CLUSTERING

Sanjay Ranka
Distinguished Professor
Department of Computer and Information Science and Engineering
www.sanjayranka.com
sanjayranka@gmail.com
352 514 4213

Notion of a Cluster is Ambiguous

Initial points.

Six Clusters

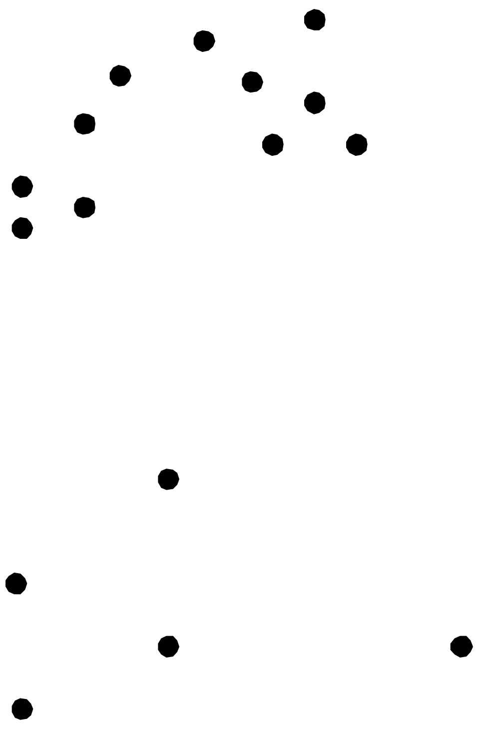
Two Clusters

Four Clusters

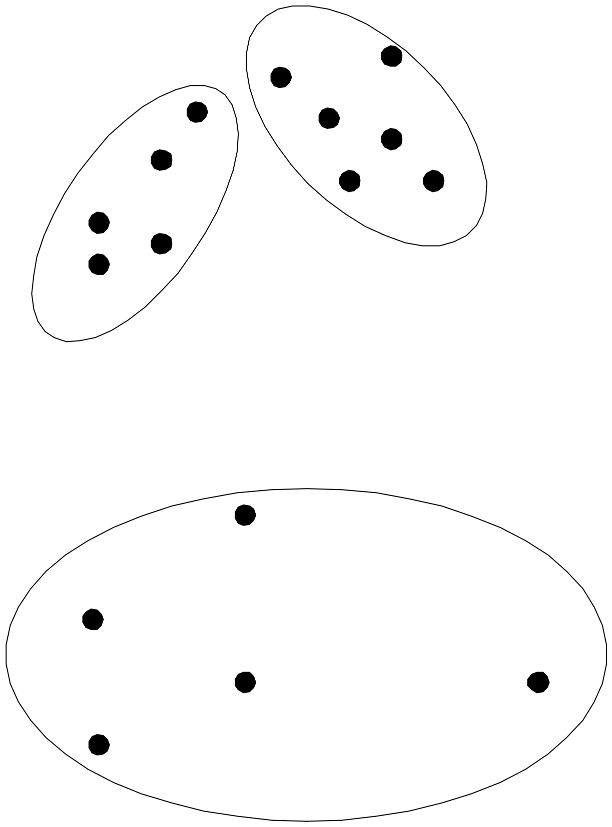
Types of Clustering

- A *clustering* is a set of clusters.
- One important distinction is 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

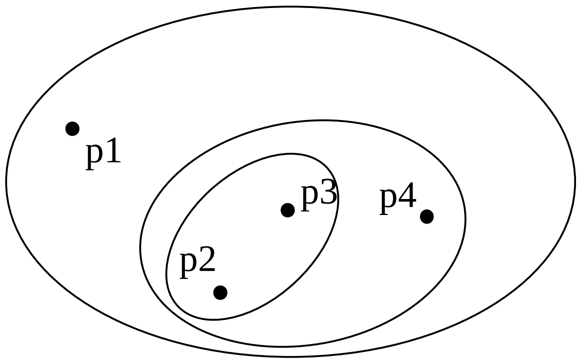


Original Points

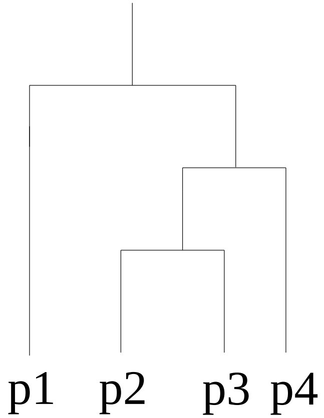


A Partitional Clustering

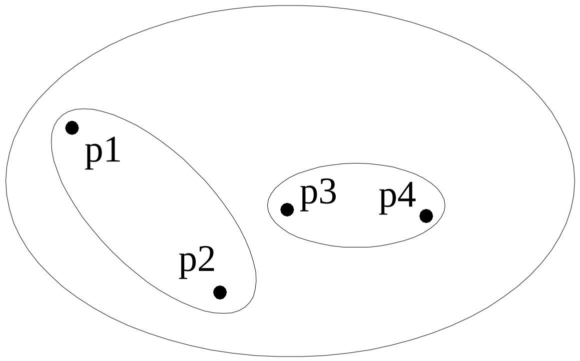
Hierarchical Clustering



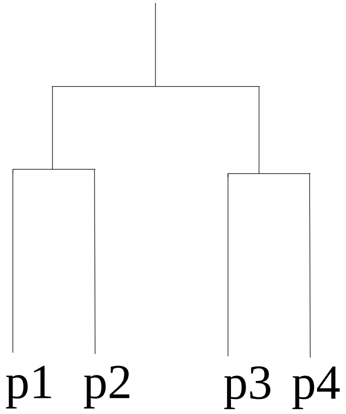
Traditional Hierarchical Clustering



Traditional Dendrogram



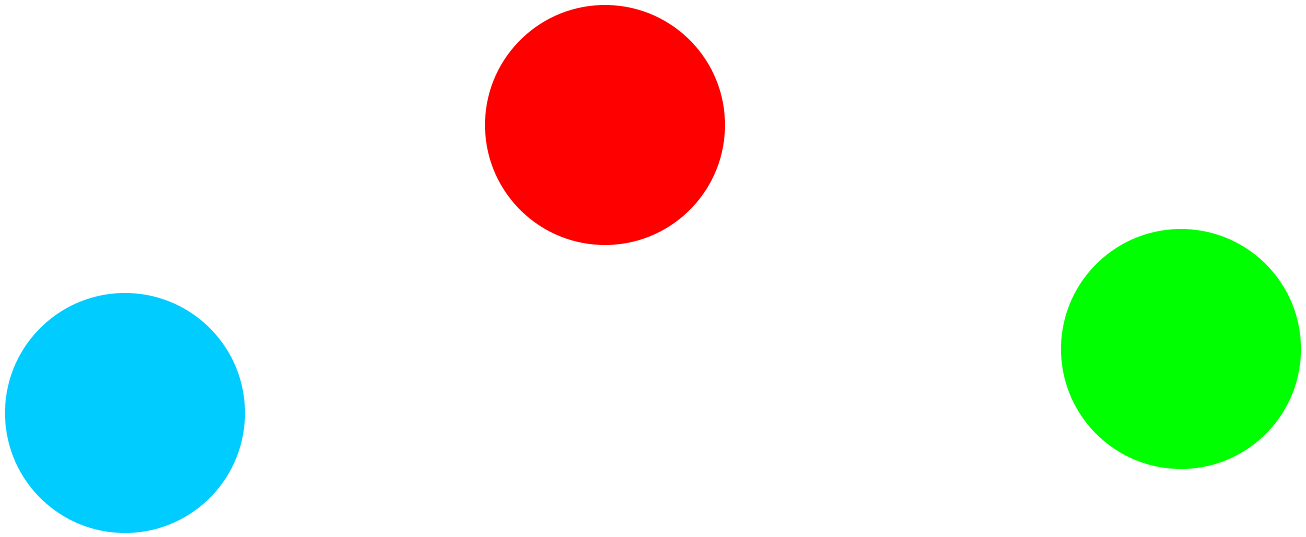
Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

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.

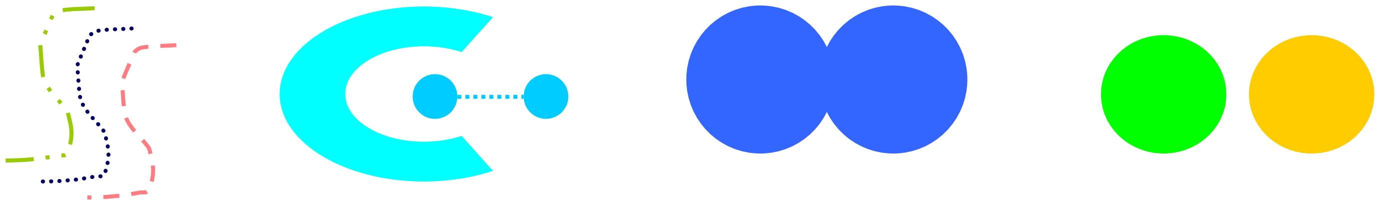


Types of Clusters: Center-Based

- 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.

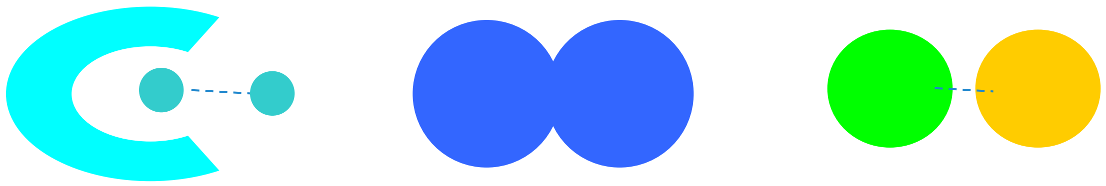
Types of Clusters: Contiguity-Based

- 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.



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.
  - The three curves don't form clusters since they fade into the noise, as does the bridge between the two small circular clusters.



