Clustering (Unsupervised Learning)

David Li

Outline

- Clustering Analysis
  - Introduction, Distance
  - K-Means Clustering
  - Hierarchical Clustering
  - DBSCAN Clustering

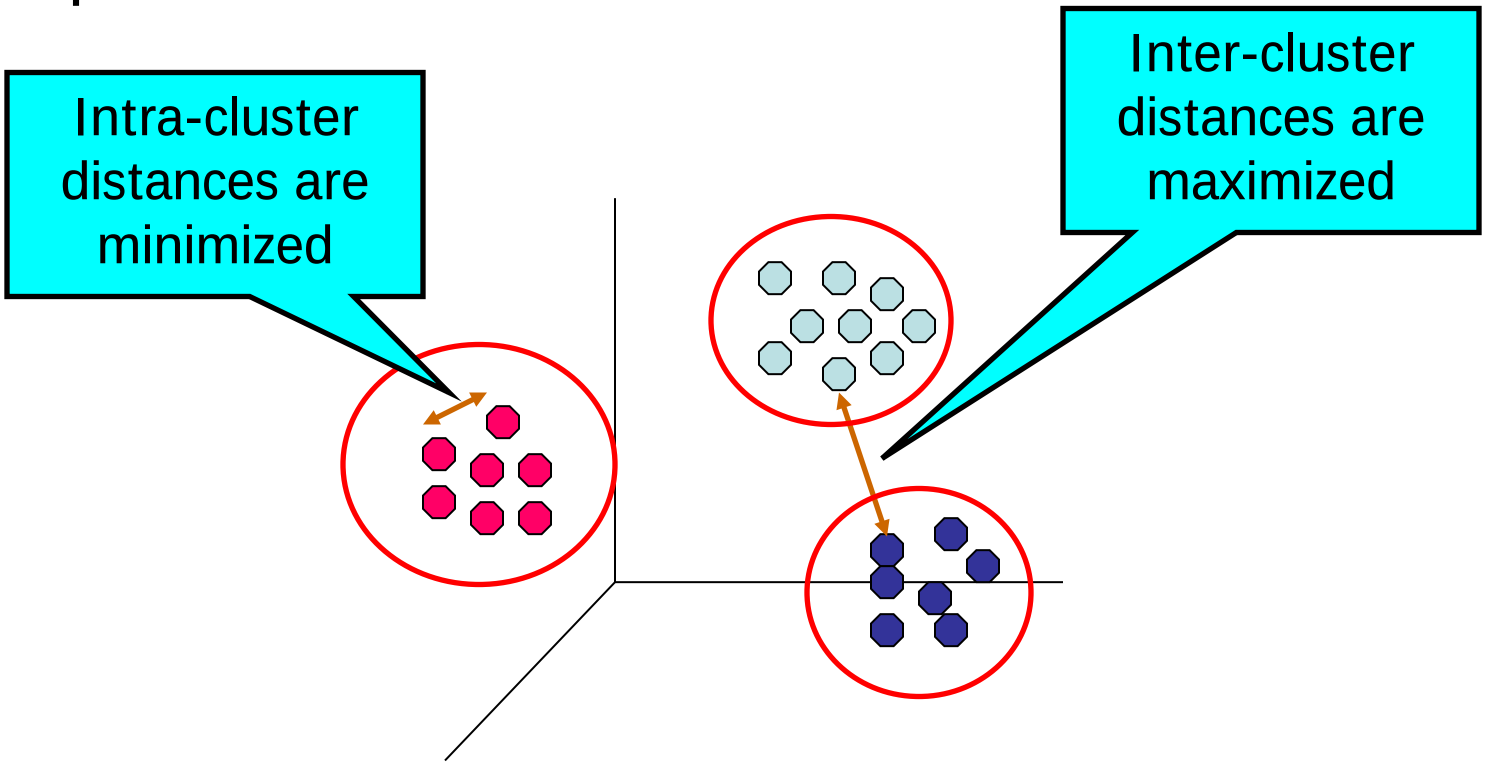
Typical Data Analysis

Three main types of statistical problems associated with most data analysis:

- Identification of important features that characterize the data (sample classes) (feature or variable selection).
- Identification of new/unknown sample classes using data (unsupervised learning – clustering)
- Classification of sample into known classes (supervised learning – classification)

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



Clustering

- Clustering is an exploratory tool to see who's running with who: Features and Samples.
- "Unsupervised"
- NOT for classification of samples. Clustering algorithms assign (or predict) a number to each data point, indicating which cluster a particular point belongs to.
- NOT for identification of important features.

Applications of Clustering

- Viewing and analyzing vast amounts of data as a whole set can be perplexing

- It is easier to interpret the data if they are partitioned into clusters by combining similar data points.

- Identification of outliers

Clustering algorithms

The types of clustering methods:

- Hierarchical Clustering Methods
  - Agglomerative *hierarchical* clustering
  - Divisive clustering
- Model Based Clustering Methods
  - COBWEB, Gaussian mixtures
- Grid Based Clustering Methods
  - STING. Wave Cluster and CLIQUE
- Density Based Clustering Methods:
  - *DBSCAN*
- Partition Clustering Methods
  - *K-Means*, K-Mediods

Distance

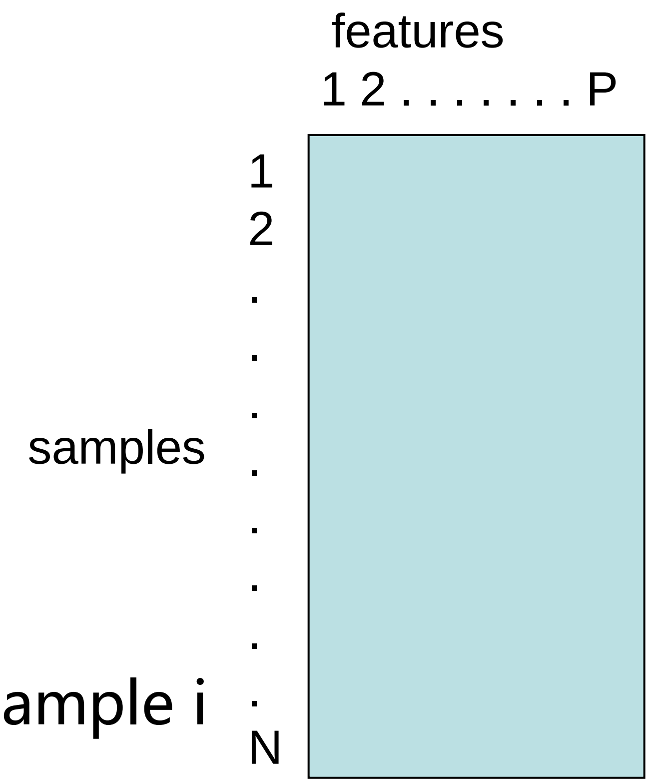
- We need a mathematical definition of distance between two points
  - Manhattan, Euclidian, Cosine, Correlation, etc
- What are points?
  - A sample’ s all observed values of features
  - A student’ s all quiz/exam grades
- What is the mathematical definition of a point?
  - The vector of features (X1, X2, X3, … Xn)
  - Like a row in table where each row is a point and columns are the features of point.

Points

- feature1= $(E_{11}, E_{21}, \ldots, E_{N1})'$
- feature2= $(E_{12}, E_{22}, \ldots, E_{N2})'$

- Sample1= $(E_{11}, E_{12}, \ldots, E_{1P})'$
- Sample2= $(E_{21}, E_{22}, \ldots, E_{2P})'$

- E_{ij}= observed value of feature j on sample i

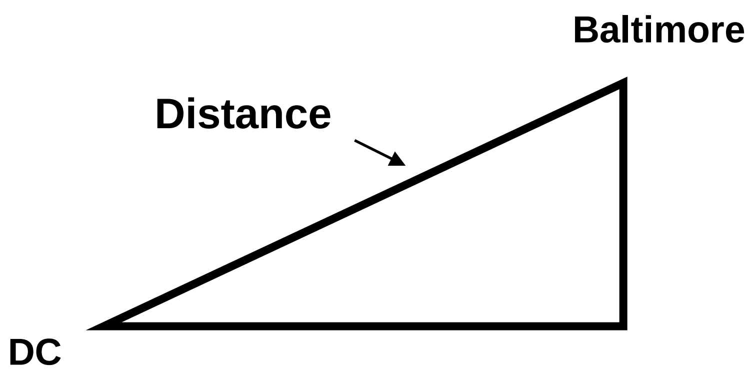


DATA MATRIX

Most Famous Distance

- Euclidean distance
  - Example distance between sample 1 and 2:
  - Sqrt of Sum of $(E_{1i} - E_{2i})^2$, $i=1,\ldots,P$

- When N is 2, this is distance as we know it:



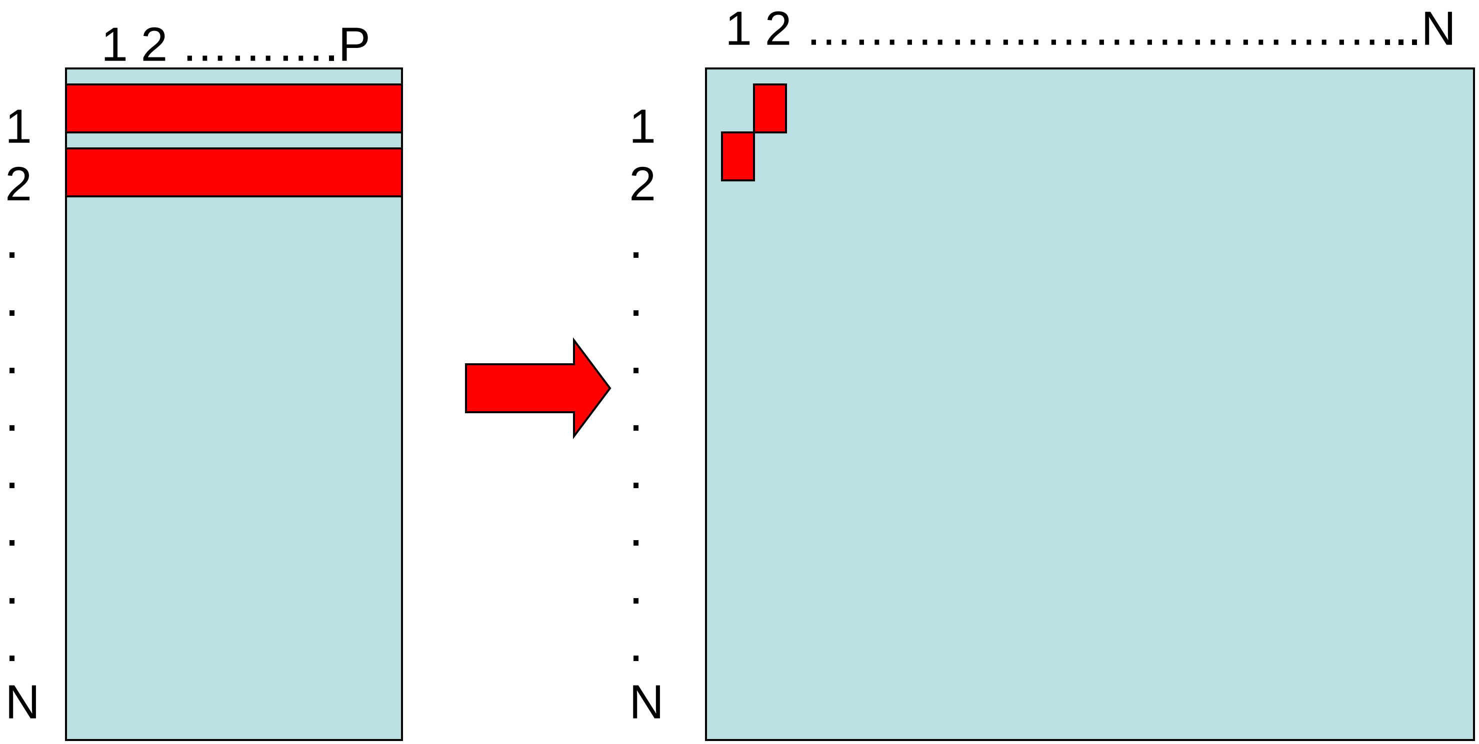
When *P* is 20,000 you have to think abstractly

Correlation can also be used to compute distance

- Pearson Correlation
- Spearman Correlation
- Uncentered Correlation
- Absolute Value of Correlation

See http://gedas.bizhat.com/dist.htm for details for your interest.

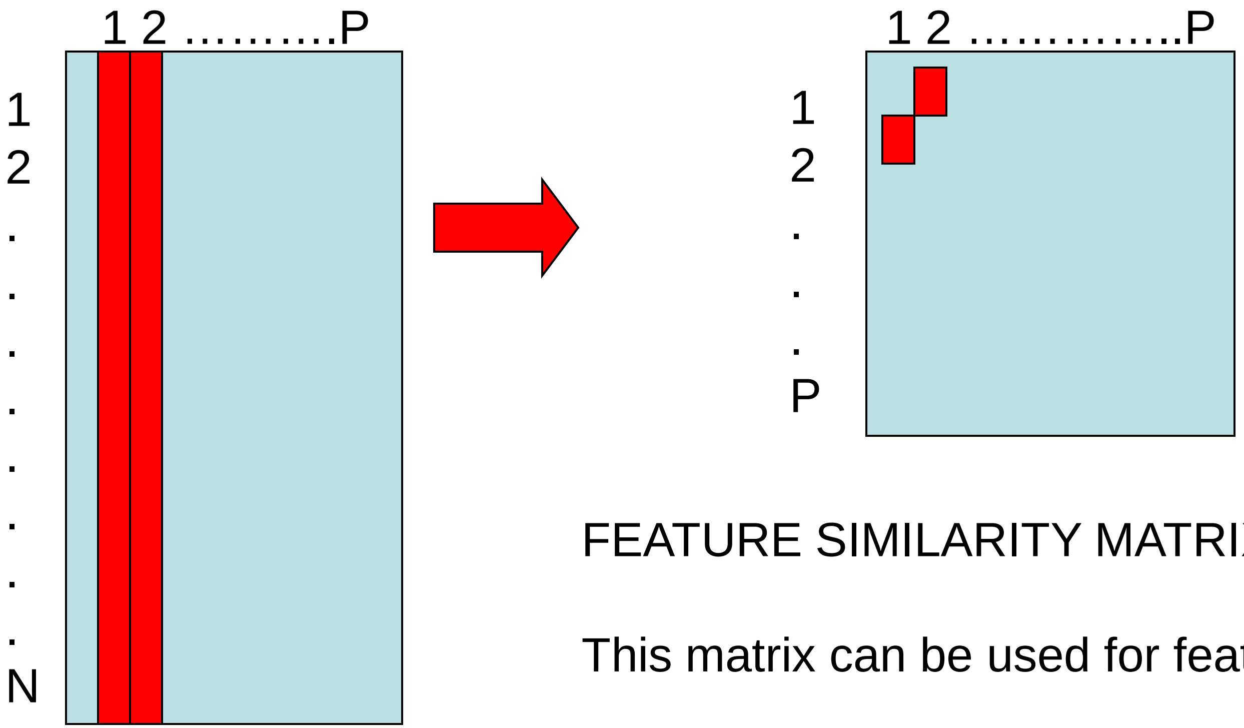
The similarity/distance matrices



DATA MATRIX

SAMPLE SIMILARITY MATRIX

The similarity/distance matrices



FEATURE SIMILARITY MATRIX

This matrix can be used for feature selection

DATA MATRIX

Feature/Sample Selection

- Do you want all features included?
- Irrelevant features will affect your results.
- Including all features: dendrogram can't all be seen at the same time.
- Perhaps screen the features?

Three commonly seen clustering approaches

- K-means/K-medoids
  - Partitioning method
  - Requires user to define K = # of clusters a priori
  - No picture to (over) interpret
- Hierarchical clustering
  - Dendrogram
  - Allows us to cluster both features and samples in one picture and see whole dataset “organized”
- DBSCAN（density based spatial clustering of applications with noise）
  - Identifying points that are in dense regions of the feature space, where many data points are close together.

