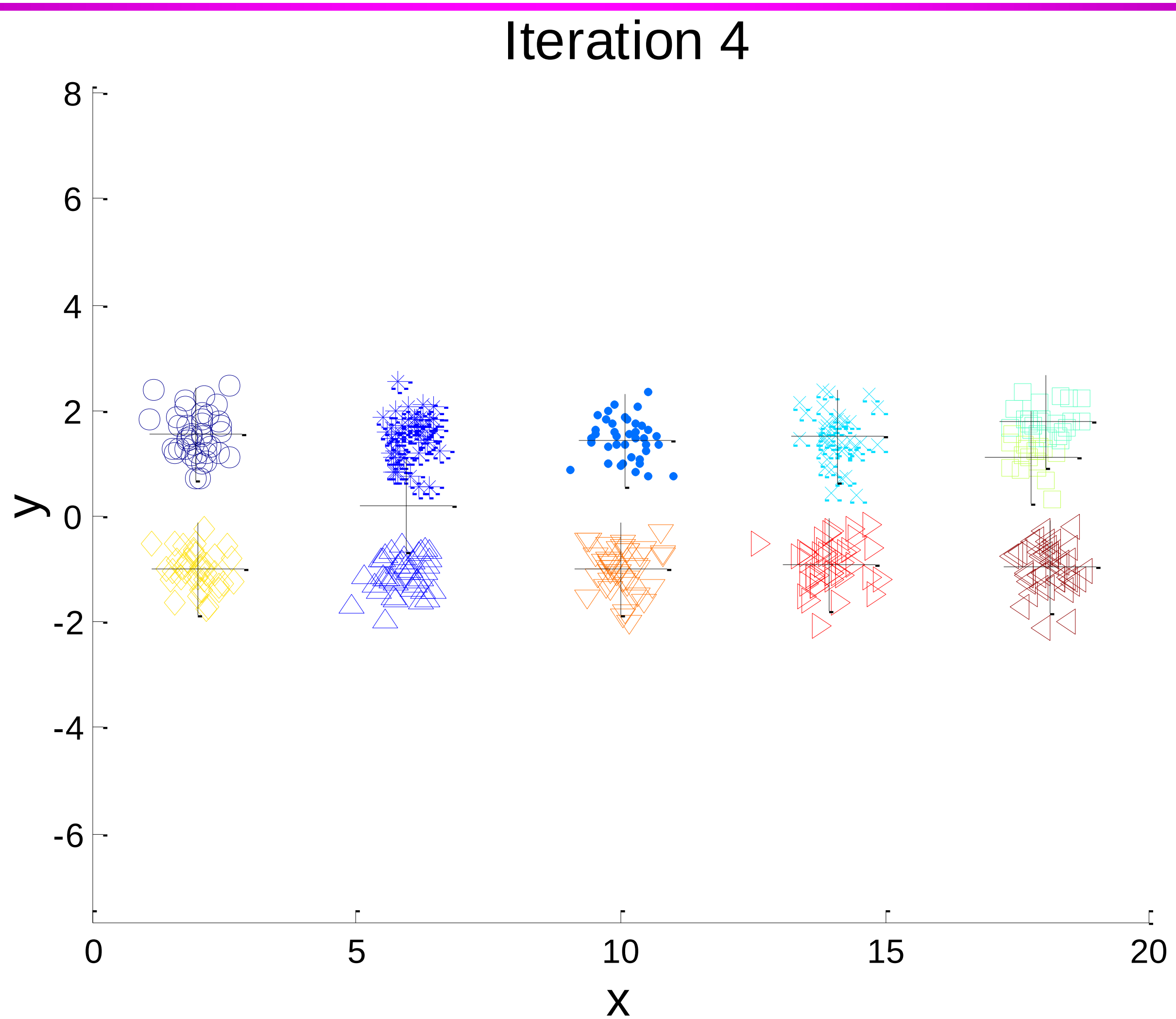
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example