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 two data objects are.
  - Is lower when objects are more alike.
  - Minimum dissimilarity is often 0.
  - Upper limit varies
- Proximity refers to a similarity or dissimilarity

Summary of Similarity/Dissimilarity for Simple Attributes

p and q are the attribute values for two data objects.

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } p = q \\ 1 & \text{if } p \neq q \end{cases}$	$s = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases}$
Ordinal	$d = \frac{\lvert p-q \rvert}{n-1}$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - \frac{\lvert p-q \rvert}{n-1}$
Interval or Ratio	$d = \lvert p - q \rvert$	$s = -d$, $s = \frac{1}{1+d}$ or $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$

Table 5.1. Similarity and dissimilarity for simple attributes

Euclidean Distance

- Euclidean Distance

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

  - where n is the number of dimensions (attributes) and p_k and q_k are, respectively, the k^{th} attributes (components) or data objects p and q.

- Standardization is necessary, if scales differ.

Minkowski Distance

- Minkowski Distance is a generalization of Euclidean Distance

$$dist = \left(\sum_{k=1}^{n} | p_k - q_k |^r \right)^{\frac{1}{r}}$$

  - where r is a parameter, n is the number of dimensions (attributes) and p_k and q_k are, respectively, the k^{th} attributes (components) or data objects p and q.

Minkowski Distance: Examples

- $r = 1$. City block (Manhattan, taxicab, L_1 norm) 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.
- $r \rightarrow \infty$. "supremum" (L_{max} norm, L_{∞} norm) distance.
  - This is the maximum difference between any component of the vectors.
- Do not confuse *r* with *n*, i.e., all these distances are defined for all numbers of dimensions.

Minkowski Distance

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

L1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

L2	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

L_∞	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

Distance Matrix

Common Properties of a Distance and Similarity

Distances, such as the Euclidean distance, have some well-known properties:

1. $d(p, q) \geq 0$ for all p and q and $d(p, q) = 0$ only if $p = q$. (Positive definiteness)
2. $d(p, q) = d(q, p)$ for all p and q. (Symmetry)
3. $d(p, r) \leq d(p, q) + d(q, r)$ for all points p, q, and r. (Triangle Inequality)

– where $d(p, q)$ is the distance (dissimilarity) between points (data objects), p and q.

A distance that satisfies these properties is a *metric*

Similarities, also have some well-known properties:

1. $s(p, q) = 1$ (or maximum similarity) only if $p = q$.
2. $s(p, q) = s(q, p)$ for all p and q. (Symmetry)

– where $s(p, q)$ is the similarity between points (data objects), p and q.

Similarity Between Binary Vectors

- Common situation is that objects, *p* and *q*, have only binary attributes.
- Compute similarities using the following quantities

  M_{01} = the number of attributes where *p* was 0 and *q* was 1
  M_{10} = the number of attributes where *p* was 1 and *q* was 0
  M_{00} = the number of attributes where *p* was 0 and *q* was 0
  M_{11} = the number of attributes where *p* was 1 and *q* was 1
- Simple Matching and Jaccard Coefficients
- SMC = number of matches / number of attributes

  $= (M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00})$
- J = number of *11* matches / number of not-both-zero attributes values

  $= (M_{11}) / (M_{01} + M_{10} + M_{11})$

SMC versus Jaccard: Example

p = 1 0 0 0 0 0 0 0 0 0
q = 0 0 0 0 0 0 1 0 0 1

$M_{01} = 2$ (the number of attributes where p was 0 and q was 1)
$M_{10} = 1$ (the number of attributes where p was 1 and q was 0)
$M_{00} = 7$ (the number of attributes where p was 0 and q was 0)
$M_{11} = 0$ (the number of attributes where p was 1 and q was 1)

$$\text{SMC} = (M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00})$$
$$= (0+7) / (2+1+0+7) = 0.7$$

$$J = (M_{11}) / (M_{01} + M_{10} + M_{11})$$
$$= 0 / (2 + 1 + 0) = 0$$

Cosine Similarity

- If d_1 and d_2 are two document vectors, then

$$\cos(d_1, d_2) = (d_1 \bullet d_2) / ||d_1|| \, ||d_2|| ,$$

where $\bullet$ indicates vector dot product and $|| d ||$ is the length of vector d.

- Example:

d_1 = 3 2 0 5 0 0 0 2 0 0

d_2 = 1 0 0 0 0 0 0 1 0 2

$d_1 \bullet d_2$= 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5

$||d_1|| = (3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)^{0.5} = (42)^{0.5} = 6.481$

$||d_2|| = (1*1+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2)^{0.5} = (6)^{0.5} = 2.245$

$\cos(d_1, d_2) = .3150$

Correlation

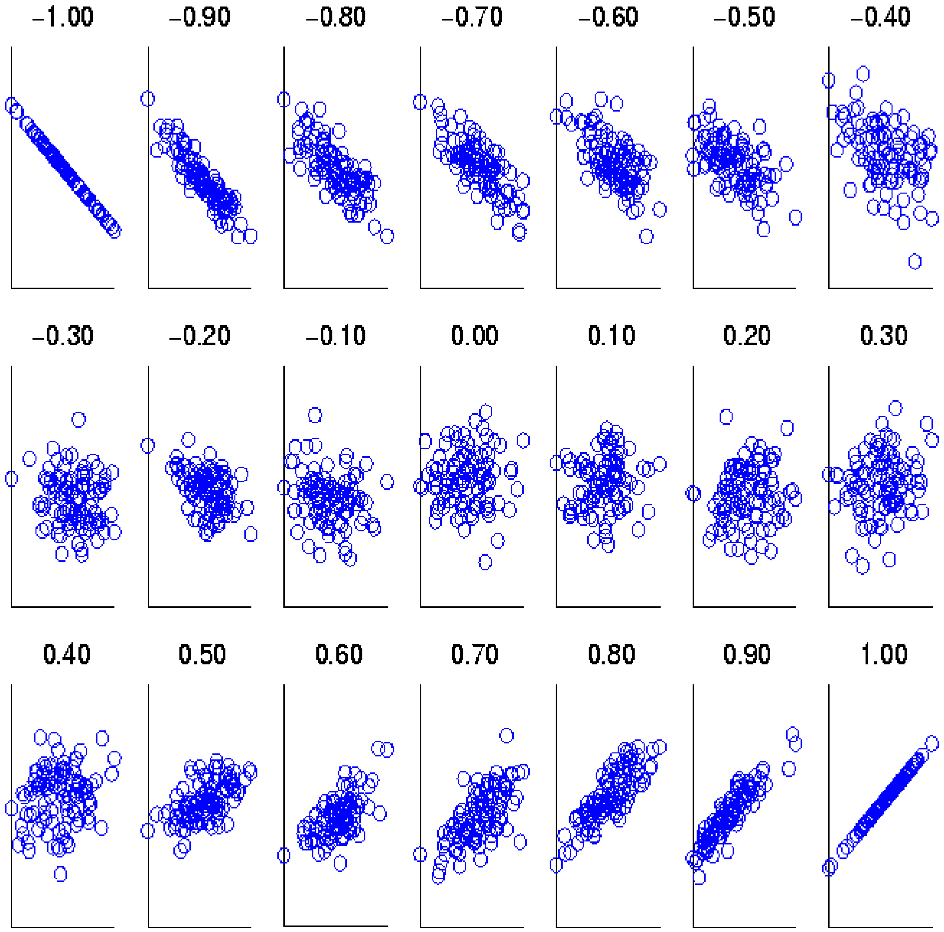
- Correlation measure the linear relationship between objects.
- To compute correlation, we standardize data objects, *p* and *q*, and then take the dot product.

$$p'_k = (p_k - mean(p)) / std(p)$$

$$q'_k = (q_k - mean(q)) / std(q)$$

$$correlation(p, q) = p' \bullet q'$$

Visually Evaluating Correlation



Scatter plots showing the similarity from –1 to 1

General Approach for Combining Similarities

- Sometimes attributes are of many different types, but an overall similarity is needed.

1. For the k^{th} attribute, compute a similarity, s_k, in the range $[0, 1]$.
2. Define an indicator variable, δ_k, for the k_{th} attribute as follows:

$$\delta_k = \begin{cases} 0 & \text{if the } k^{th} \text{ attribute is a binary asymmetric attribute and both objects have} \\ & \text{a value of } 0\text{, or if one of the objects has a missing values for the } k^{th} \text{ attribute} \\ 1 & \text{otherwise} \end{cases}$$

3. Compute the overall similarity between the two objects using the following formula:

$$similarity(p, q) = \frac{\sum_{k=1}^{n} \delta_k s_k}{\sum_{k=1}^{n} \delta_k}$$

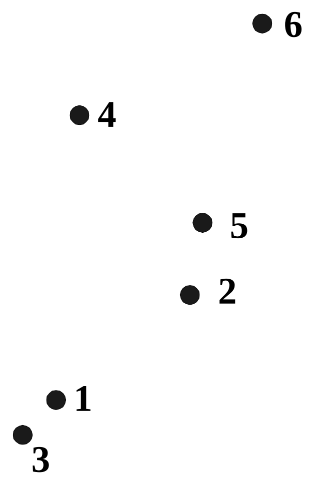
Weighted Similarity

- May not want to treat all attributes the same.
  - Use weights w_k which are between 0 and 1 and sum to 1.