K-means Clustering

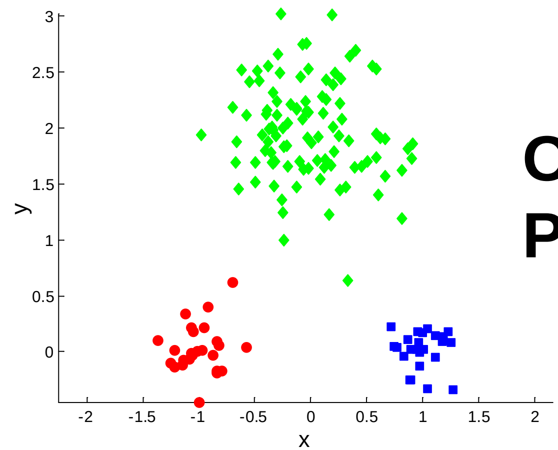
- Partition clustering method
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified (how do you know K?
- The basic algorithm is very simple

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

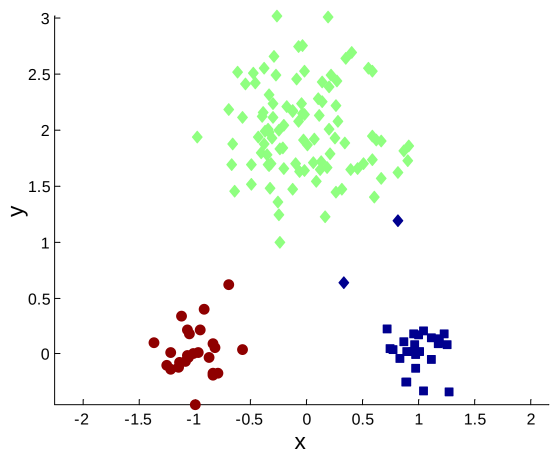
K-means Clustering – Details

- 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 "distance", e.g. Euclidean distance, 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'

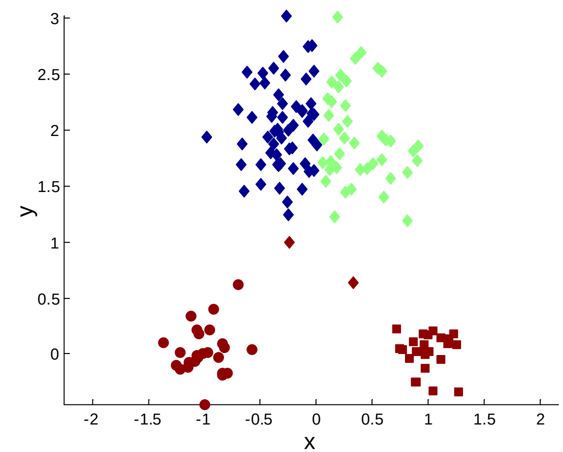
Two different K-means Clustering



Original Points

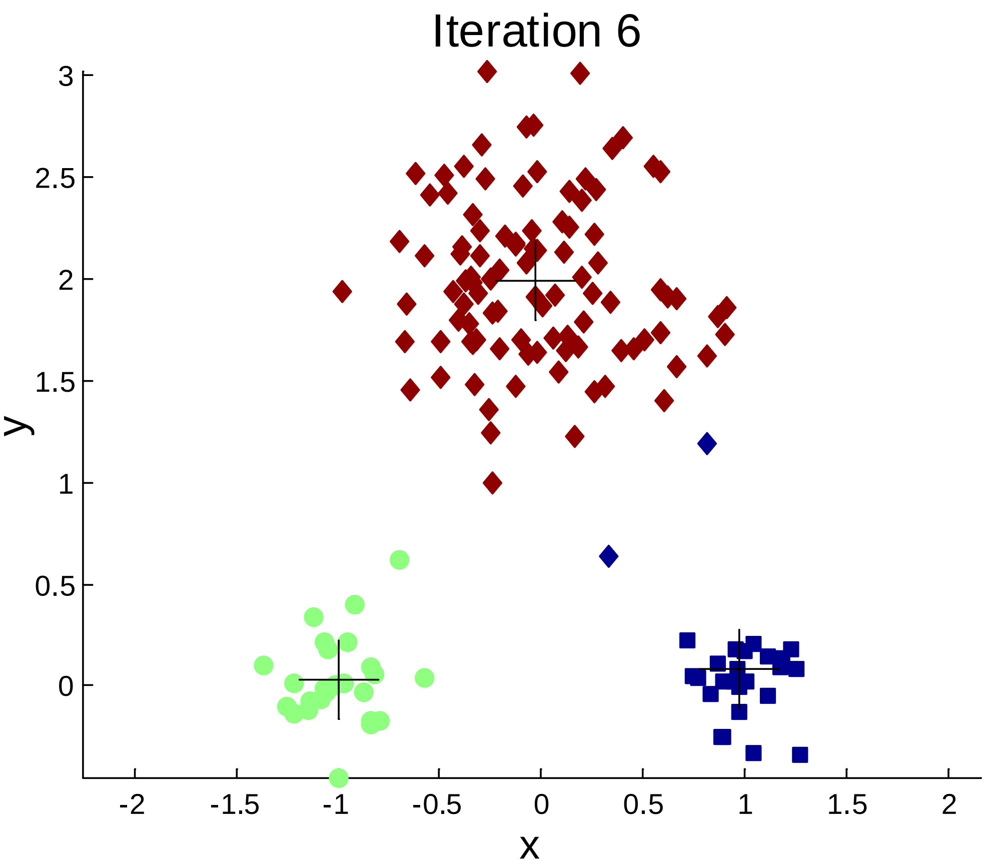


Optimal Clustering

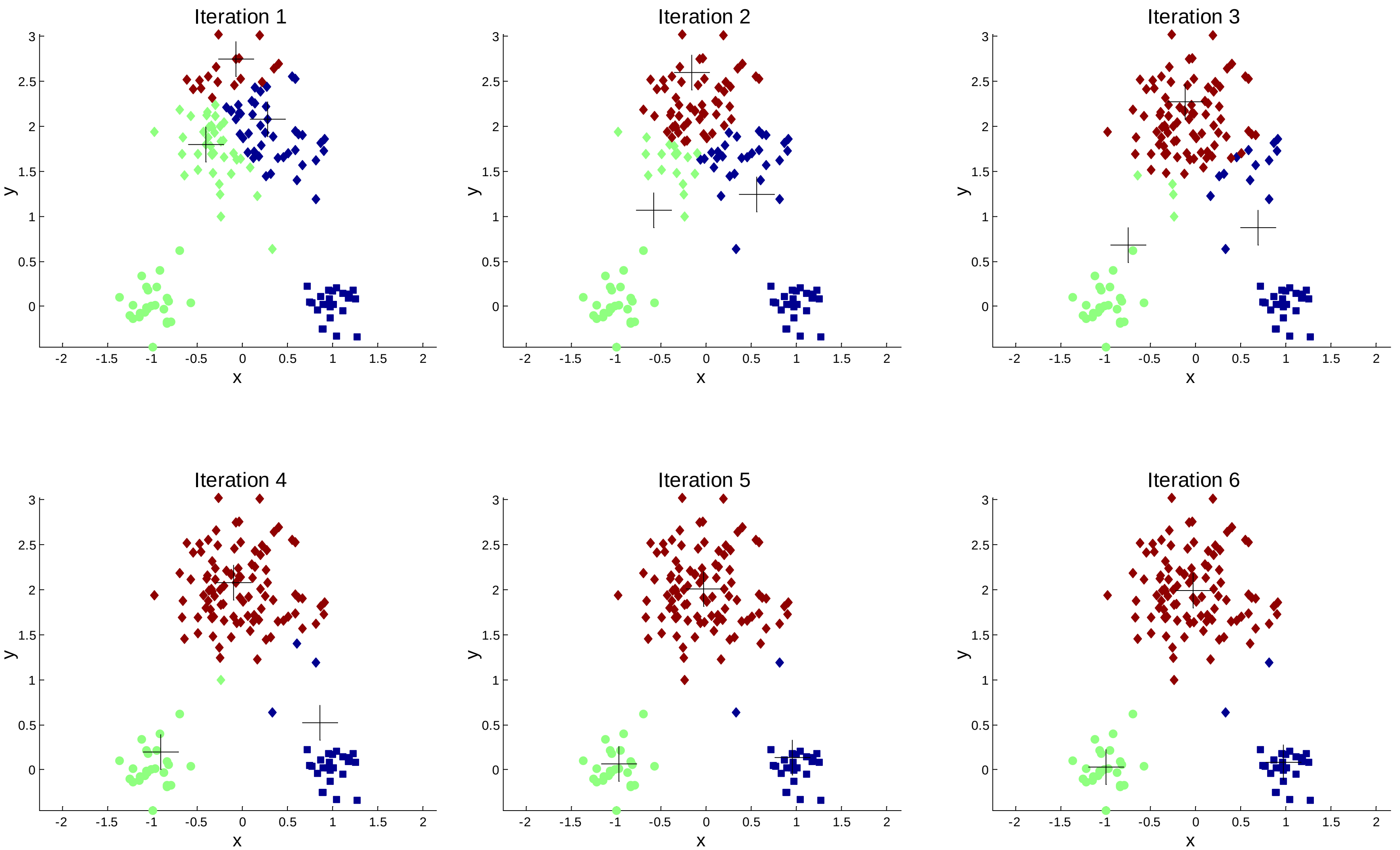


Sub-optimal Clustering

Importance of Choosing Initial Centroids



Importance of Choosing Initial Centroids



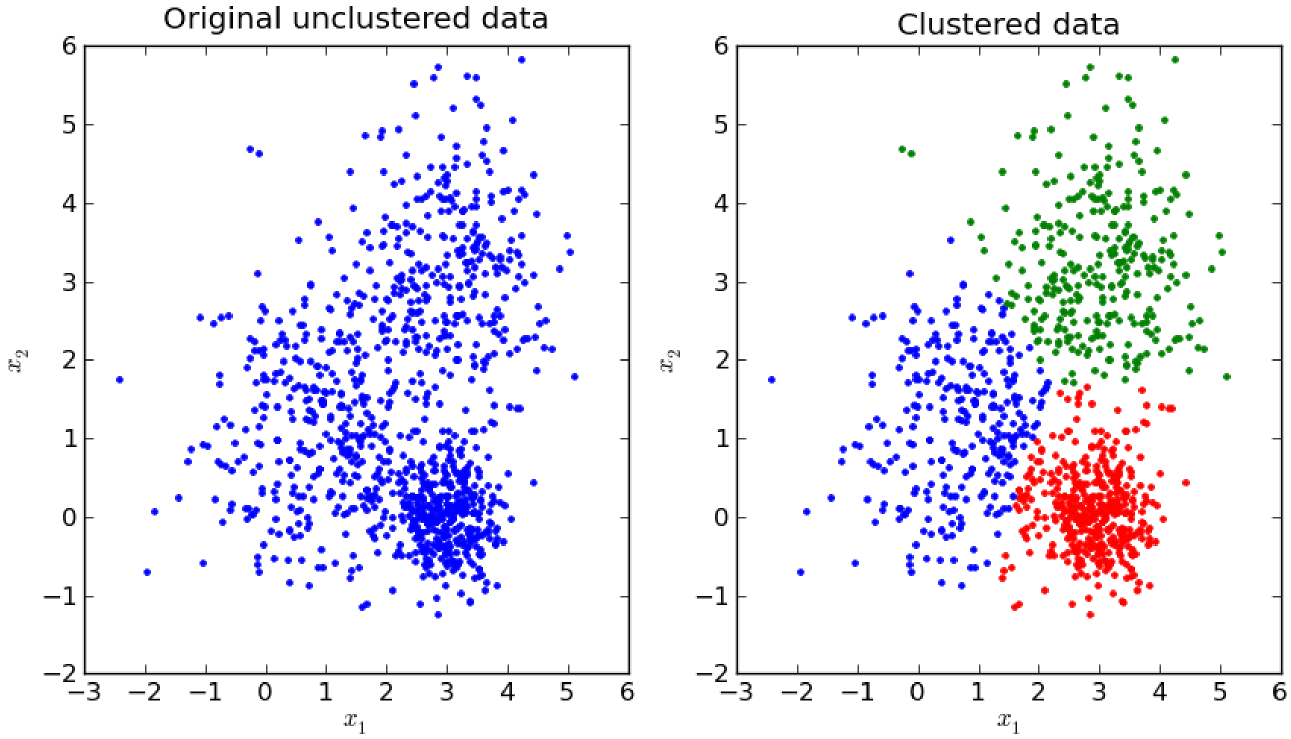
Evaluating K-means Clusters

- 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 and m_i is the representative point for cluster C_i
    - can show that m_i corresponds to the center (mean) of the cluster
  - 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

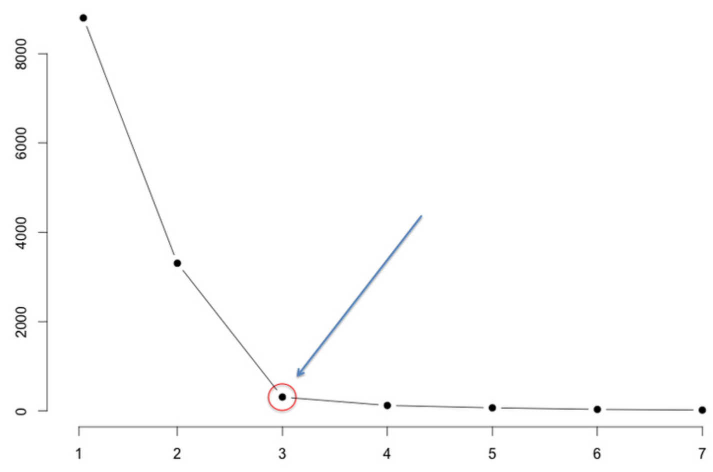
How do you know the optimal K?



- How do you know K=3?
- Is 3 the optimal K?

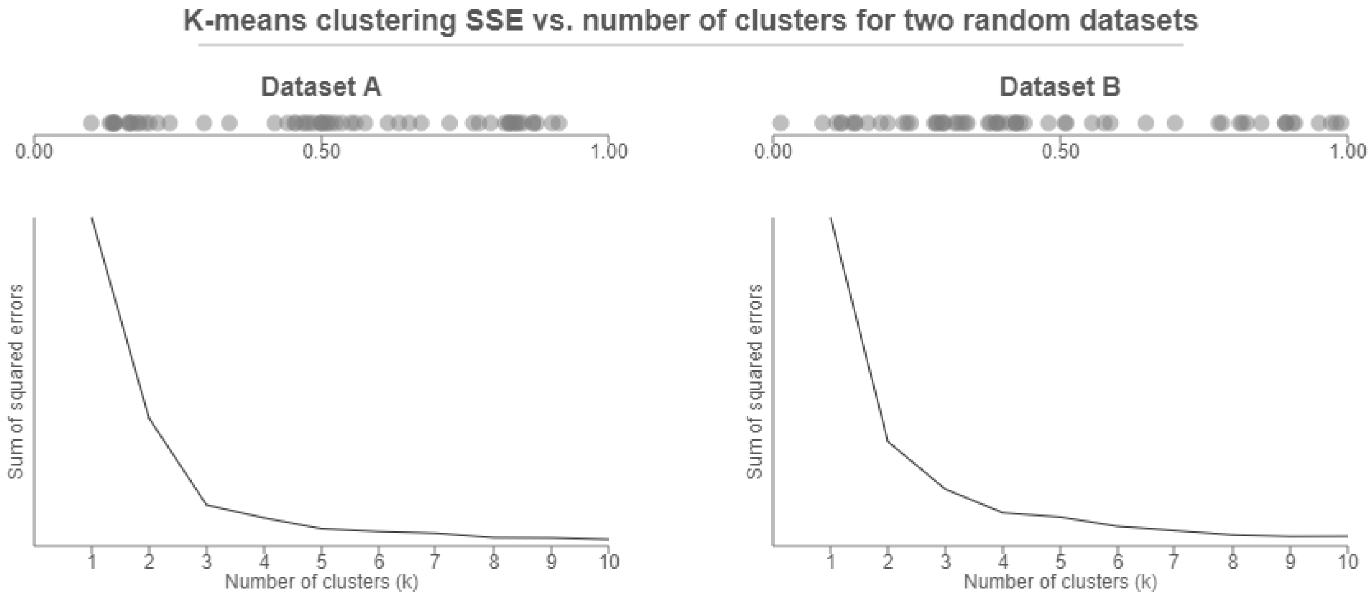
Determine the Optimal K - Elbow Method

- There are two methods to find the K in k-means
  - The Elbow Method
  - The Silhouette Method
- Elbow Method
  - run the algorithm for different values of K(say K = 10 to 1)
  - plot the K values against SSE(Sum of Squared Errors).
  - select the value of K for the elbow point as shown in the figure, i.e., choose the k for which SSE becomes first starts to diminish.



Determine the Optimal K - Elbow Method

However, the elbow may not be always clear and sharp.
We could choose k to be either 3 or 4.



In such an ambiguous case, we may use the Silhouette Method.

Determine the Optimal K - Silhouette Method

- The silhouette value measures how similar a point is to its own cluster (cohesion) compared to other clusters (separation).
- The range of the Silhouette value is between +1 and -1.
- A **high value is desirable** and indicates that the point is placed in the correct cluster.
- If many points have a negative Silhouette value, it may indicate that we have created too many or too few clusters.

Determine the Optimal K - Silhouette Method

•The Silhouette Value ***s(i)*** for each data point ***i*** is defined as follows:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \text{ if } |C_i| > 1$$

and

$$s(i) = 0, \text{ if } |C_i| = 1$$

•***Note****: s(i) is defined to be equal to zero if* ***i*** *is the only point in the cluster. This is to prevent the number of clusters from increasing significantly with many single-point clusters.*

Determine the Optimal K - Silhouette Method

•***a(i)*** is the measure of similarity of the point ***i*** to its own cluster. It is measured as the average distance of ***i*** from other points in the cluster.

For data point $i \in C_i$ (data point i in the cluster C_i), let

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j)$$

•***b(i)*** depicts average nearest cluster distance i.e. average distance to the instances of the next closest cluster.