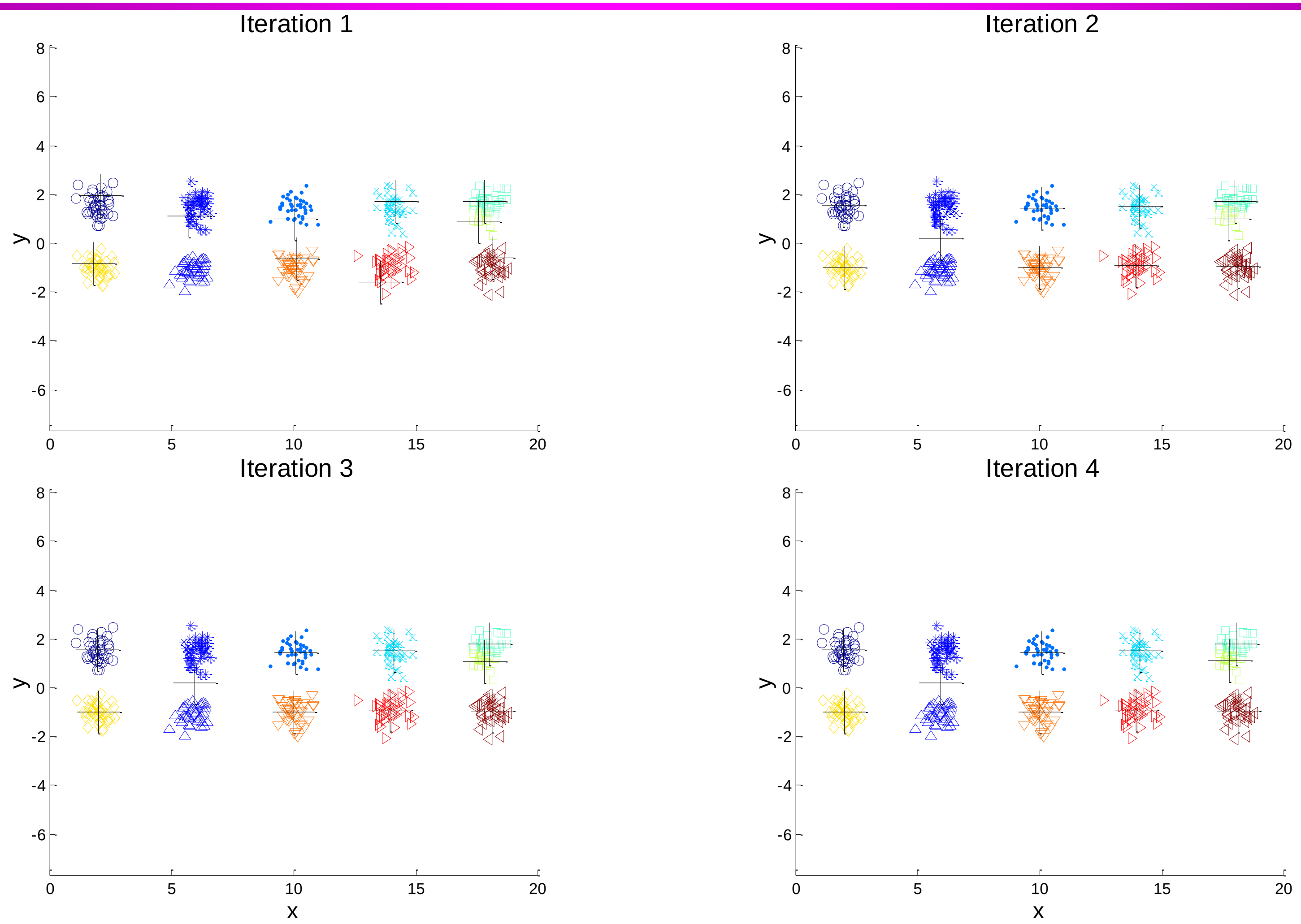
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example



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

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
- Post-processing
- **Bisecting K-means**
  - Not as susceptible to initialization issues

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 is a 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

Bisecting K-means Example