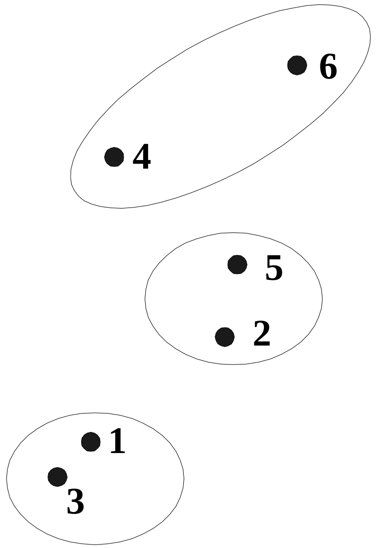
$$similarity(p,q) = \frac{\sum_{k=1}^{n} w_k \delta_k s_k}{\sum_{k=1}^{n} \delta_k}$$

$$distance(p,q) = \left(\sum_{k=1}^{n} w_k |p_k - q_k|^r \right)^{1/r}$$

Partitional Clustering

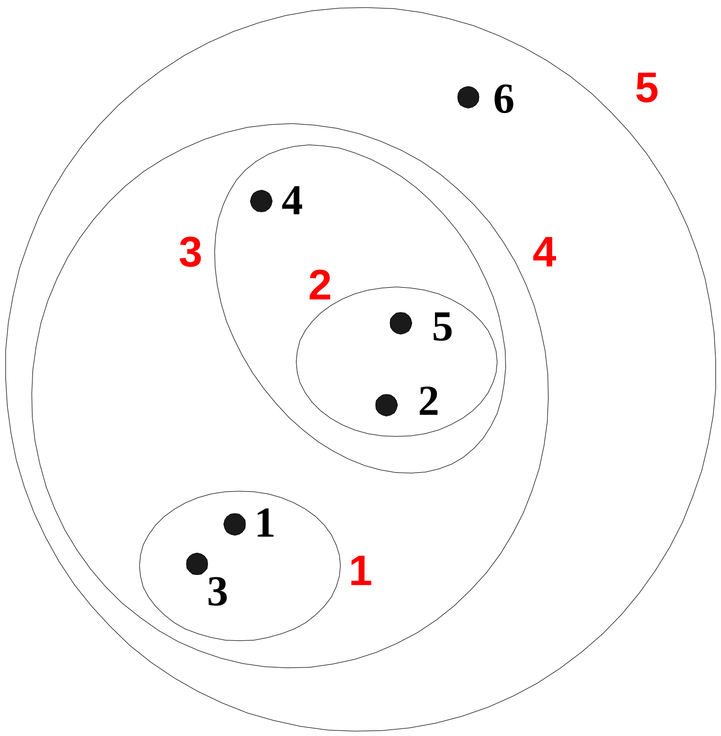


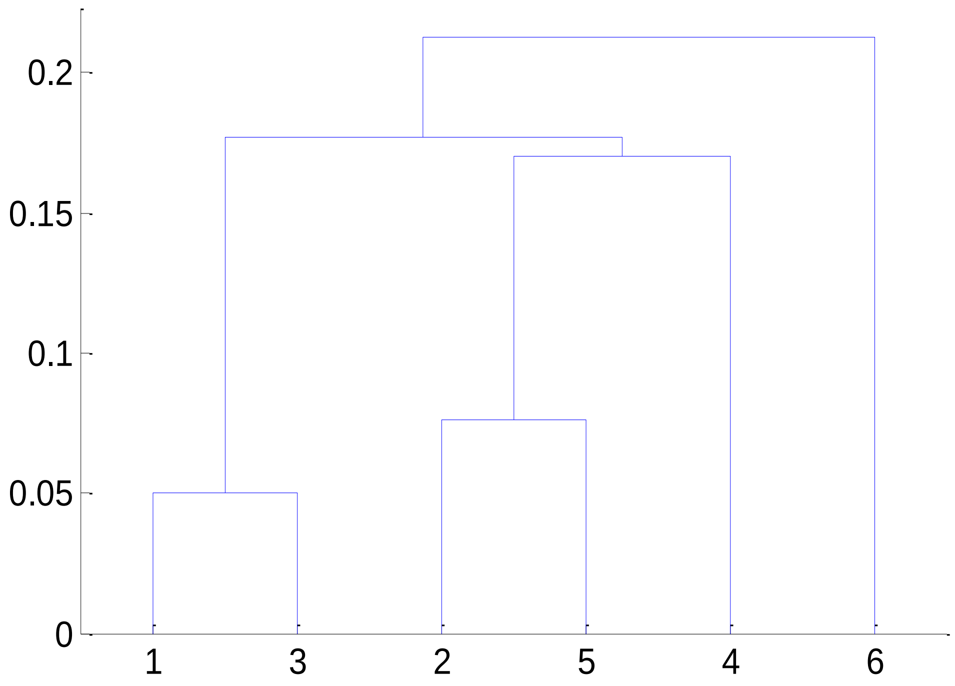
Original Points



A Partitional Clustering

Hierarchical Clustering





Traditional Hierarchical Clustering

Traditional Dendrogram

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.

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

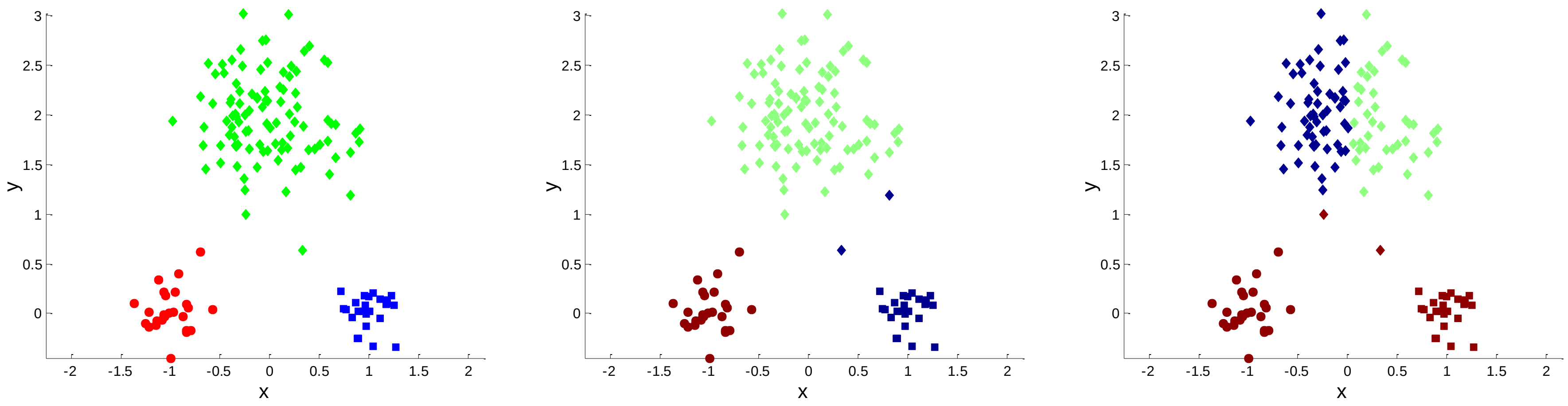
K-means Clustering – Details

- Initial centroids are often chosen randomly.
  - Clusters produced vary from one run to another.
- The centroid is (typically) the mean of the points in the cluster.
- ‘Closeness’ 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

Evaluating K-means Clusters

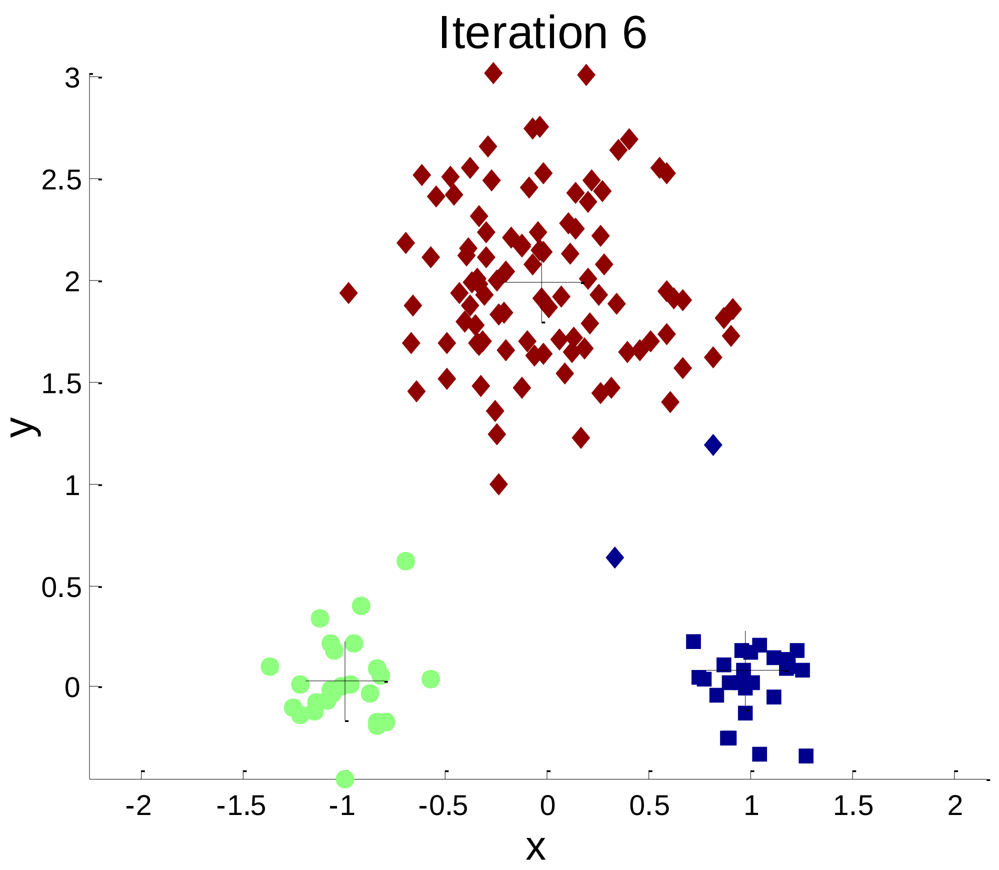
- Most common measure is the *Sum of the Squared Error* (SSE)
  - For each point, the error is the distance to the nearest cluster.
  - To get SSE, we square these errors and sum them.
  - Given two clusters, we can choose the one with the smallest error.
  - One easy way to reduce SSE is to increase K, the number of clusters.
    - A good clustering with smaller K can have a lower SSE than a poor clustering with higher K.