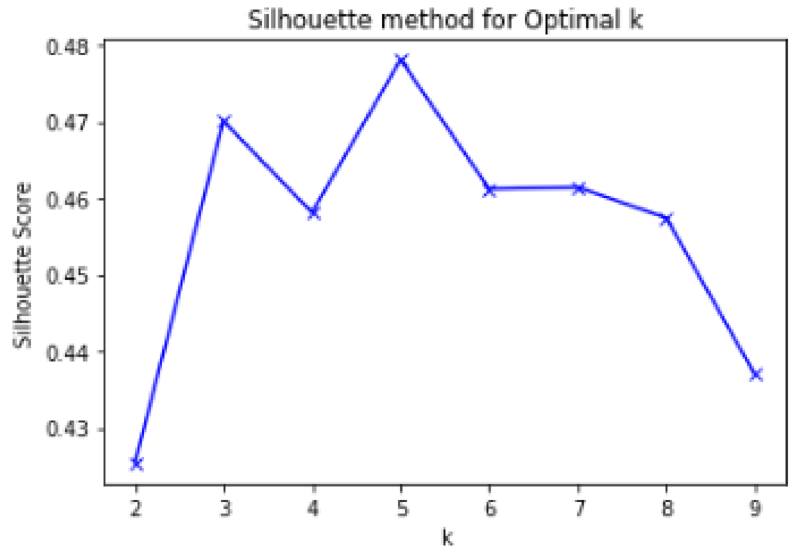
For each data point $i \in C_i$, we now define

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j)$$

d(i, j) *is the distance between points* ***i*** *and* ***j****. It can be any distance metric.*

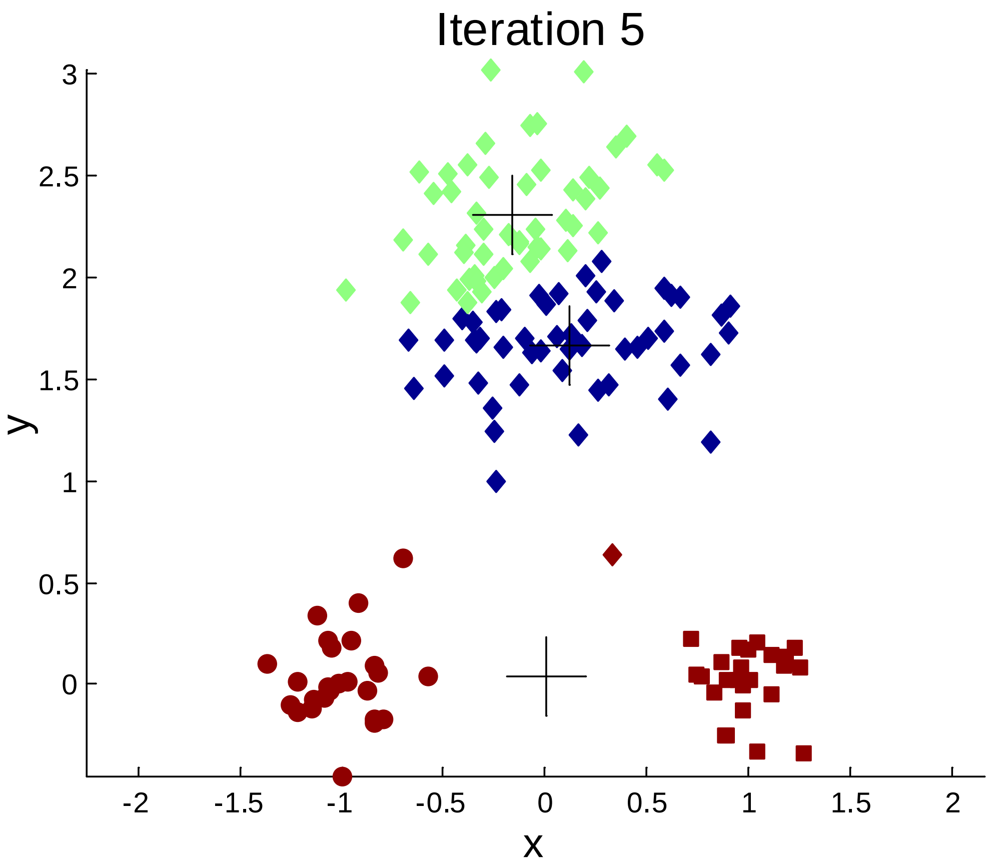
Determine the Optimal K - Silhouette Method

- High Silhouette Score is desirable.
- The Silhouette Score reaches its ***global maximum at the optimal k***.
- This should ideally appear as a peak in the Silhouette Value-versus-k plot.

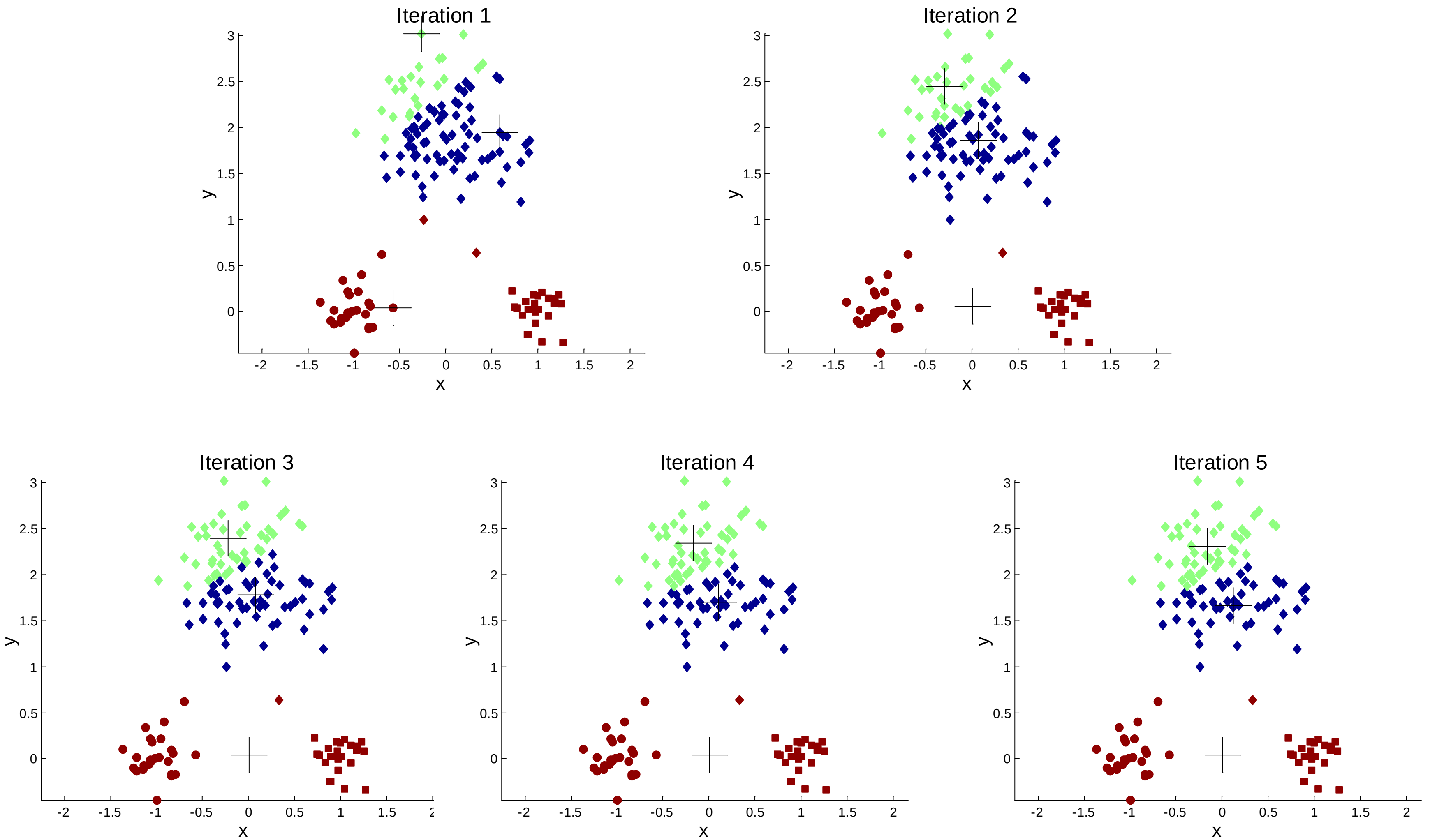


- As per this method k=3 was a local optima, whereas k=5 should be chosen for the number of clusters.

Importance of Choosing Initial Centroids …



Importance of Choosing Initial Centroids …



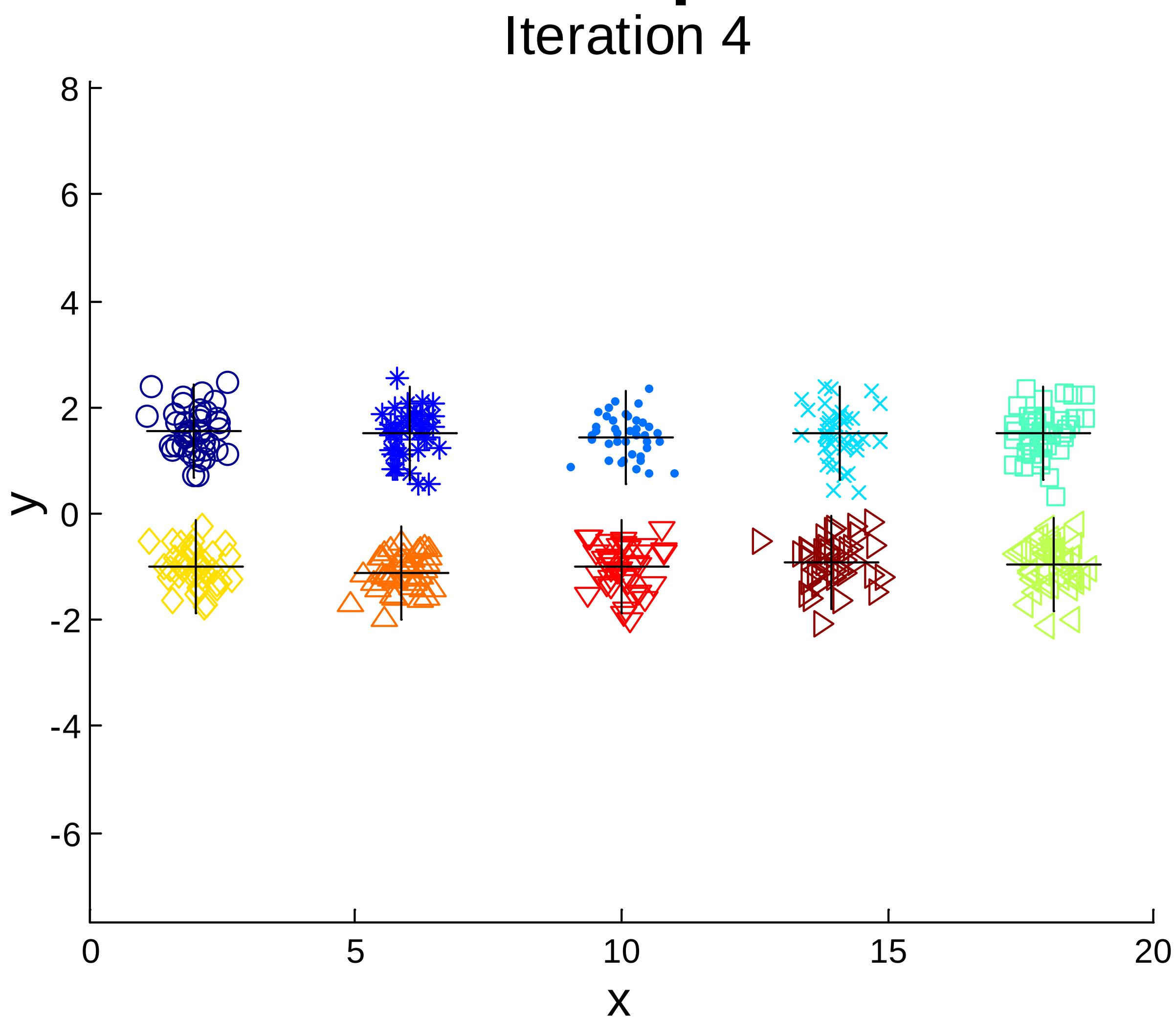
Problems with Selecting Initial Points

- 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

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

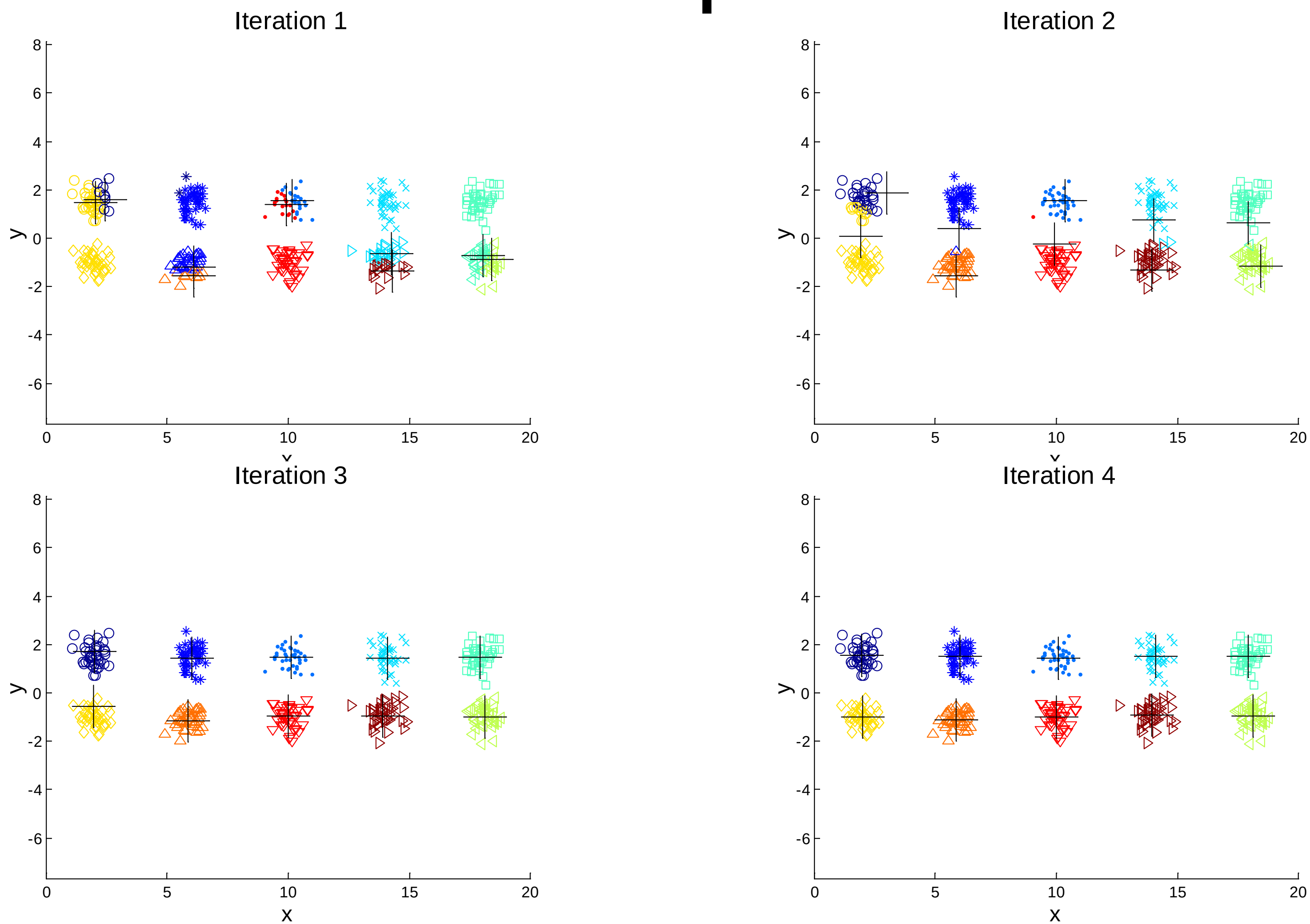
  - For example, if K = 10, then probability = $10!/10^{10}$ = 0.00036
  - Sometimes the initial centroids will readjust themselves in 'right' way, and sometimes they don't
  - Consider an example of five pairs of clusters

10 Clusters Example



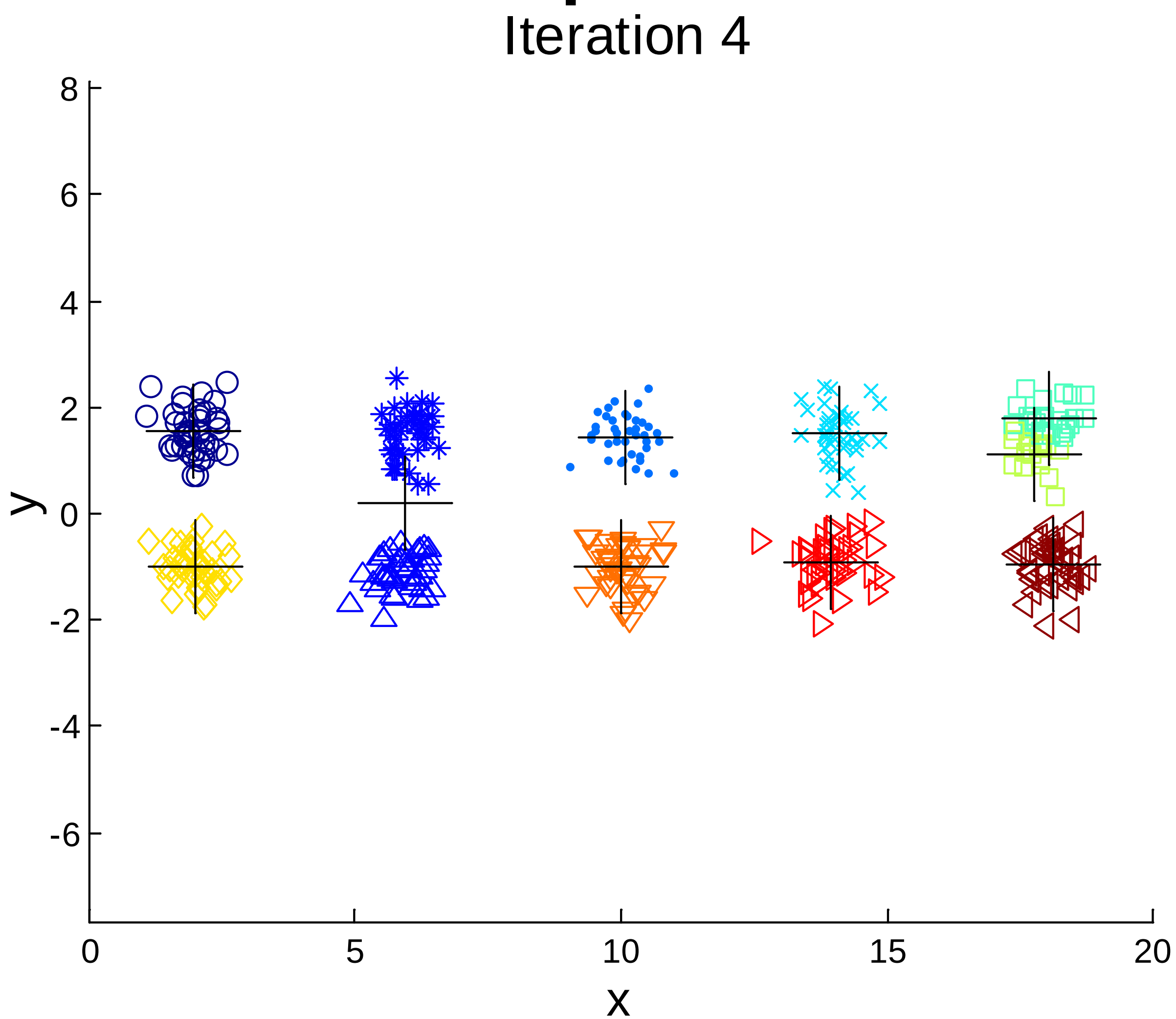
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example



