# Clustering (Unsupervised Learning)

David Li

# Outline

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

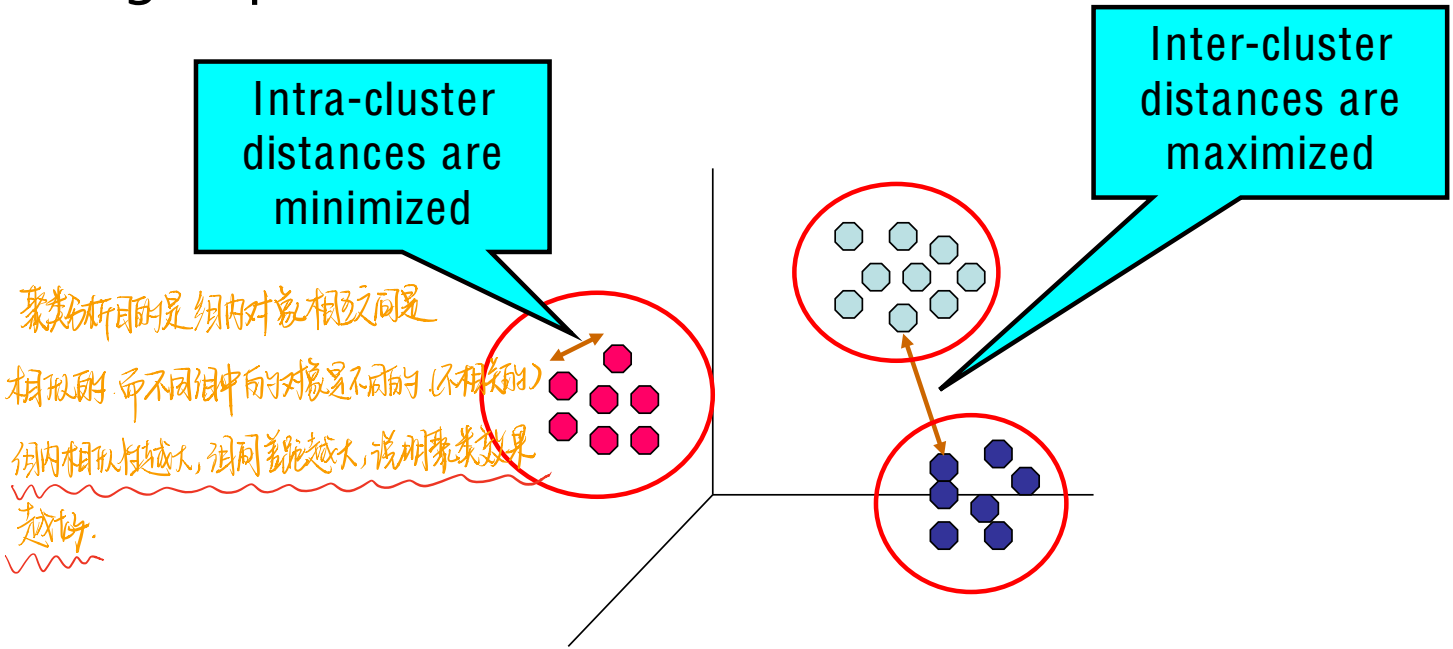
# Typical Data Analysis

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). 特征选择
- Identification of new/unknown sample classes using data (unsupervised learning – clustering) 聚类分析
- Classification of sample into known classes (supervised learning – classification) 分类分析

# 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

聚类效果好坏依赖两个因素 { 衡量距离的标准 / 聚类算法

# Clustering algorithms

The types of clustering methods:

- Hierarchical Clustering Methods 层次聚类
  - Agglomerative *hierarchical* clustering
  - Divisive clustering
- Model Based Clustering Methods
  - COBWEB, Gaussian mixtures
- Grid Based Clustering Methods 基于网格
  - STING. Wave Cluster and CLIQUE
- Density Based Clustering Methods: 基于密度
  - *DBSCAN*
- Partition Clustering Methods 划分聚类
  - *K-Means*, K-Mediods