Two different K-means Clustering

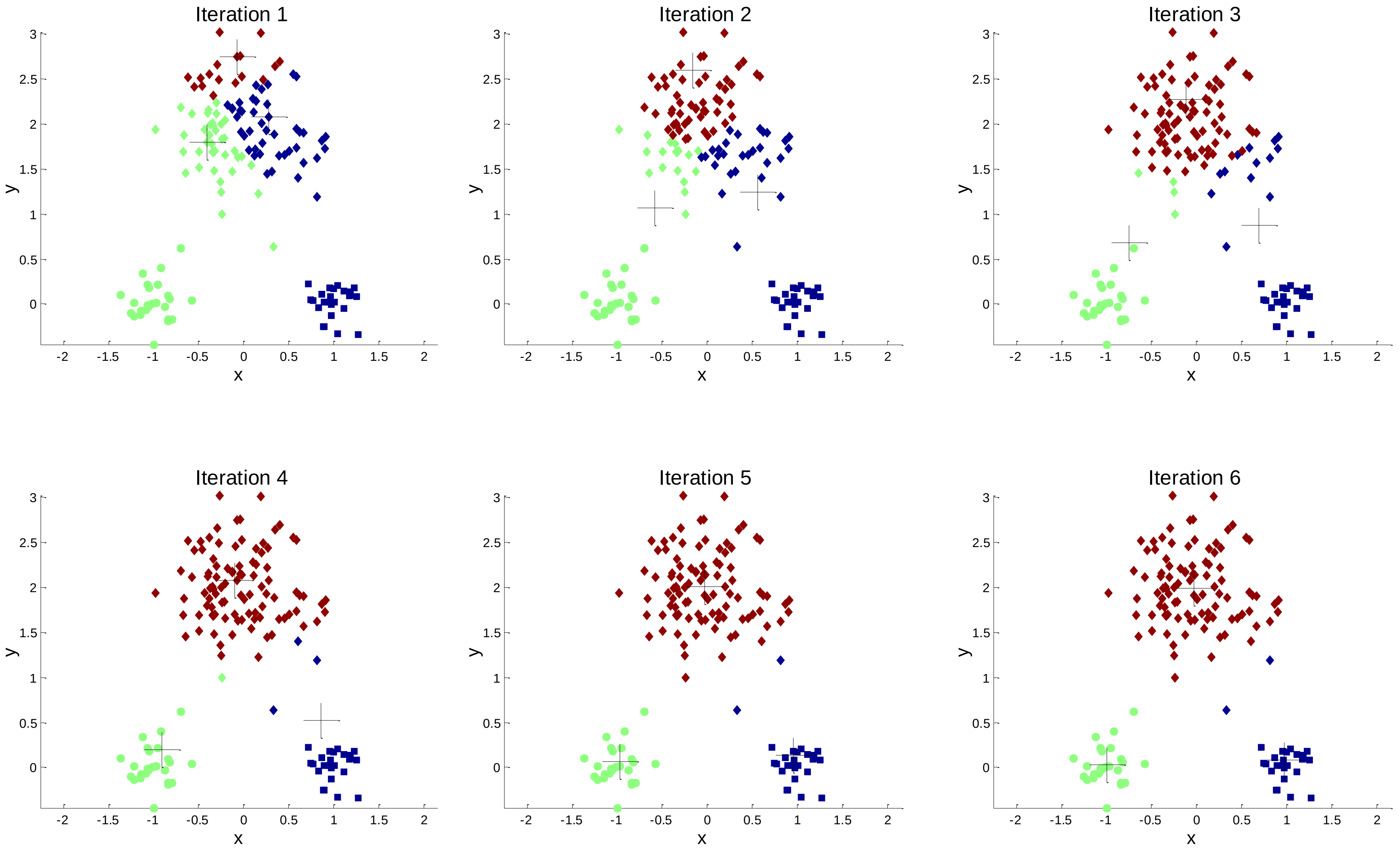


Original Points Optimal Clustering Sub-optimal Clustering

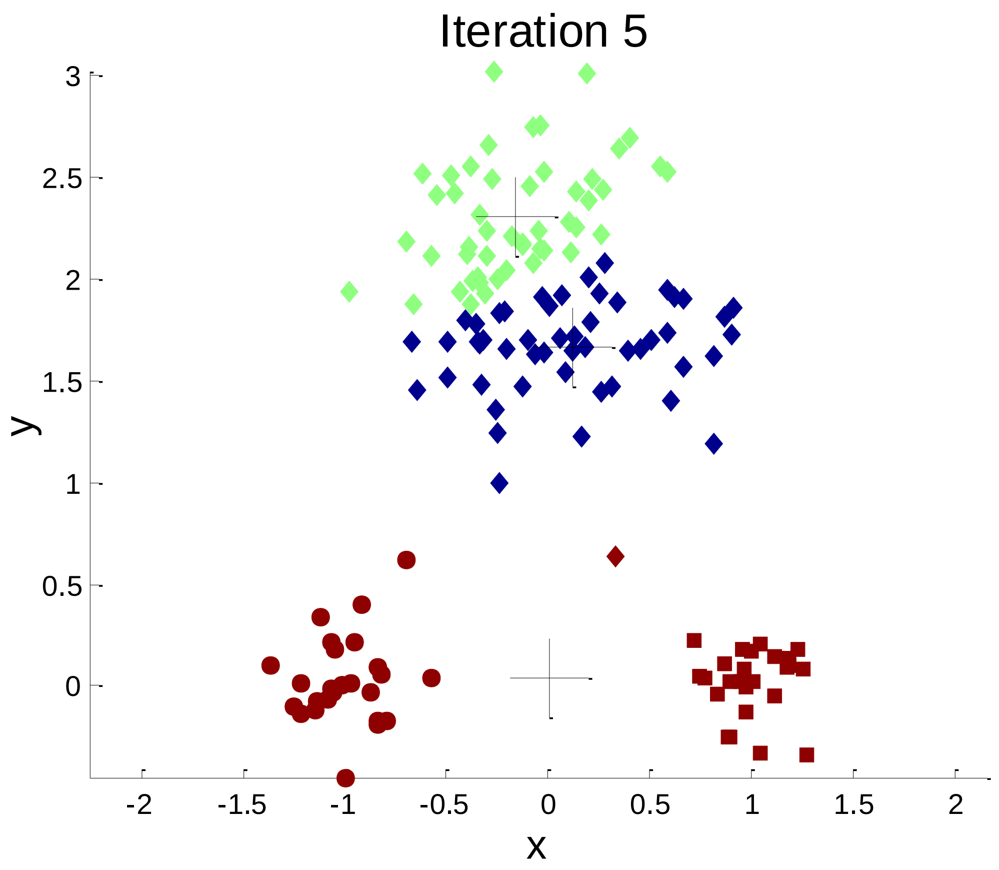
Importance of Choosing - Initial Centroids



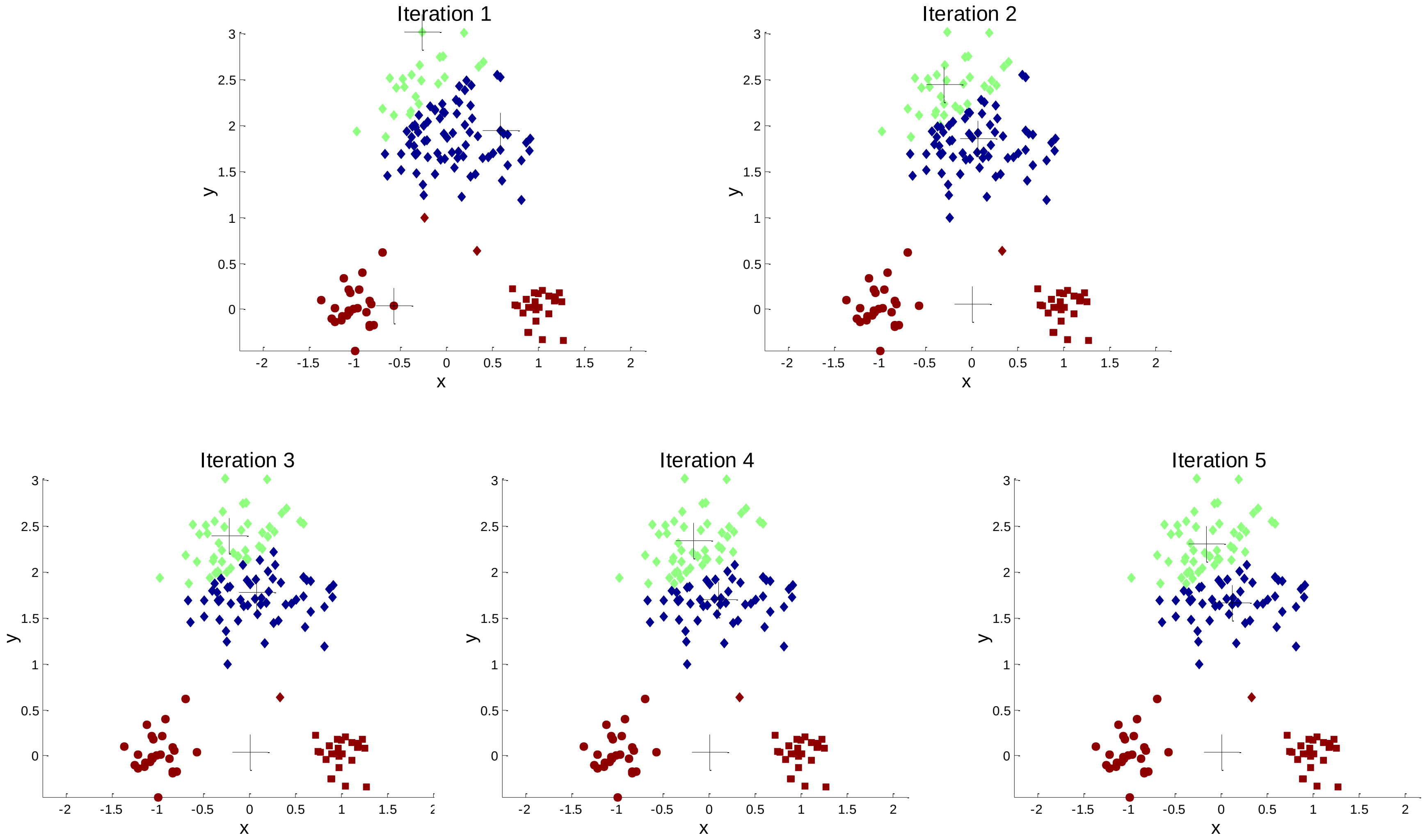
Importance of Choosing - Initial Centroids