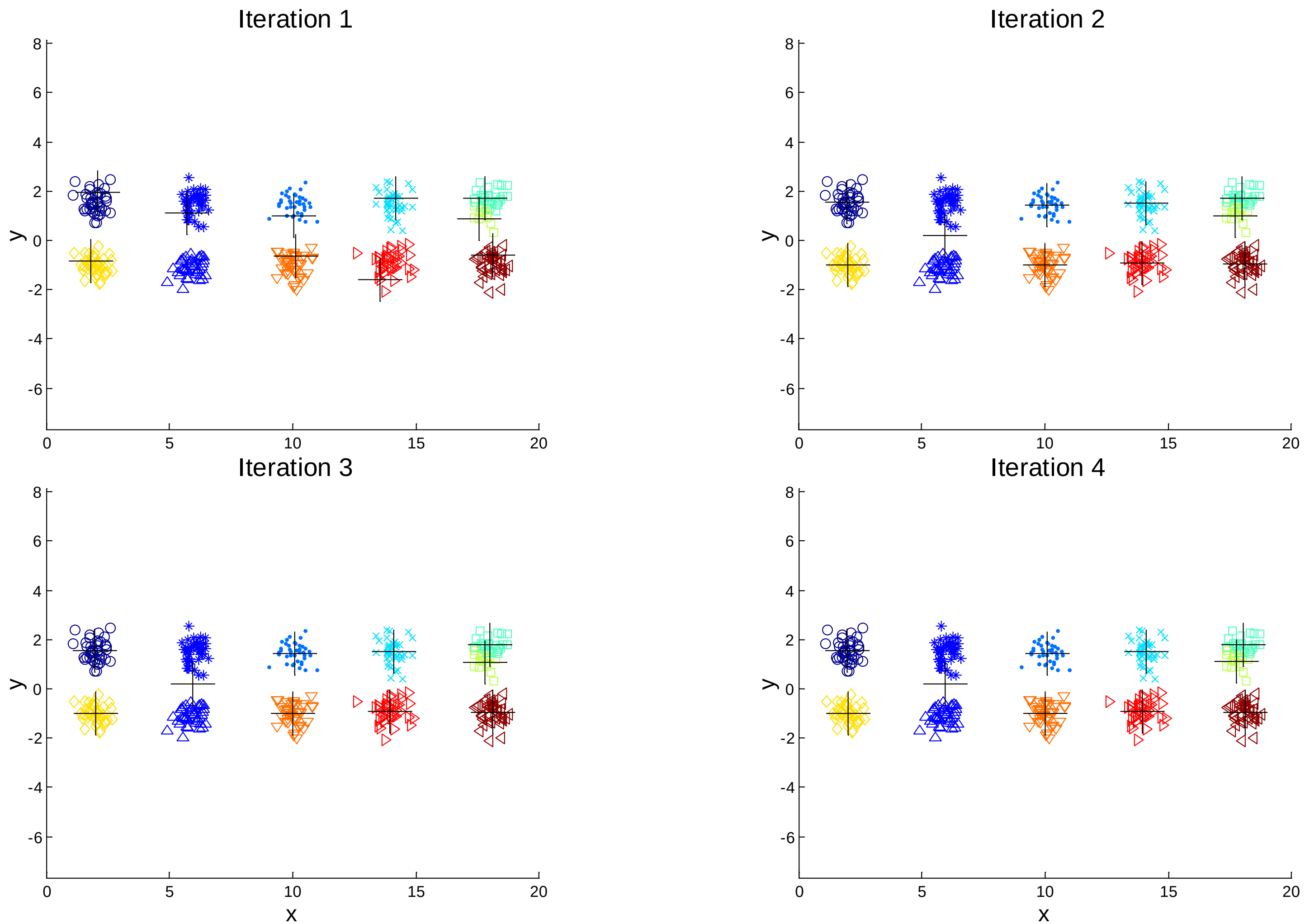
Starting with two initial centroids in one cluster of each pair of clusters

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

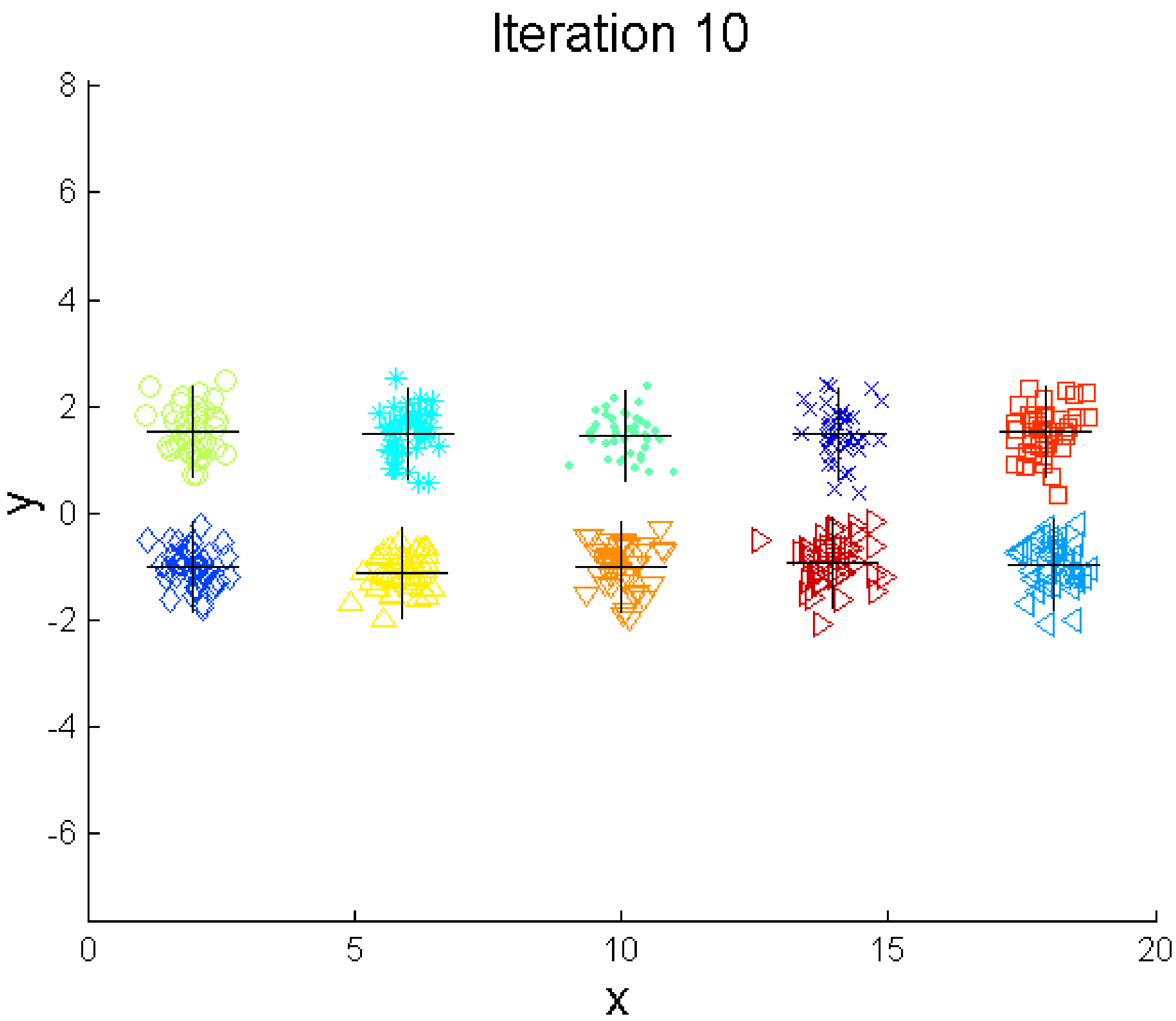
- Multiple runs
  - Helps, but probability is not on your side
- Sample and use hierarchical clustering to determine initial centroids
- Select more than k initial centroids and then select among these initial centroids
  - Select most widely separated
- Bisecting K-means
  - Not as susceptible to initialization issues

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

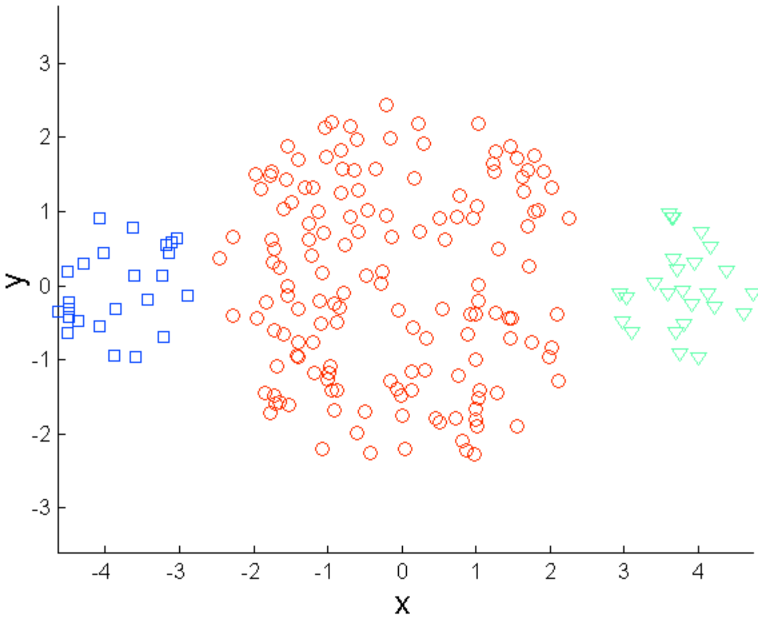


Limitations of K-means

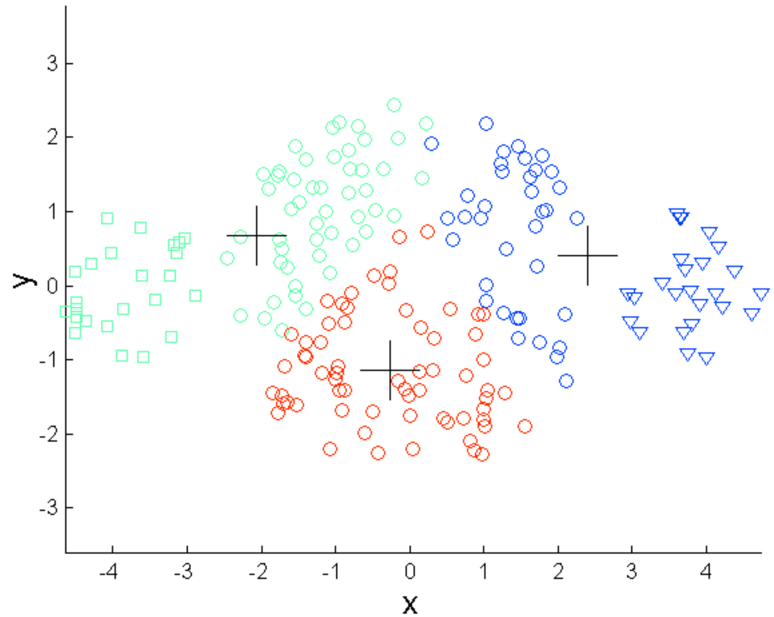
- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes

- K-means has problems when the data contains outliers.

Limitations of K-means: Differing Sizes

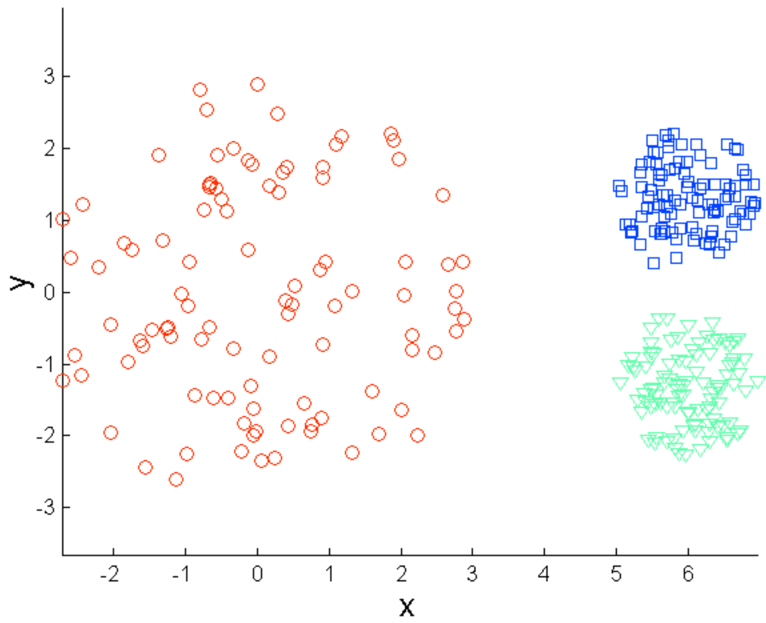


Original Points

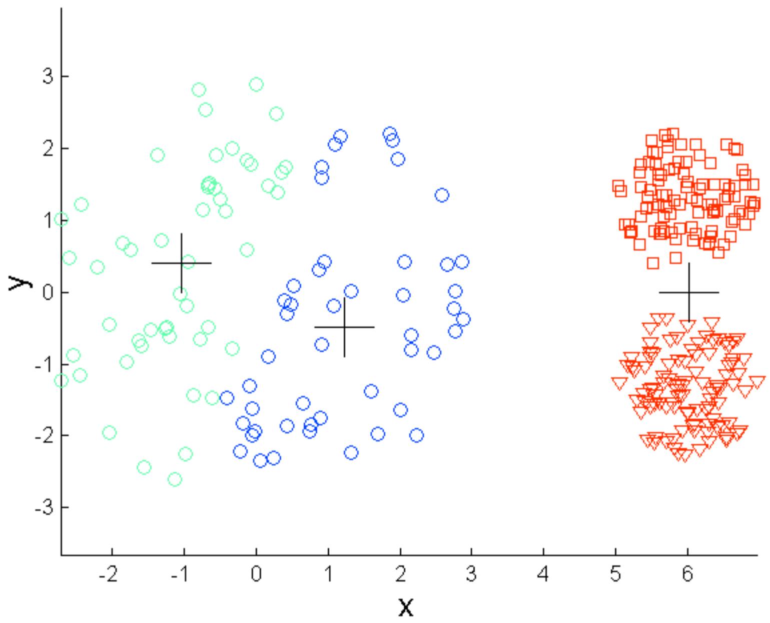


K-means (3 Clusters)