Importance of Choosing Initial Centroids …



Importance of Choosing Initial Centroids …



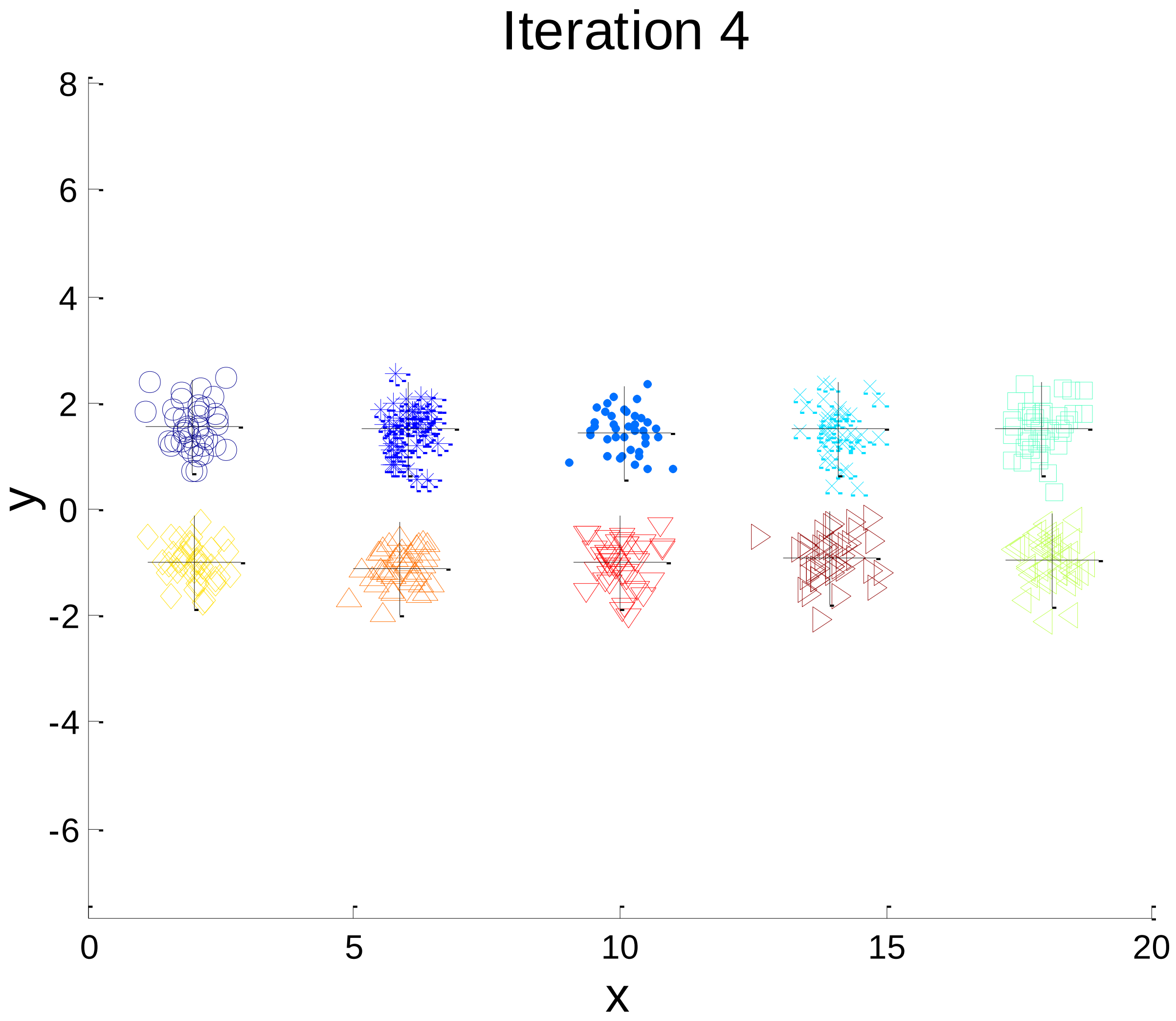
Problems with Selecting Initial Points

- 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

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

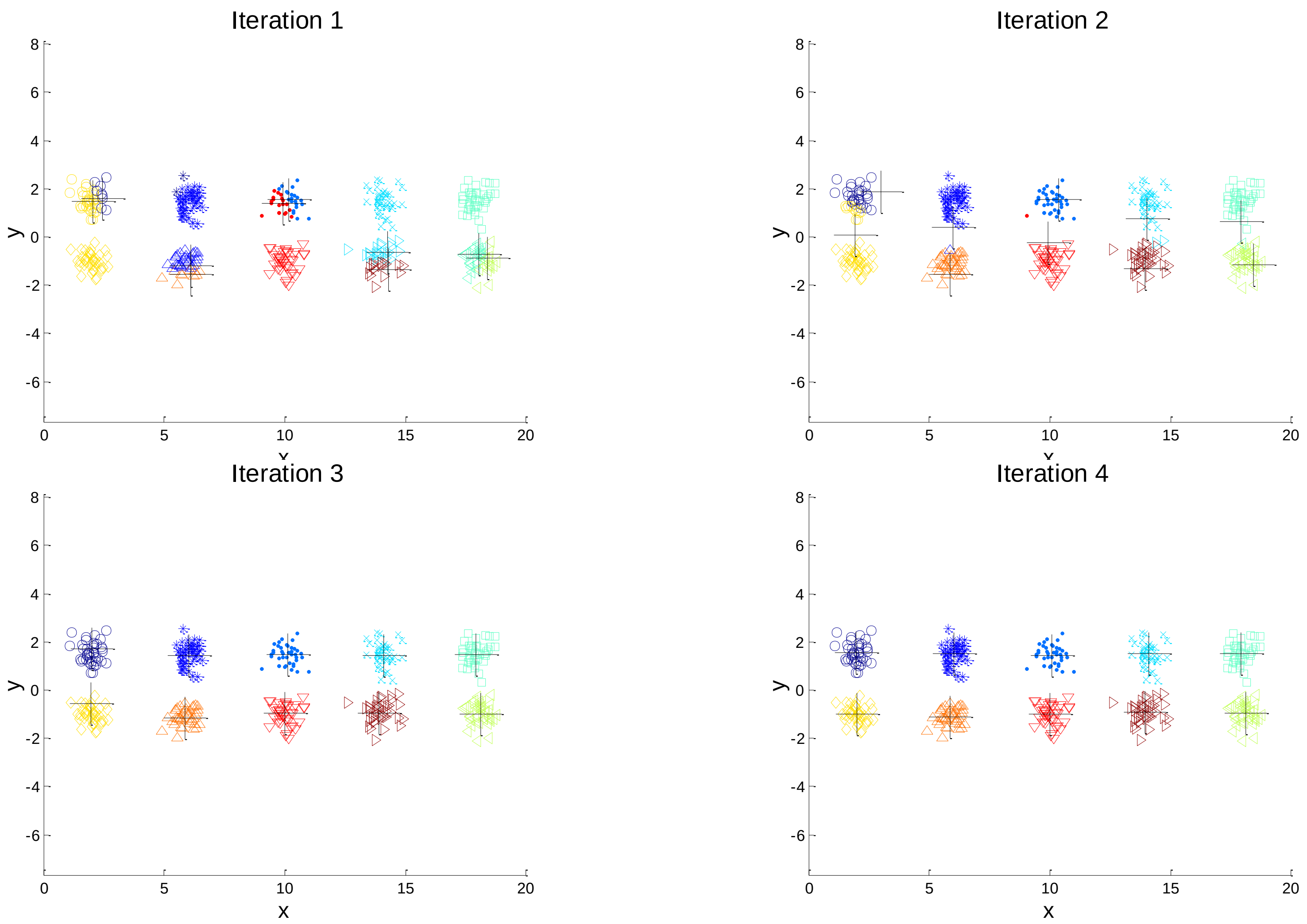
  - For example, if K = 10, then probability = 10!/1010 = 0.00036
  - Sometimes the initial centroids will readjust themselves in ‘right’ way, and sometimes they don’t
  - Consider an example of five pairs of clusters

10 Clusters Example



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

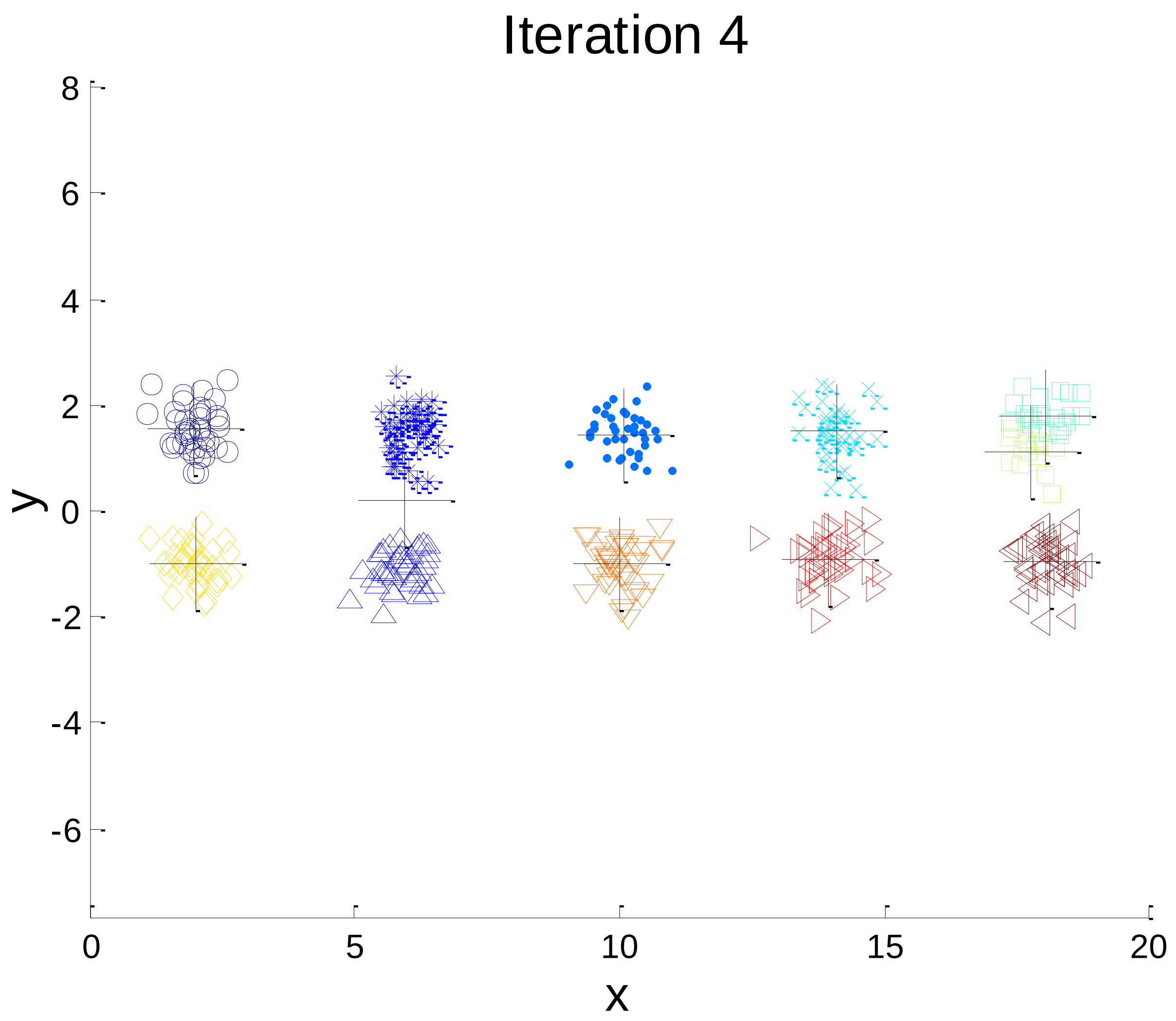
10 Clusters Example



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

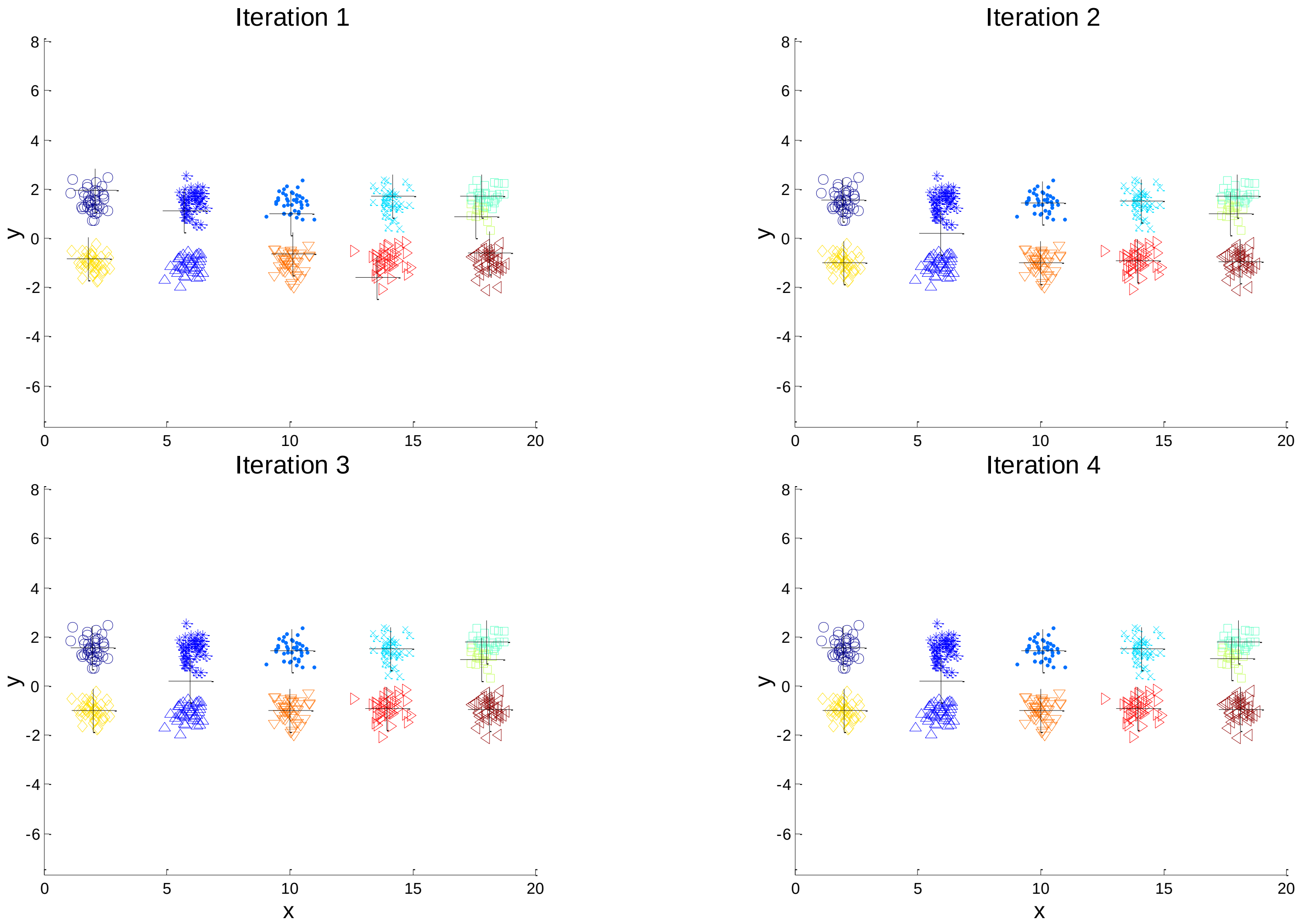
10 Clusters Example

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
- Bisecting K-means
  - Not as susceptible to initialization issues
- 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

Pre-processing and Post-processing

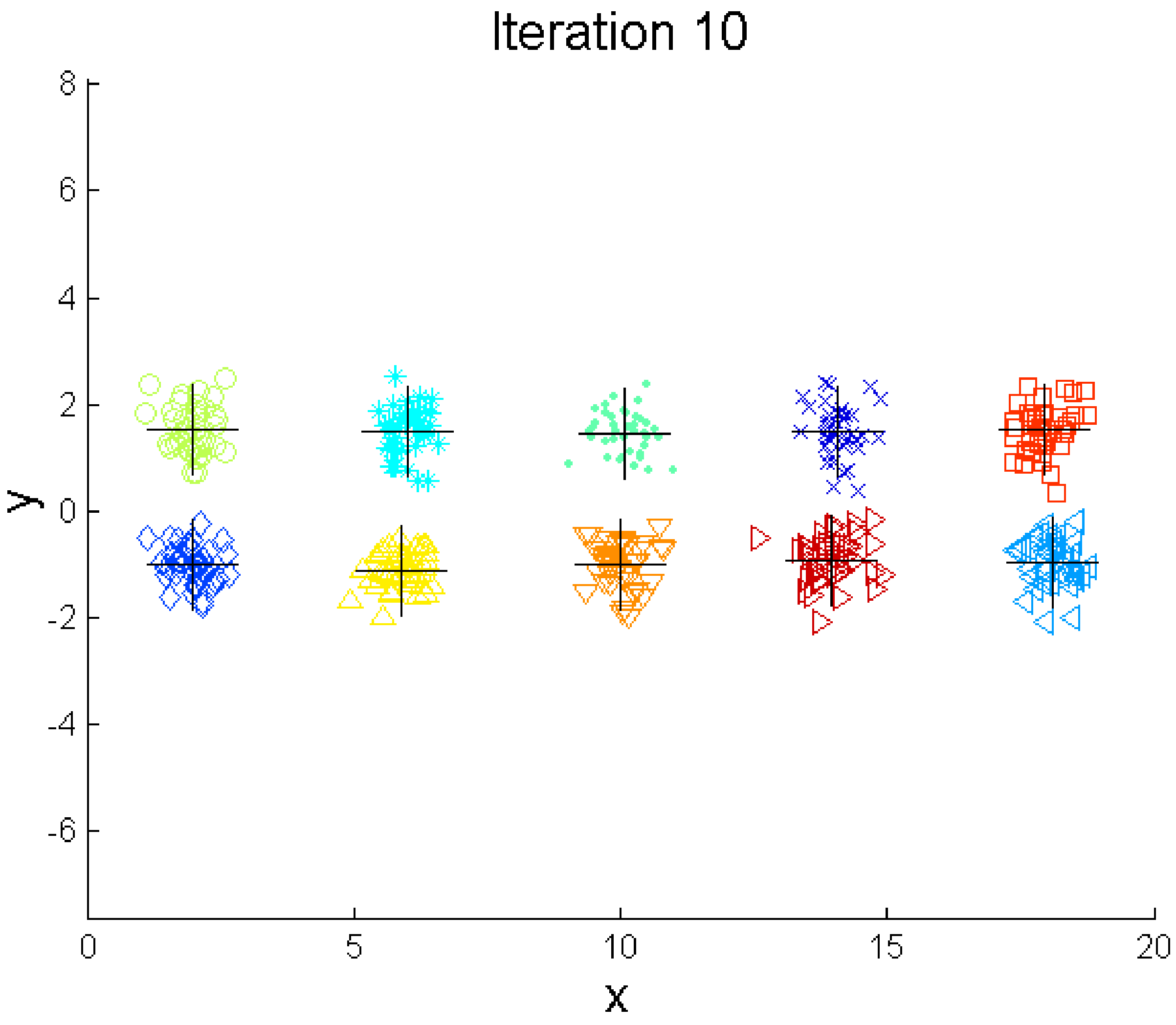
- Pre-processing
  - Normalize data so distance computations are fast.
  - Eliminate outliers
- Post-processing
  - Eliminate small clusters that may represent outliers
  - Split ‘loose’ clusters, i.e., clusters with relatively high SSE
  - Merge clusters that are ‘close’ and that have relatively low 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

```
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
```