Limitations of K-means: Differing Density

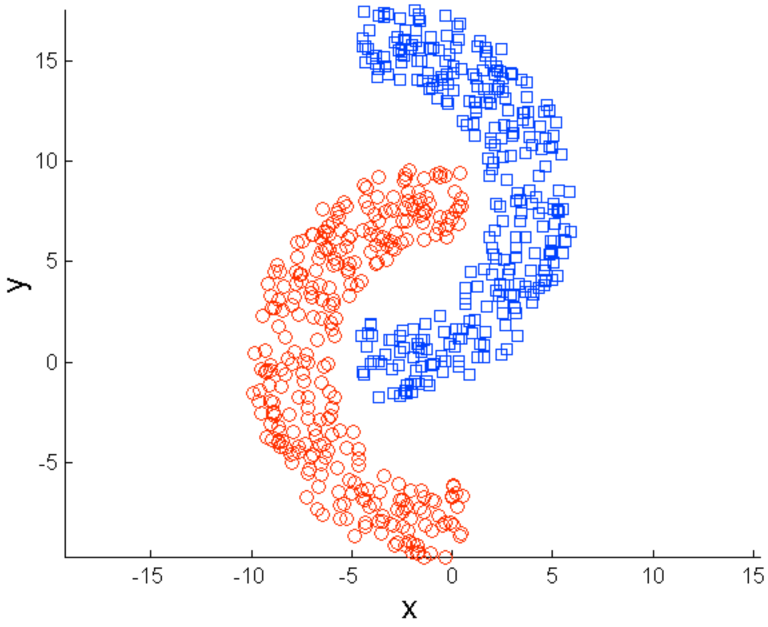


Original Points

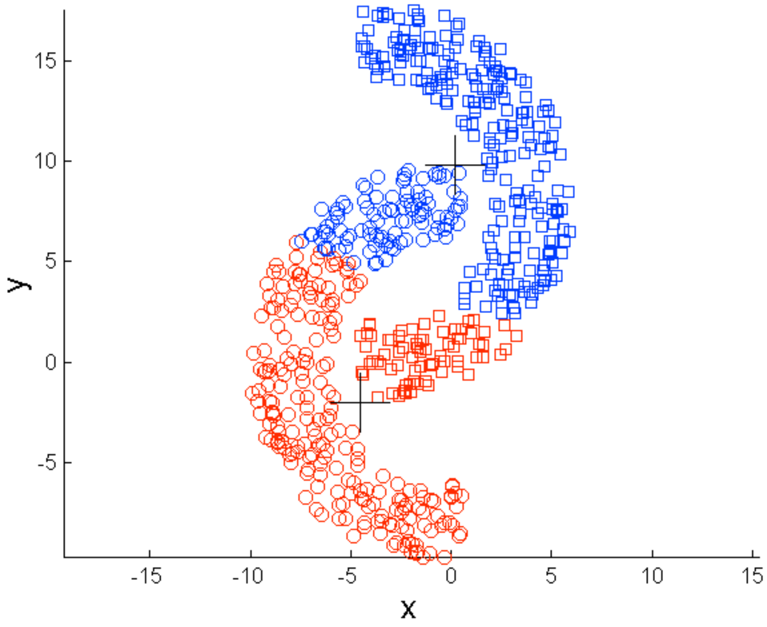


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes

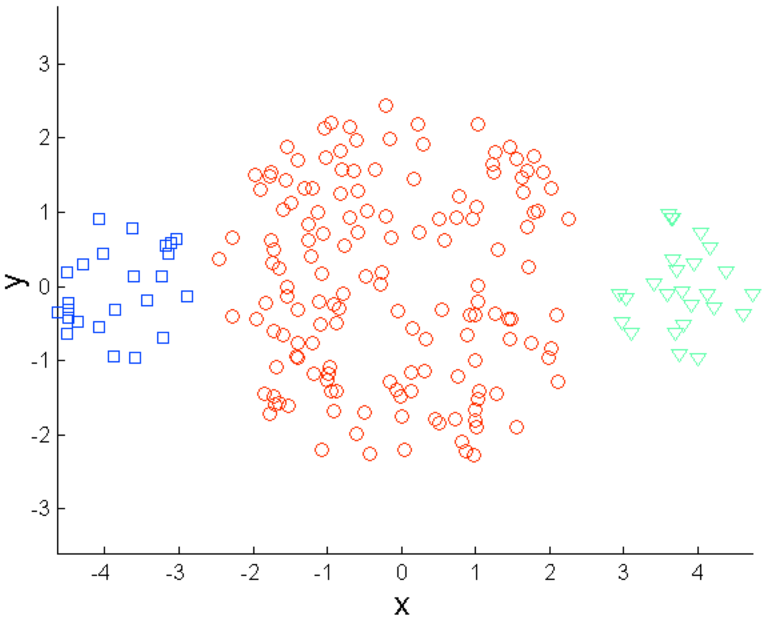


Original Points

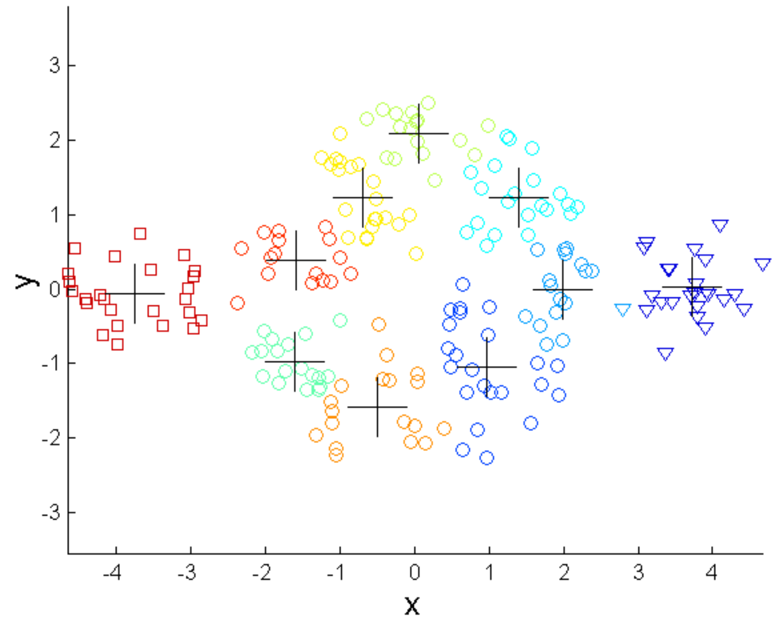


K-means (2 Clusters)

Overcoming K-means Limitations



Original Points

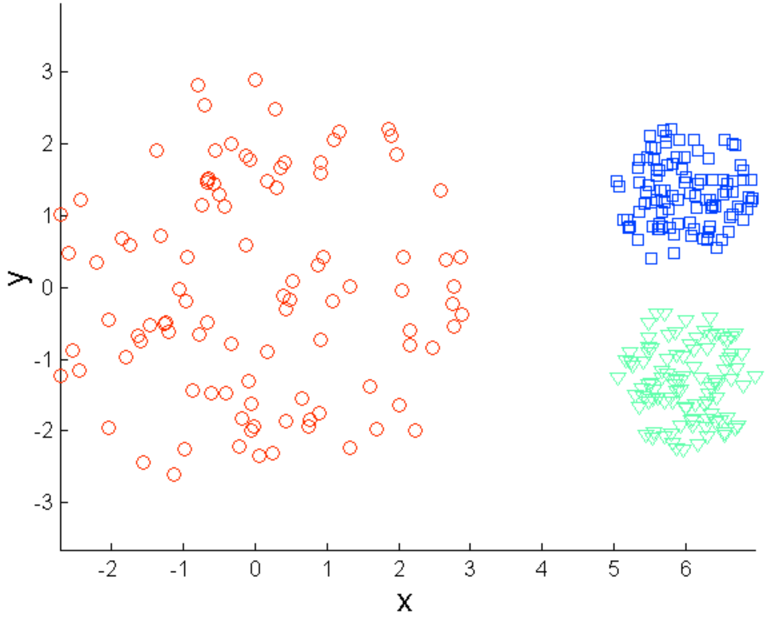


K-means Clusters

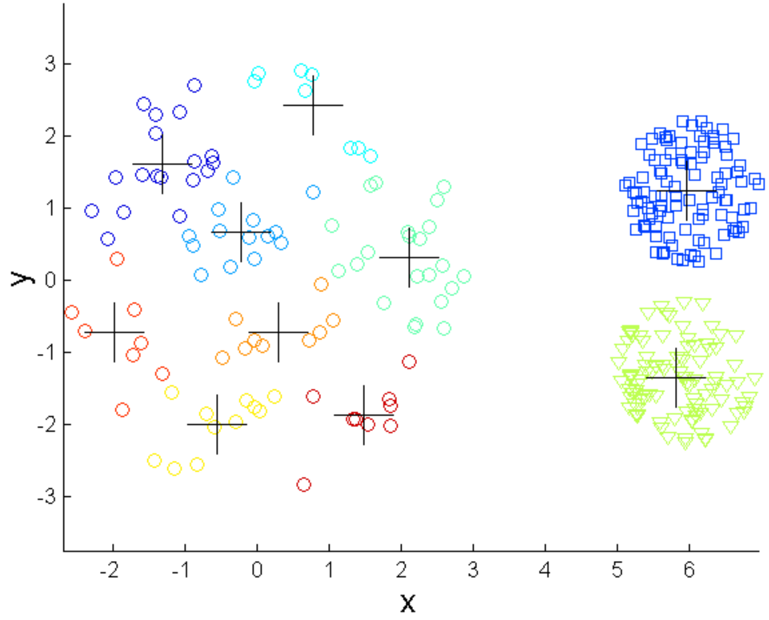
One solution is to use many clusters.

Find parts of clusters, but need to put together.

Overcoming K-means Limitations

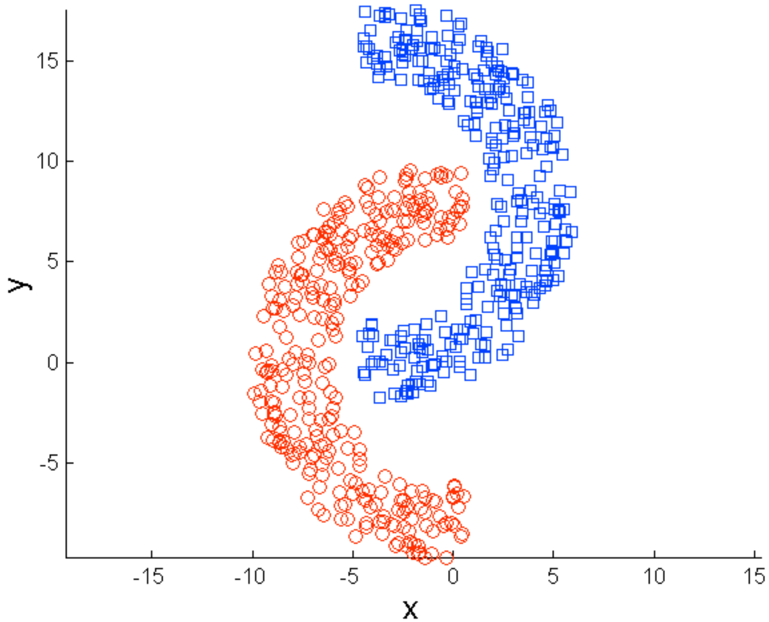


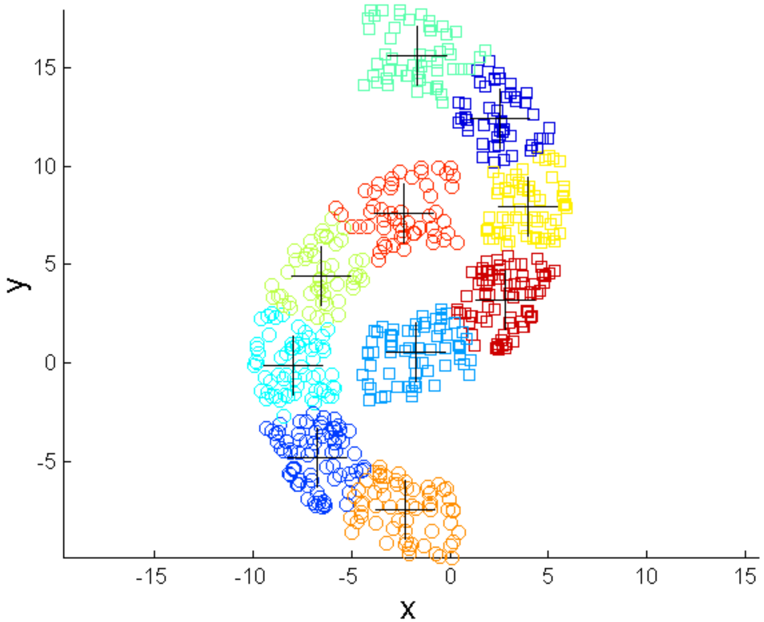
Original Points



K-means Clusters

Overcoming K-means Limitations

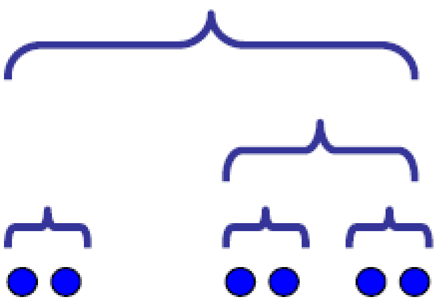




Original Points **K-means Clusters**

Hierarchical clustering

- Probably the most popular clustering algorithm in this area
- First presented in this context by Eisen in 1998



dendrogram

- Agglomerative (bottom-up)
- Algorithm:
  1. Initialize: each item a cluster
  2. Iterate:
     - select two most *similar* clusters
     - merge them
  3. Halt: when there is only one cluster left

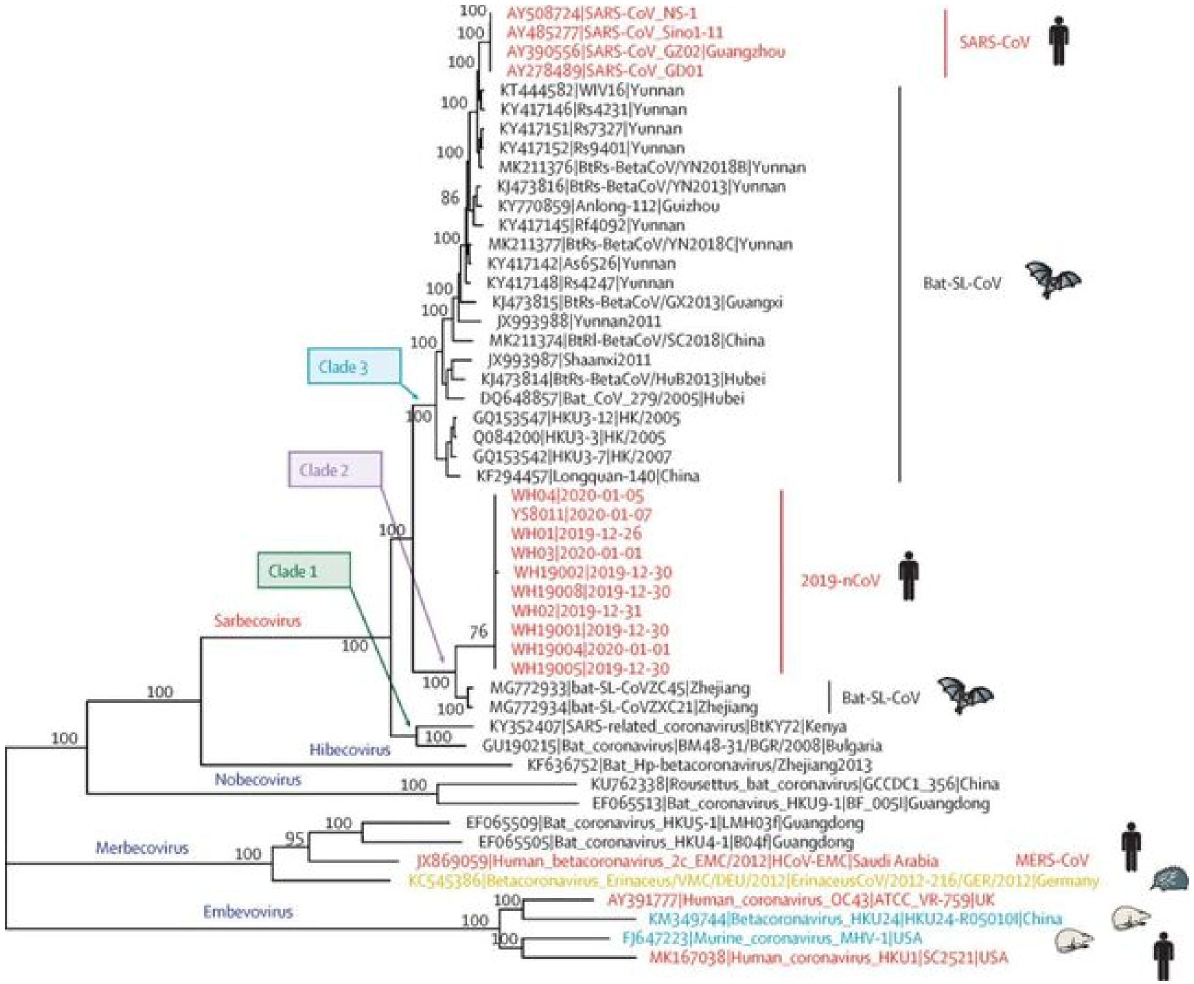
Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by 'cutting' the dendogram at the proper level

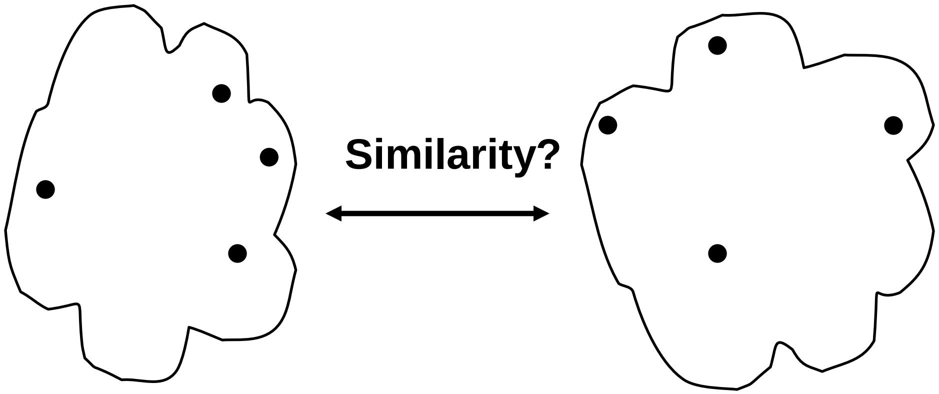
- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)

Example

Phylogenetic analysis of the complete genomes of 2019-nCoV and of the representative Betacoronavirus viruses https://spainsnews.com/ten-facts-against-coronavirus-alarmism/



How to Define Inter-Cluster Similarity

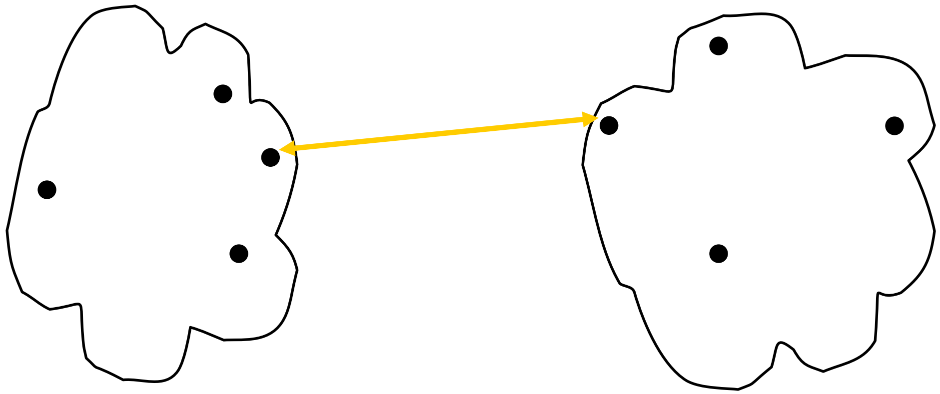


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Distance Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

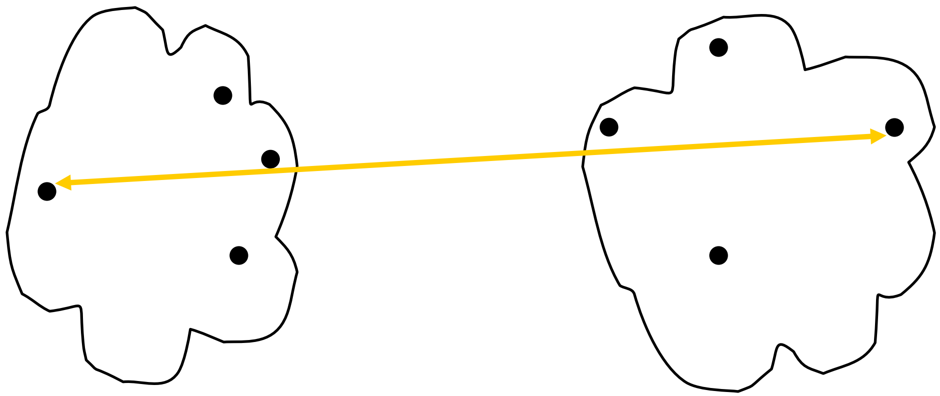


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Distance Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

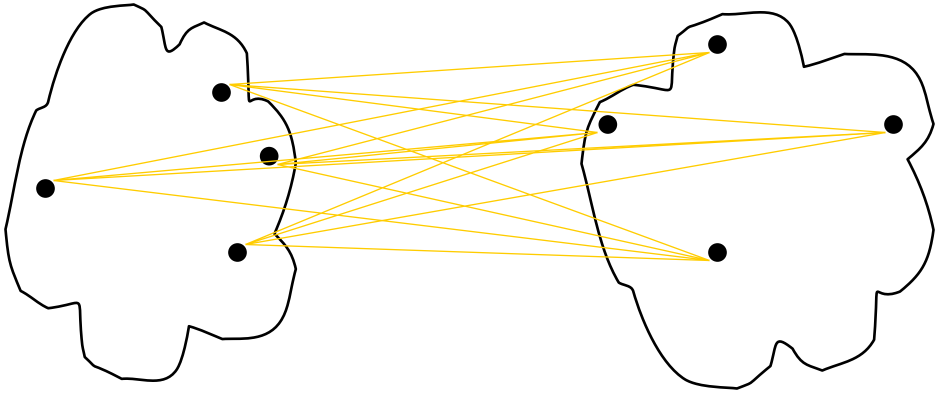


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Distance Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

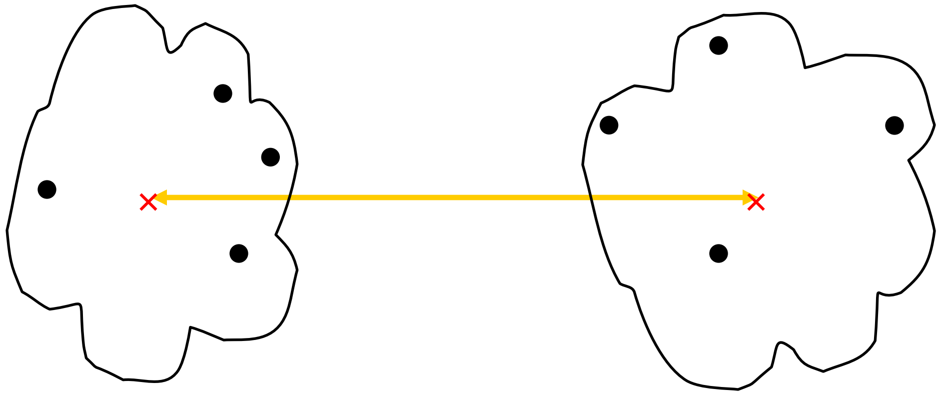


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Distance Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Distance Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

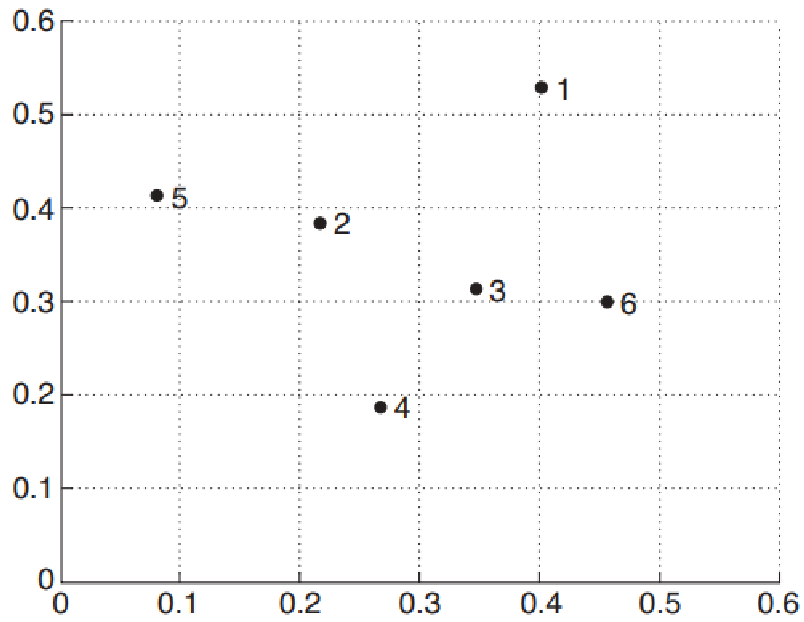
Cluster Similarity: MIN or Single Link

- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.

$$proximity(C_i, C_j) = \min_{\mathbf{x} \in C_i, \mathbf{y} \in C_j} proximity(\mathbf{x}, \mathbf{y})$$

Hierarchical Clustering: MIN

	X	Y
P1	0.4005	0.5306
P2	0.2148	0.3854
P3	0.3457	0.3156
P4	0.2652	0.1875
P5	0.0789	0.4139
P6	0.4548	0.3022



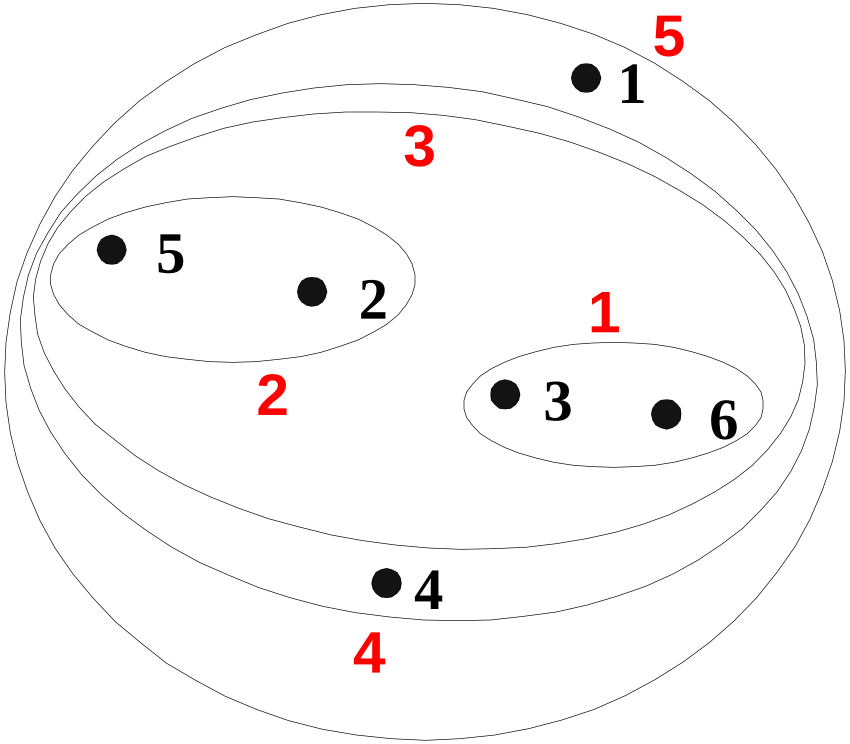
Hierarchical Clustering: MIN

Euclidian Distance Matrix

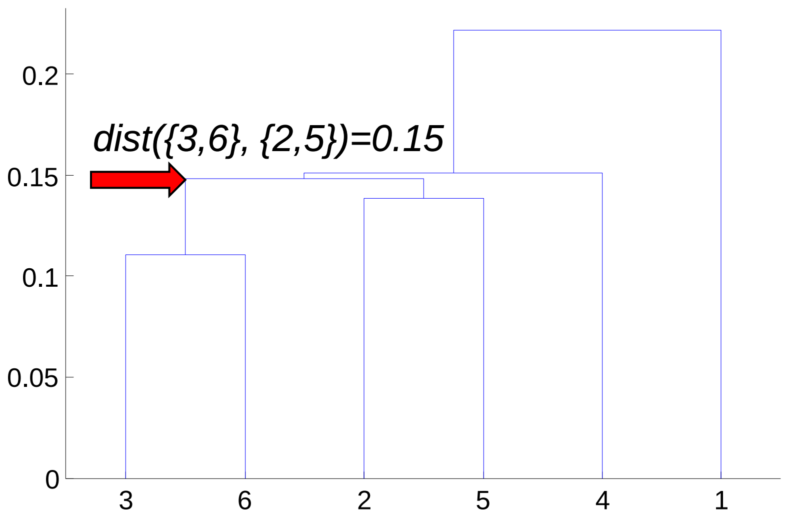
	P1	P2	P3	P4	P5	P6
P1	0	0.2357	0.2218	0.3688	0.3421	0.2347
P2	0.2357	0	0.1483	0.2042	0.1388	0.2540
P3	0.2218	0.1483	0	0.1513	0.2843	0.1100
P4	0.3688	0.2042	0.1513	0	0.2932	0.2216
P5	0.3421	0.1388	0.2843	0.2932	0	0.3921
P6	0.2347	0.2540	0.1100	0.2216	0.3921	0

$$
\begin{aligned}
dist(\{3,6\},\{2,5\}) &= \min(dist(3,2), dist(6,2), dist(3,5), dist(6,5)) \\
&= \min(0.15, 0.25, 0.28, 0.39) \\
&= 0.15.
\end{aligned}
$$

Hierarchical Clustering: MIN

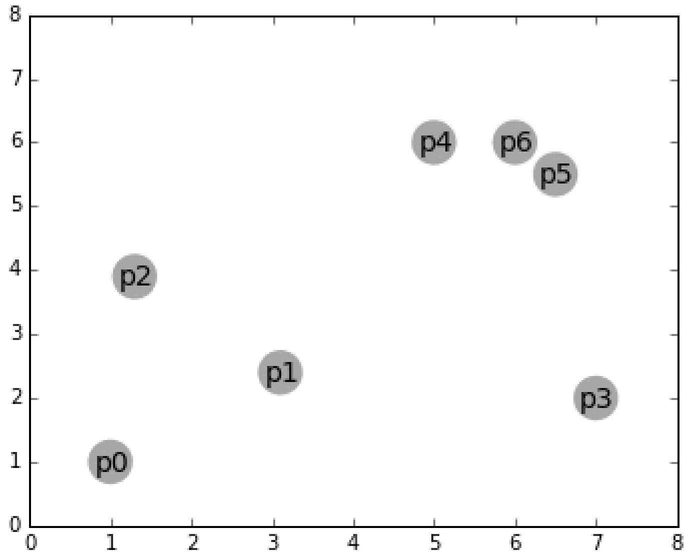


Nested Clusters

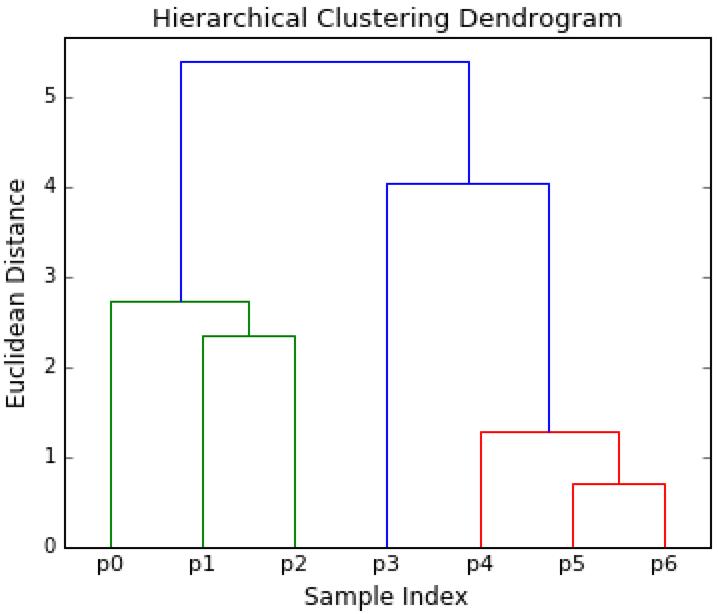


Dendrogram

Example: MIN



p0
p1
p2
p3
p4
p5
p6



Hierarchical Clustering Dendrogram
Euclidean Distance
Sample Index
p0
p1
p2
p3
p4
p5
p6

A Complete Example: MIN

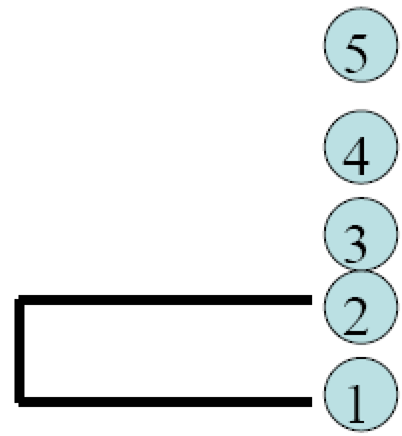
$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & & & & \\ 2 & 2 & 0 & & & \\ 3 & 6 & 3 & 0 & & \\ 4 & 10 & 9 & 7 & 0 & \\ 5 & 9 & 8 & 5 & 4 & 0 \end{array} \Rightarrow \begin{array}{c|cccc} & (1,2) & 3 & 4 & 5 \\ \hline (1,2) & 0 & & & \\ 3 & \boxed{3} & 0 & & \\ 4 & 9 & 7 & 0 & \\ 5 & 8 & 5 & 4 & 0 \end{array}$$

The minimum distance indicates to merge the two clusters

$$d_{(1,2),3} = \min\{d_{1,3}, d_{2,3}\} = \min\{6,3\} = 3$$

$$d_{(1,2),4} = \min\{d_{1,4}, d_{2,4}\} = \min\{10,9\} = 9$$

$$d_{(1,2),5} = \min\{d_{1,5}, d_{2,5}\} = \min\{9,8\} = 8$$

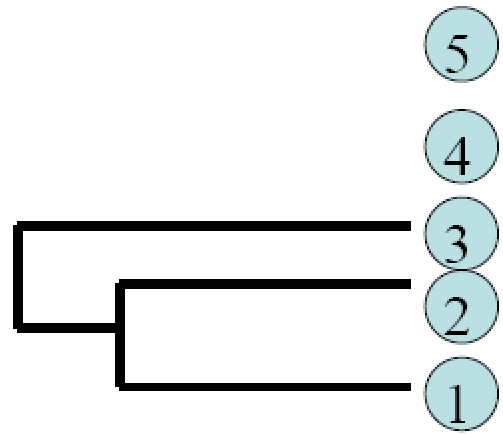


A Complete Example: MIN

$$
\begin{array}{c|ccccc}
 & 1 & 2 & 3 & 4 & 5 \\
\hline
1 & 0 & & & & \\
2 & 2 & 0 & & & \\
3 & 6 & 3 & 0 & & \\
4 & 10 & 9 & 7 & 0 & \\
5 & 9 & 8 & 5 & 4 & 0
\end{array}
\;\Rightarrow\;
\begin{array}{c|cccc}
 & (1,2) & 3 & 4 & 5 \\
\hline
(1,2) & 0 & & & \\
3 & 3 & 0 & & \\
4 & 9 & 7 & 0 & \\
5 & 8 & 5 & 4 & 0
\end{array}
\;\Rightarrow\;
\begin{array}{c|ccc}
 & (1,2,3) & 4 & 5 \\
\hline
(1,2,3) & 0 & & \\
4 & 7 & 0 & \\
5 & 5 & 4 & 0
\end{array}
$$

The minimum distance indicates to merge the two clusters

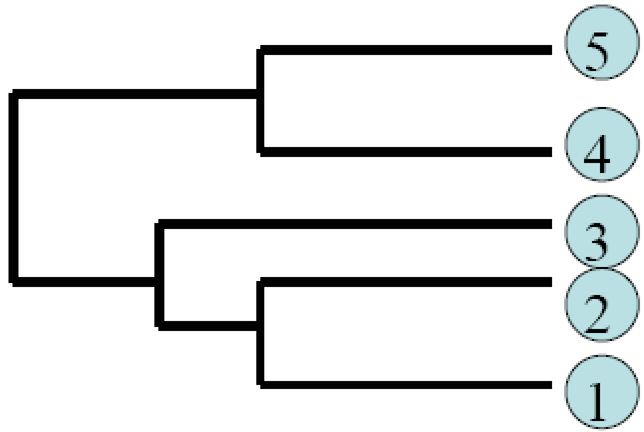
$$d_{(1,2,3),4} = \min\{d_{(1,2),4}, d_{3,4}\} = \min\{9,7\} = 7$$
$$d_{(1,2,3),5} = \min\{d_{(1,2),5}, d_{3,5}\} = \min\{8,5\} = 5$$