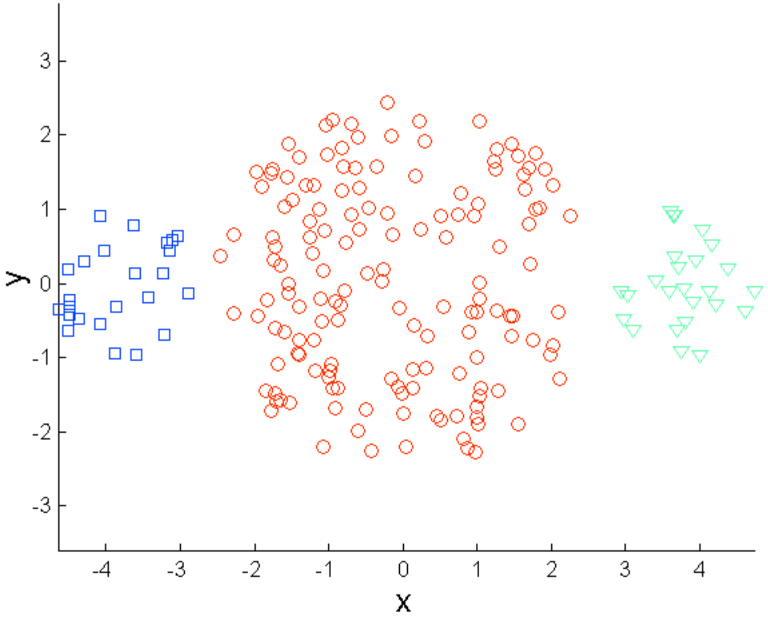
Bisecting K-means Example



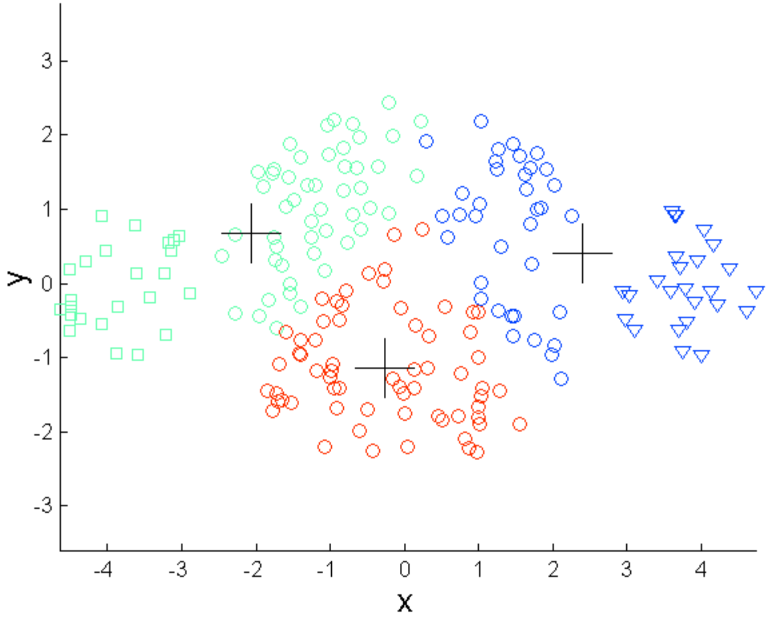
Limitations of K-means

- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes
- K-means has problems when the data contains outliers.
- One solution is to use many clusters.
  - Find parts of clusters, but need to put together.

Limitations of K-means: Differing Sizes

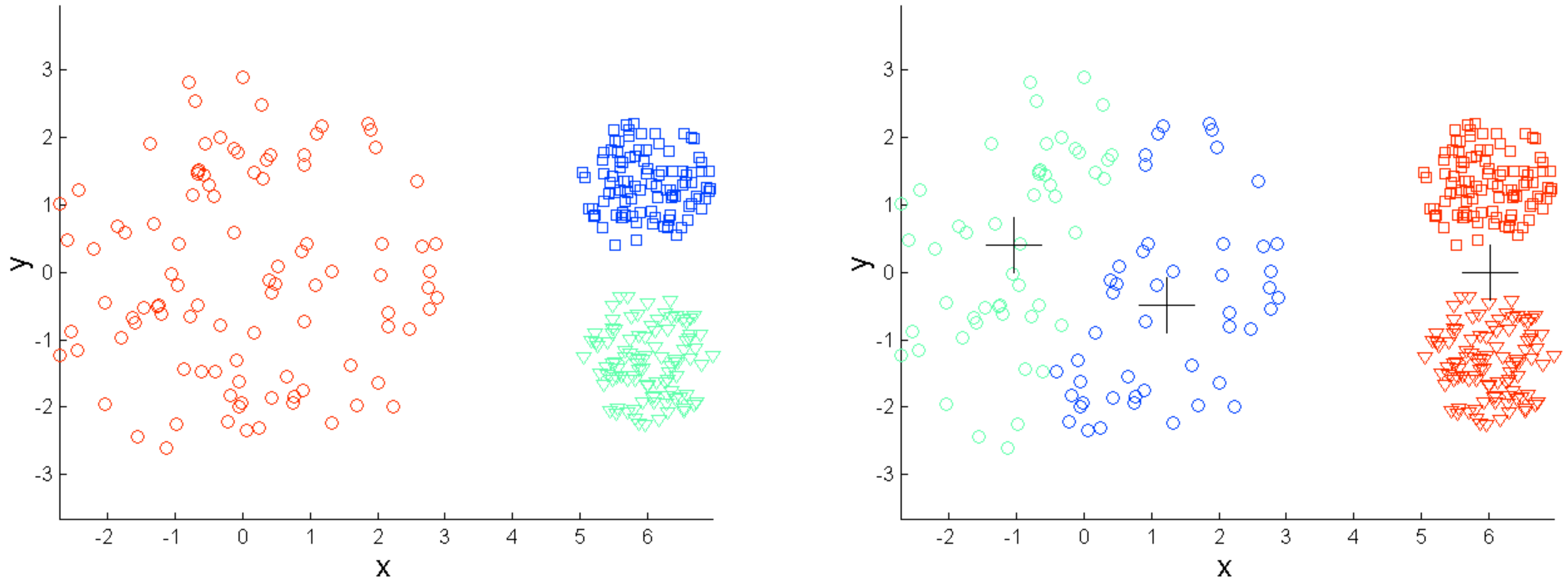


Original Points



K-means Clusters

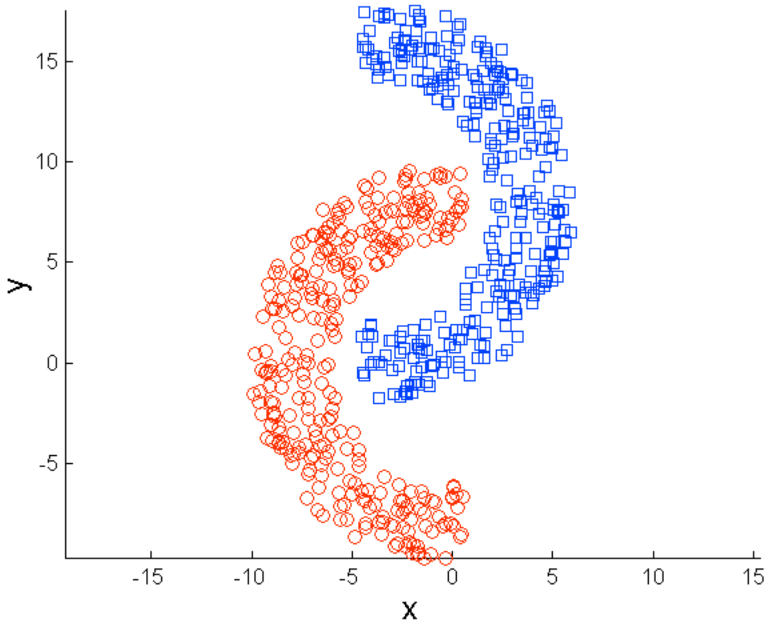
Limitations of K-means: Differing Density



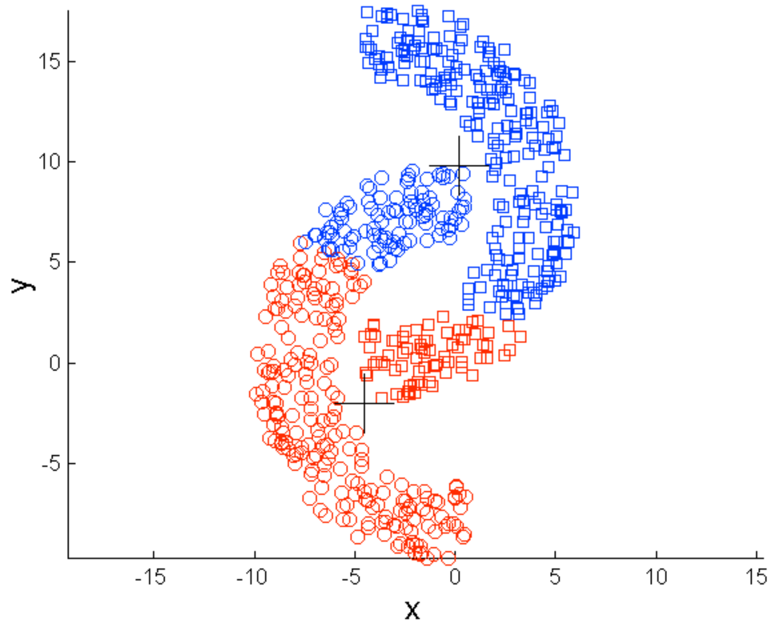
Original Points

K-means Clusters

Limitations of K-means: Non-globular Shapes

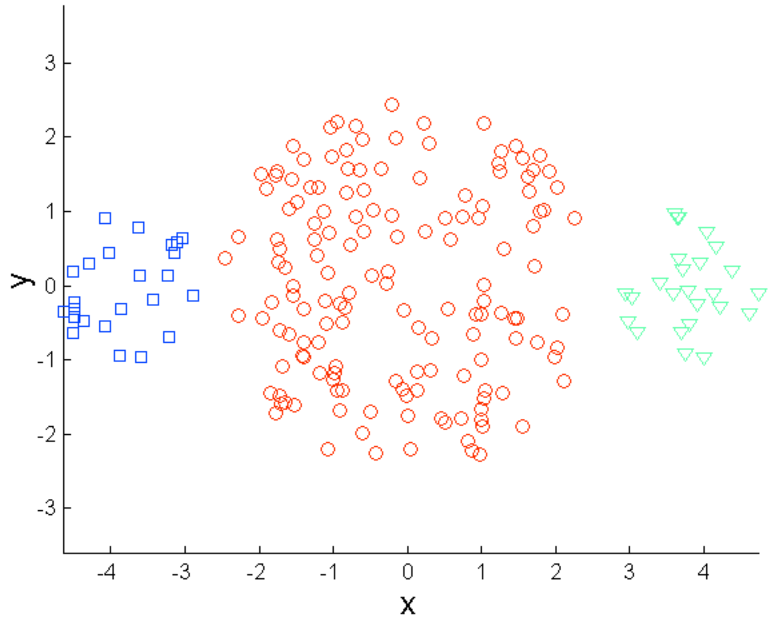


Original Points

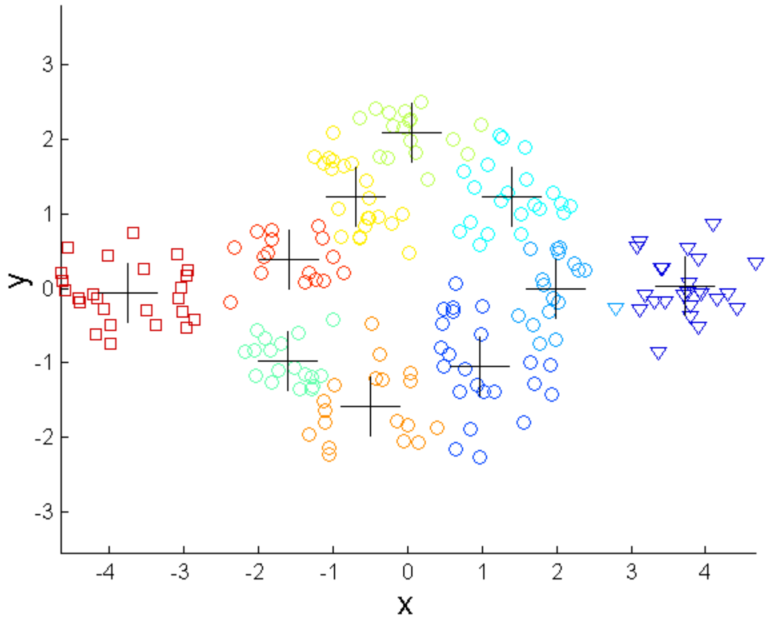


K-means Clusters

Overcoming K-means Limitations

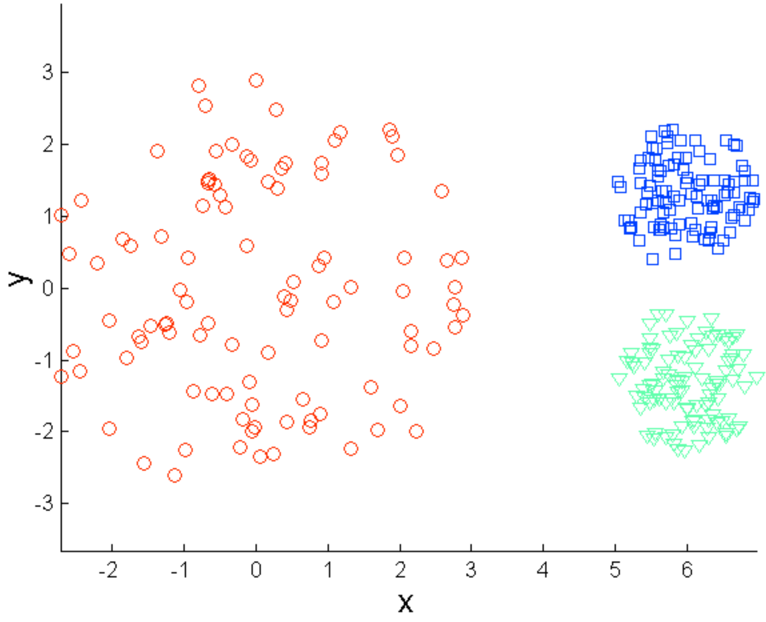


Original Points

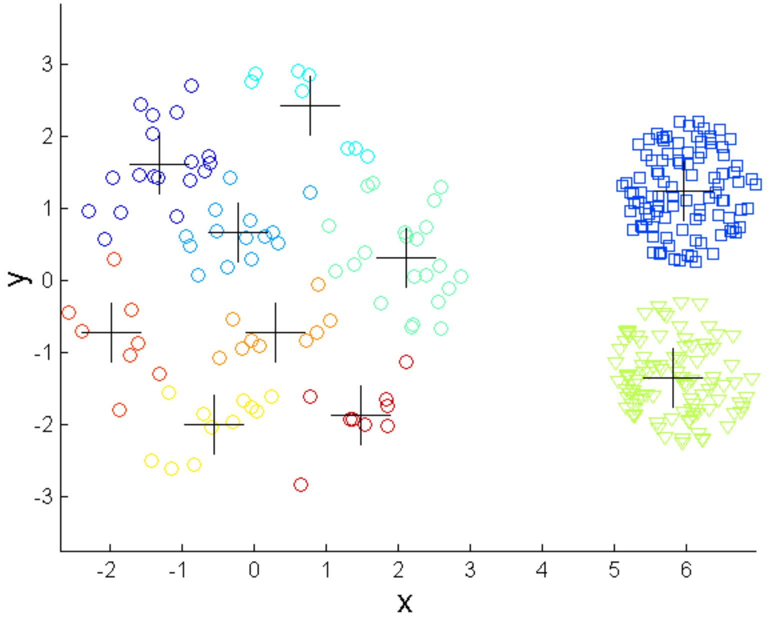


K-means Clusters

Overcoming K-means Limitations

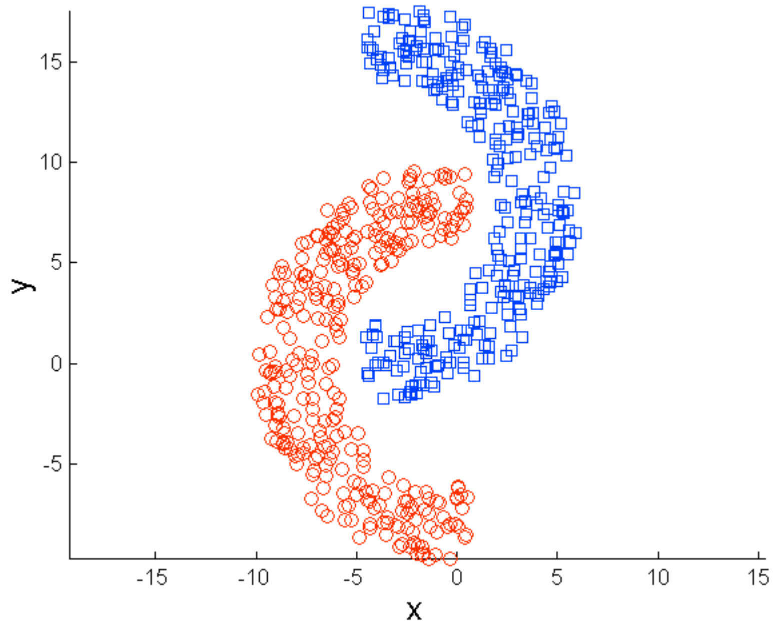


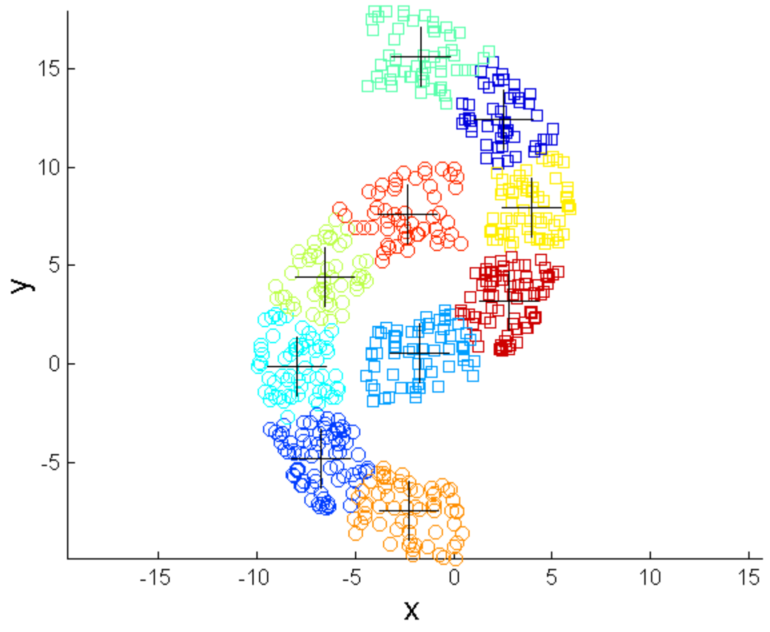
Original Points



K-means Clusters

Overcoming K-means Limitations





Original Points

K-means Clusters