A Complete Example: MIN

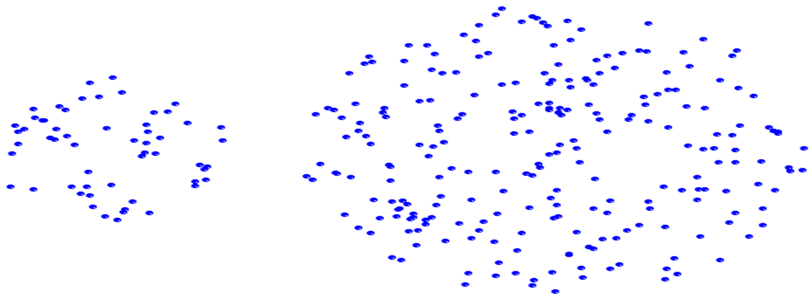
$$
\begin{array}{c|ccccc|}
 & 1 & 2 & 3 & 4 & 5 \\
1 & 0 & & & & \\
2 & 2 & 0 & & & \\
3 & 6 & 3 & 0 & & \\
4 & 10 & 9 & 7 & 0 & \\
5 & 9 & 8 & 5 & 4 & 0
\end{array}
\Rightarrow
\begin{array}{c|cccc|}
 & (1,2) & 3 & 4 & 5 \\
(1,2) & 0 & & & \\
3 & 3 & 0 & & \\
4 & 9 & 7 & 0 & \\
5 & 8 & 5 & 4 & 0
\end{array}
\Rightarrow
\begin{array}{c|ccc|}
 & (1,2,3) & 4 & 5 \\
(1,2,3) & 0 & & \\
4 & 7 & 0 & \\
5 & 5 & 4 & 0
\end{array}
$$

$$d_{(1,2,3),(4,5)} = \min\{d_{(1,2,3),4}, d_{(1,2,3),5}\} = 5$$

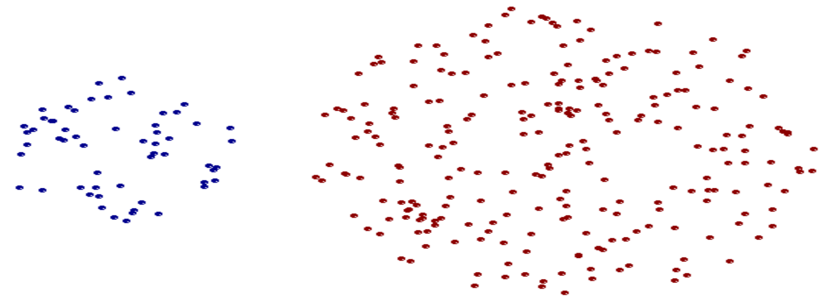


Strength of MIN

- **Can handle non-elliptical shapes**



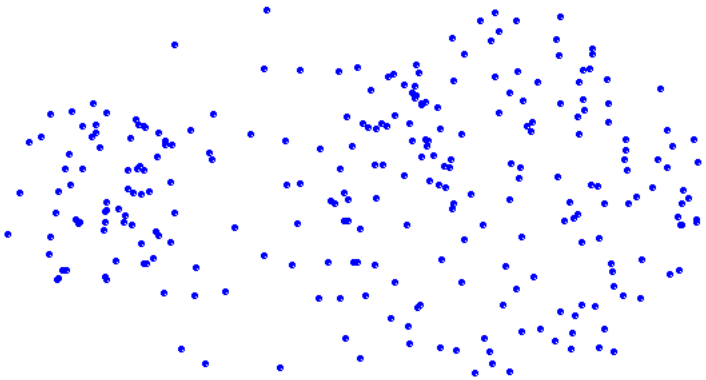
Original Points



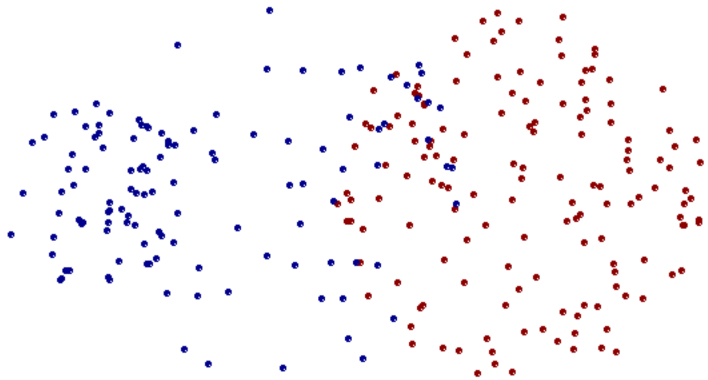
Two Clusters

The min distance between islands is short, so all of the Florida keys are connected by bridges and merged to state of Florida

Limitations of MIN



Original Points



Two Clusters

- **Sensitive to noise and outliers that often shorten the min distance**

Cluster Similarity: MAX or Complete Linkage

- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters
  - Determined by all pairs of points in the two clusters

$$proximity\,(C_i, C_j) = \max_{\mathbf{x}\in C_i, \mathbf{y}\in C_j} proximity\,(\mathbf{x}, \mathbf{y})$$

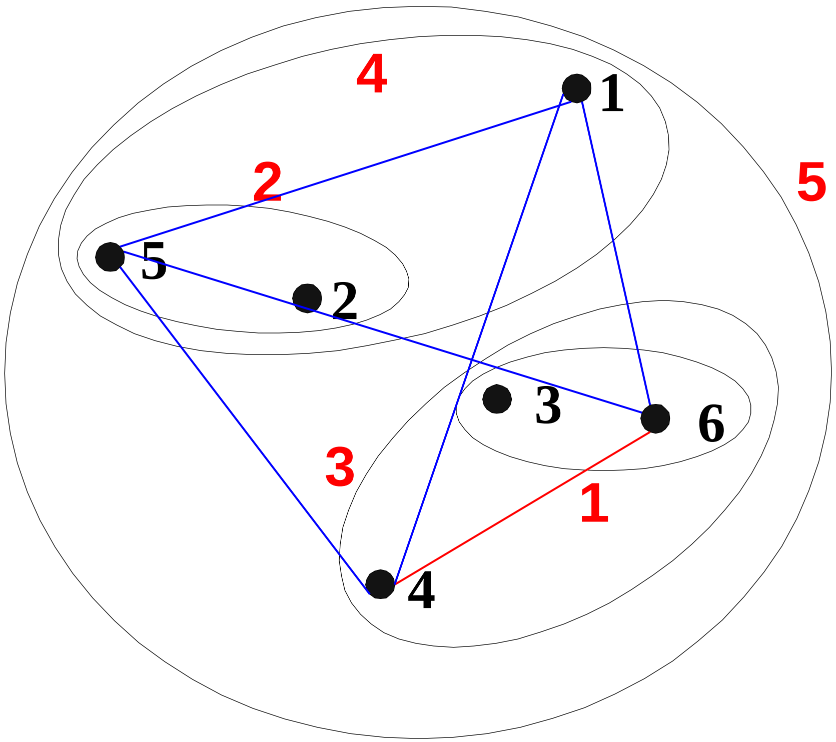
Hierarchical Clustering: MAX

Euclidian Distance Matrix

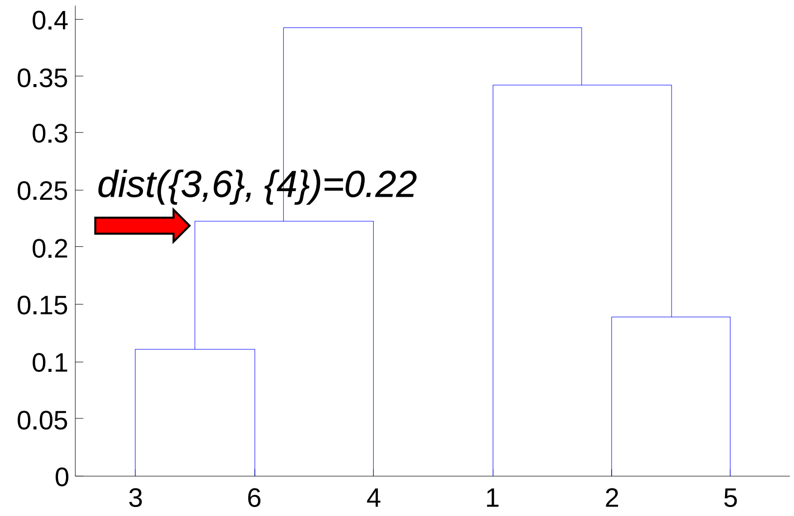
	P1	P2	P3	P4	P5	P6
P1	0	0.2357	0.2218	0.3688	0.3421	0.2347
P2	0.2357	0	0.1483	0.2042	0.1388	0.2540
P3	0.2218	0.1483	0	0.1513	0.2843	0.1100
P4	0.3688	0.2042	0.1513	0	0.2932	0.2216
P5	0.3421	0.1388	0.2843	0.2932	0	0.3921
P6	0.2347	0.2540	0.1100	0.2216	0.3921	0

$$
\begin{aligned}
dist(\{3,6\},\{4\}) &= \max(dist(3,4), dist(6,4)) \\
&= \max(0.15, 0.22) \\
&= 0.22. \\
dist(\{3,6\},\{2,5\}) &= \max(dist(3,2), dist(6,2), dist(3,5), dist(6,5)) \\
&= \max(0.15, 0.25, 0.28, 0.39) \\
&= 0.39. \\
dist(\{3,6\},\{1\}) &= \max(dist(3,1), dist(6,1)) \\
&= \max(0.22, 0.23) \\
&= 0.23.
\end{aligned}
$$

Hierarchical Clustering: MAX

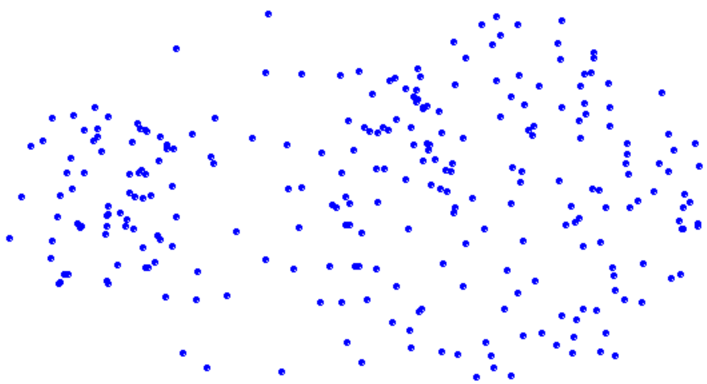


Nested Clusters

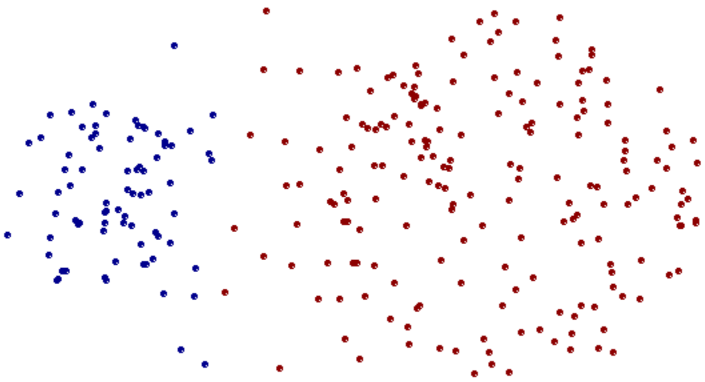


Dendrogram

Strength of MAX



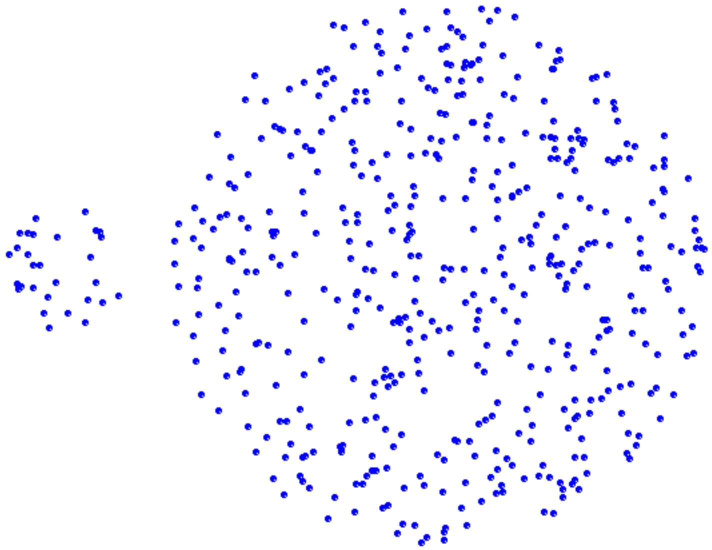
Original Points

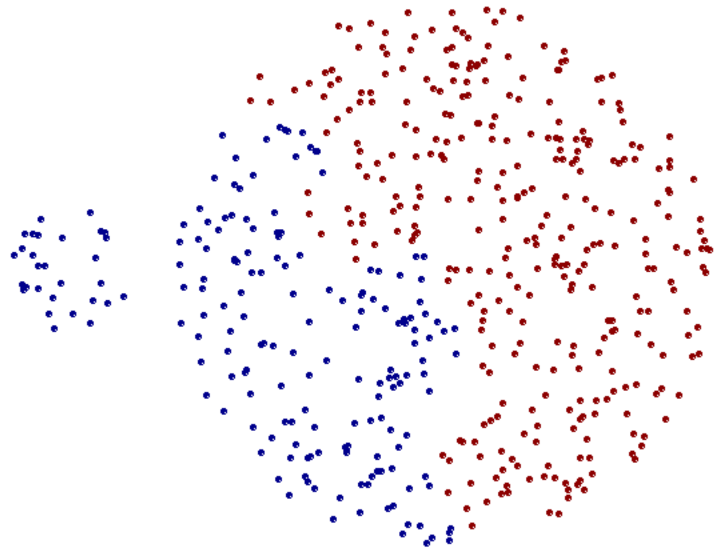


Two Clusters

- **Less susceptible to noise and outliers that often extend the max distance**

Limitations of MAX





Original Points **Two Clusters**

- **Tends to break large clusters**
- **Biased towards globular clusters**

Cluster Similarity: Group Average

- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

  – Need to use average connectivity for scalability since total proximity favors large clusters

$$proximity(C_i, C_j) = \frac{\sum_{\substack{\mathbf{x} \in C_i \\ \mathbf{y} \in C_j}} proximity(\mathbf{x}, \mathbf{y})}{m_i * m_j}$$

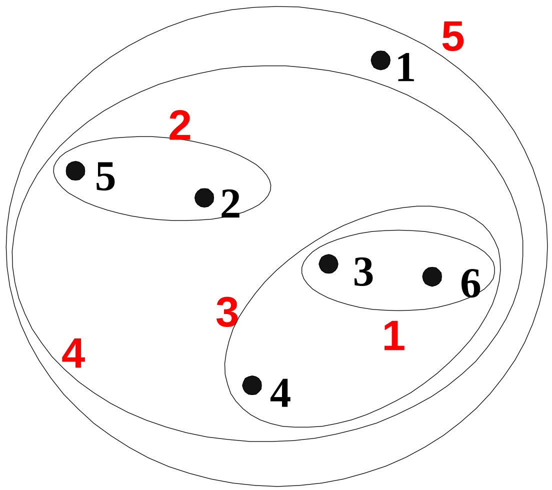
Hierarchical Clustering: Group Average

Euclidian Distance Matrix

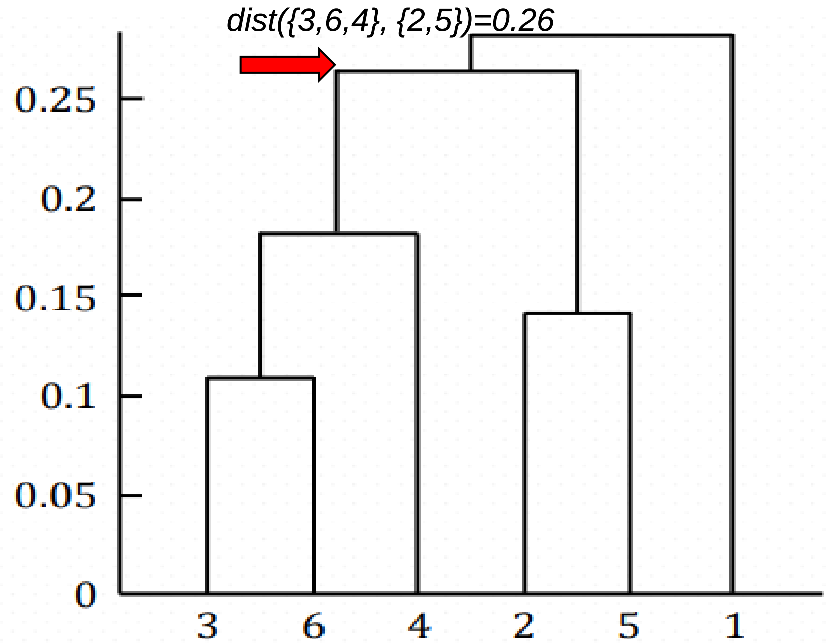
	P1	P2	P3	P4	P5	P6
P1	0	0.2357	0.2218	0.3688	0.3421	0.2347
P2	0.2357	0	0.1483	0.2042	0.1388	0.2540
P3	0.2218	0.1483	0	0.1513	0.2843	0.1100
P4	0.3688	0.2042	0.1513	0	0.2932	0.2216
P5	0.3421	0.1388	0.2843	0.2932	0	0.3921
P6	0.2347	0.2540	0.1100	0.2216	0.3921	0

$$
\begin{aligned}
dist(\{3,6,4\},\{1\}) &= (0.22 + 0.37 + 0.23)/(3 * 1) \\
&= 0.28 \\
dist(\{2,5\},\{1\}) &= (0.2357 + 0.3421)/(2 * 1) \\
&= 0.2889 \\
dist(\{3,6,4\},\{2,5\}) &= (0.15 + 0.28 + 0.25 + 0.39 + 0.20 + 0.29)/(6 * 2) \\
&= 0.26
\end{aligned}
$$

Hierarchical Clustering: Group Average



Nested Clusters

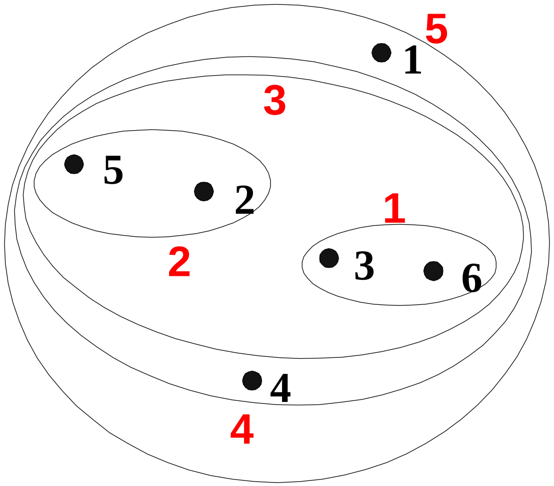


Dendrogram

Hierarchical Clustering: Group Average

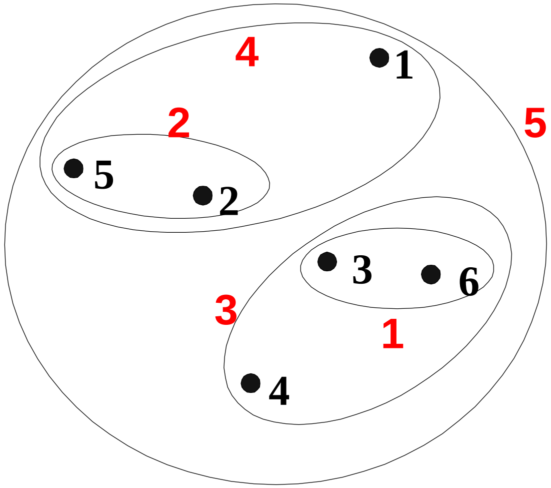
- Compromise between Single and Complete Link
- Strengths
  - Less susceptible to noise and outliers
- Limitations
  - Biased towards globular clusters

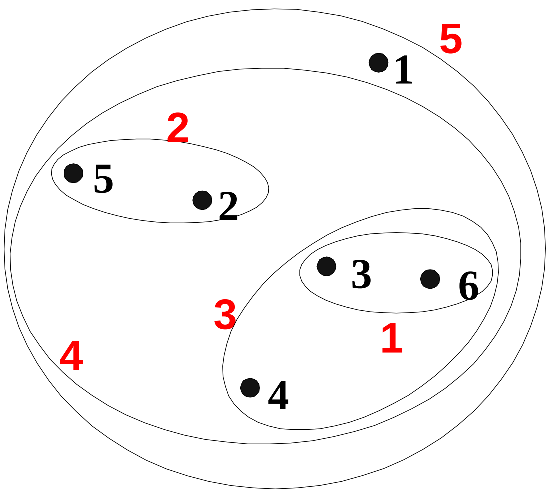
Hierarchical Clustering: Comparison



MIN

MAX





Group Average

Hierarchical Clustering: Problems and Limitations

- Once a decision is made to combine two clusters, it cannot be undone
- No objective function is directly minimized
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Difficulty handling different sized clusters and convex shapes
  - Breaking large clusters

DBSCAN: Density-Based Clustering

- DBSCAN is a Density-Based Clustering algorithm

- Reminder: In density based clustering we partition points into dense regions separated by not-so-dense regions.
  - Why is Philadelphia not part of big NYC area?

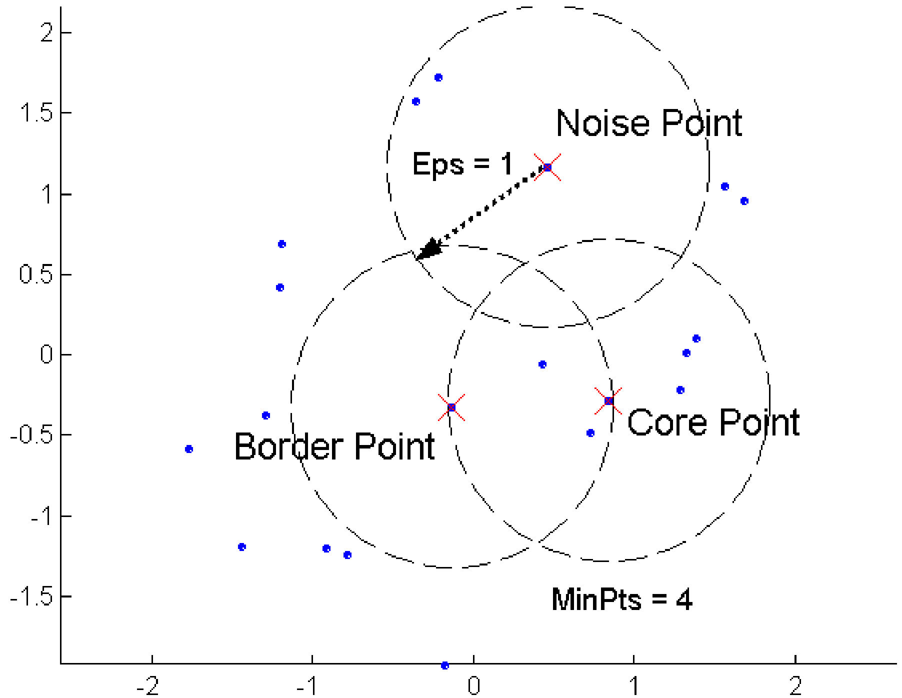
- Important Questions:
  - How do we measure density?
  - What is a dense region?

- DBSCAN:
  - Density at point p: number of points within a circle of radius Eps
  - Dense Region: A circle of radius Eps that contains at least MinPts points

DBSCAN

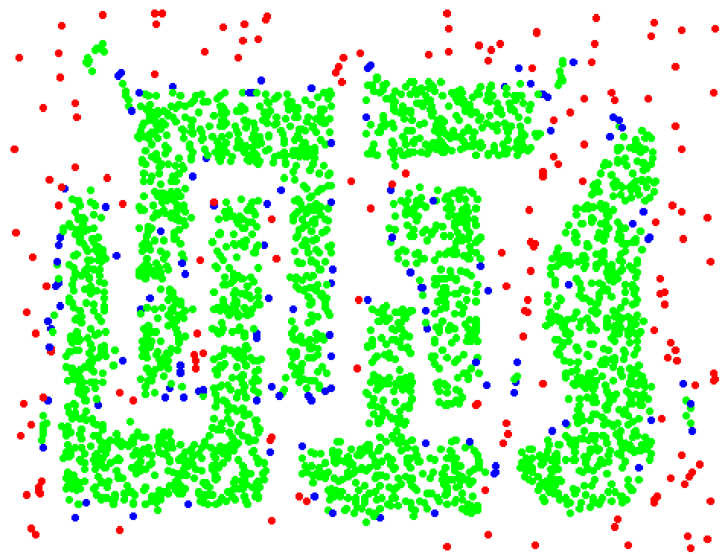
- Characterization of points
  - A point is a core point if it has more than a specified number of points (MinPts) within Eps
    - These points belong in a dense region and are at the interior of a cluster
  - A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point.
  - A noise point is any point that is not a core point or a border point.

DBSCAN: Core, Border, and Noise Points



DBSCAN: Core, Border and Noise Points

Original Points

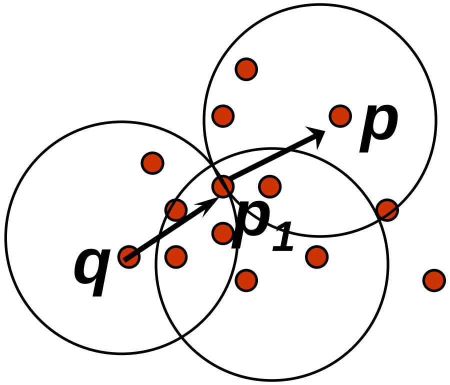


Point types: core, border and noise

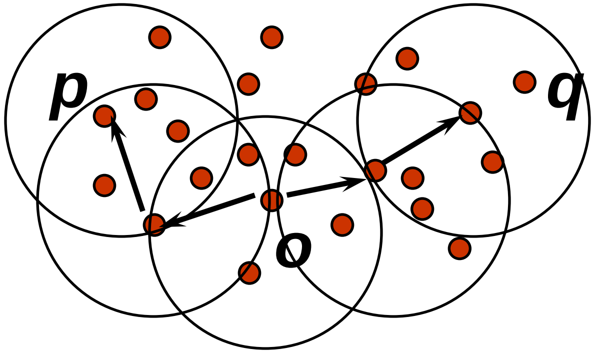
Eps = 10, MinPts = 4

DBSCAN: More Concepts

- Density-reachable:
  - A point *p* is density-reachable from a point *q* wrt. *Eps*, *MinPts* if there is a chain of points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i



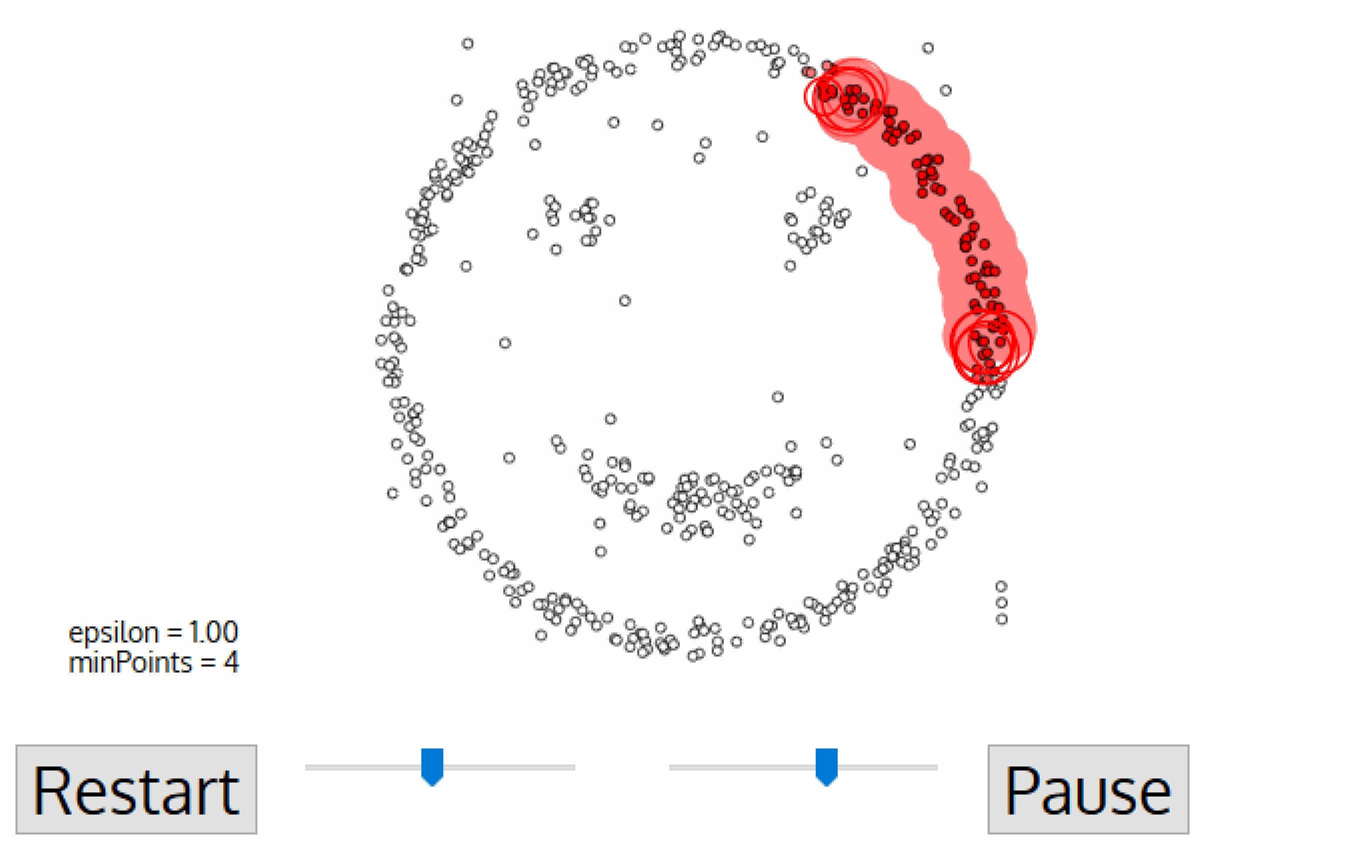
- Density-connected
  - A point *p* is density-connected to a point *q* wrt. *Eps*, *MinPts* if there is a point *o* such that both, *p* and *q* are density-reachable from *o* wrt. *Eps* and *MinPts*.



DBSCAN Algorithm

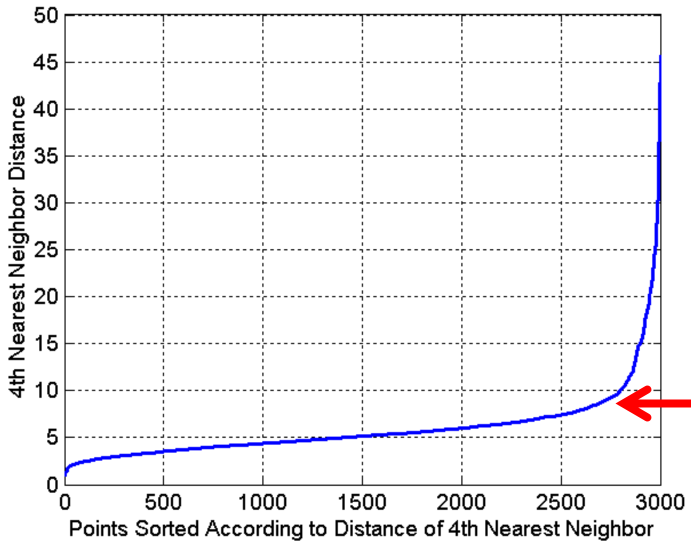
- Label points as core, border and noise
- Eliminate noise points
- For every core point p that has not been assigned to a cluster
  - Create a new cluster with the point p and all the points that are density-connected to p.
- Assign border points to the cluster of the closest core point.
- (very similar to boundary detection and color filling in computer graphics)

DBSCAN Algorithm



DBSCAN: Determining Eps and MinPts

- Idea is that for points in a cluster, their k^{th} nearest neighbors are at roughly the same distance
- Noise points have the k^{th} nearest neighbor at farther distance
- So, plot sorted distance of every point to its k^{th} nearest neighbor
- Find the distance d where there is a "knee" in the curve
  - Eps = d, MinPts = k
- Or, based on domain expert who knows the mechanism

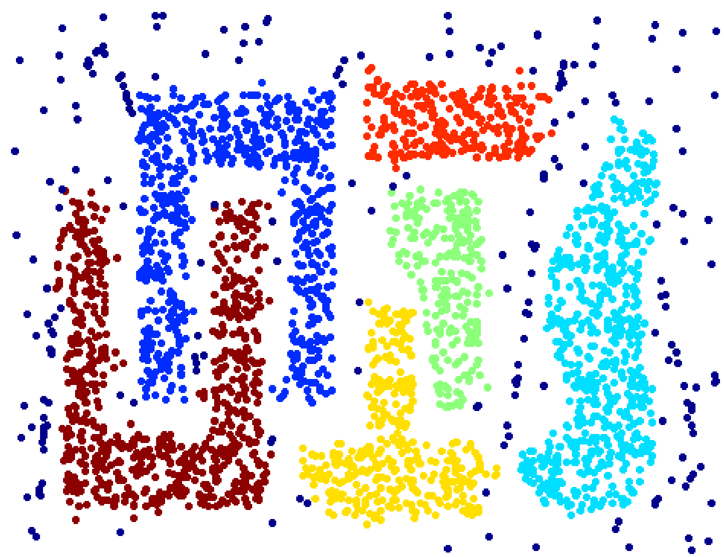


It is the radius crossing the border between dense region and sparse region

Eps ~ 7-10
MinPts = 4

When DBSCAN Works Well

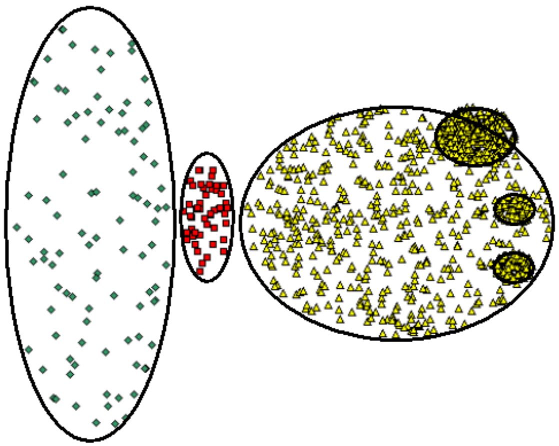
Original Points



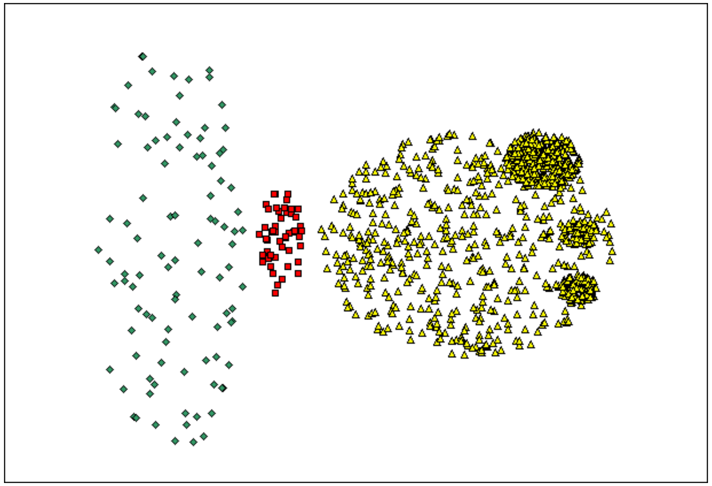
Clusters

- Resistant to Noise
- It groups points that are closely packed together, expanding clusters in any direction where there are nearby points, thus dealing with different shapes of clusters.
- Assume that the density within a cluster has a lower bound

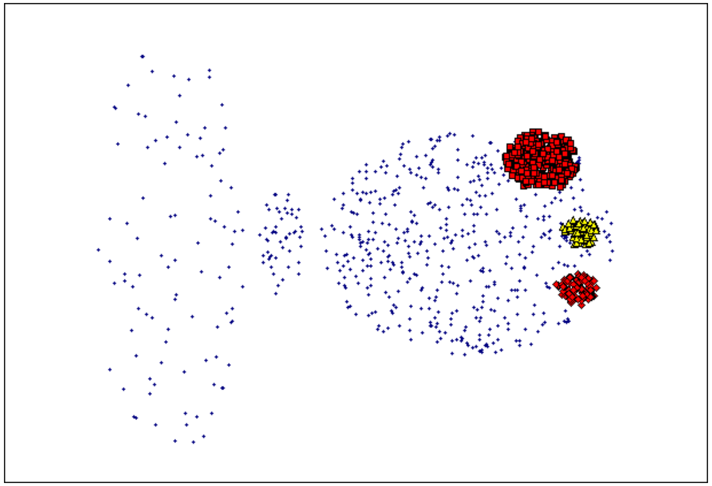
When DBSCAN Does NOT Work Well



Original Points



(MinPts=4, Eps=9.92).



(MinPts=4, Eps=9.75)

- Cannot handle varying densities
- Sensitive to parameters—hard to determine the correct set of parameters
- Dimensions may make a big difference

R packages/functions

- Hierarchical clustering
  - hclust
- Kmeans
  - Kmeans
- DBSCAN
  - dbscan

Acknowledgments

- Tan, Steinbach, Kumar: for some of the slides adapted or modified from their book *Introduction to Data Mining* slides
- Carlo Colantuoni: for some of the slides adapted or modified from his lecture slides at JHU
- Ziv Bar-Joseph: for some of the slides adapted or modified from his lecture slides at CMU