# Distance

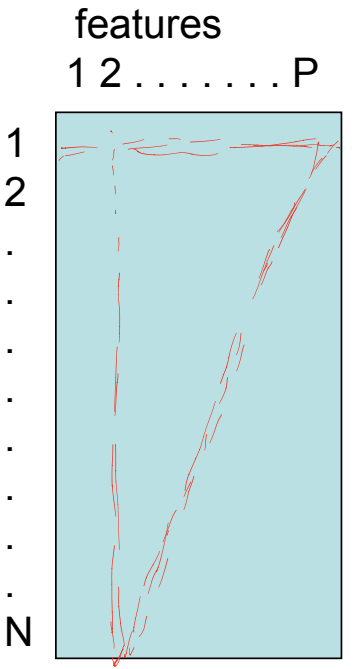
- We need a mathematical definition of distance between two points
  - Manhattan, Euclidian, Cosine, Correlation, etc
- What are points?
  - A sample' s all observed values of features
  - A student' s all quiz/exam grades
- What is the mathematical definition of a point?
  - The vector of features (X1, 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

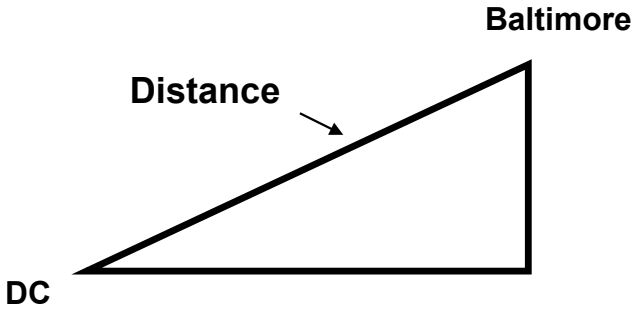
Manhattan 距离: $d(x,y) = |x_1 - y_1| + |x_2 - y_2| + \cdots |x_p - y_p|$.

# Most Famous Distance

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

$d(x,y) = \sqrt{|x_1-y_1|^2 + |x_2-y_2|^2 + \cdots + |x_p-y_p|^2}$

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



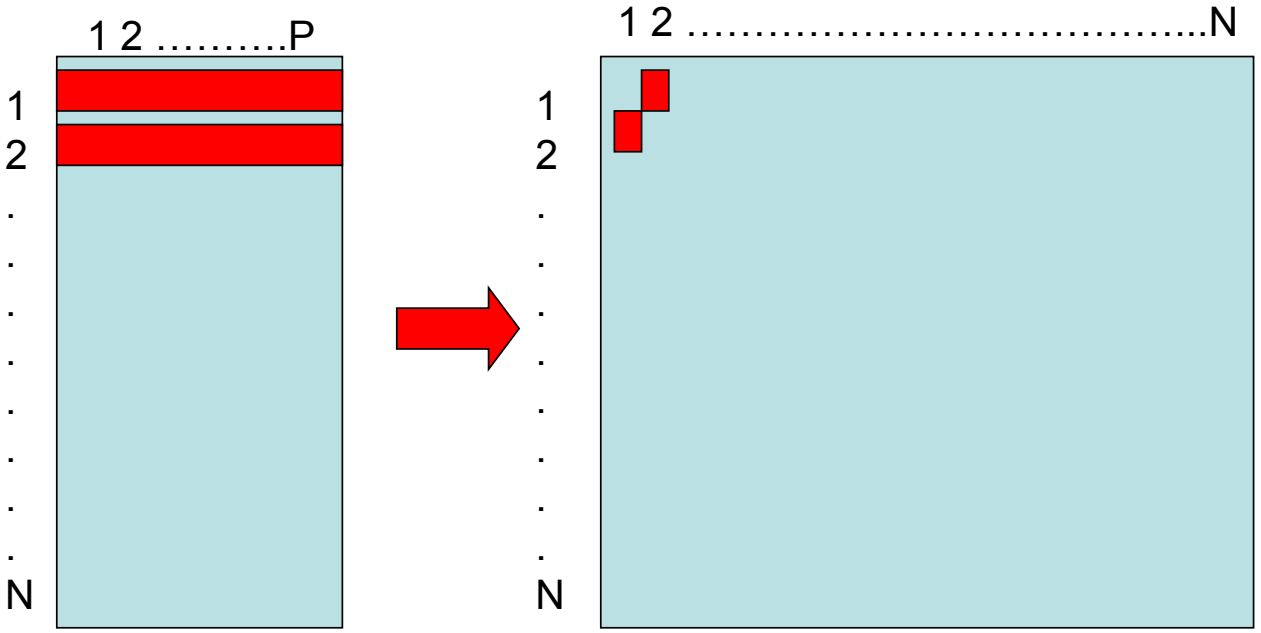

**When *P* is 20,000 you have to think abstractly**

# 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.

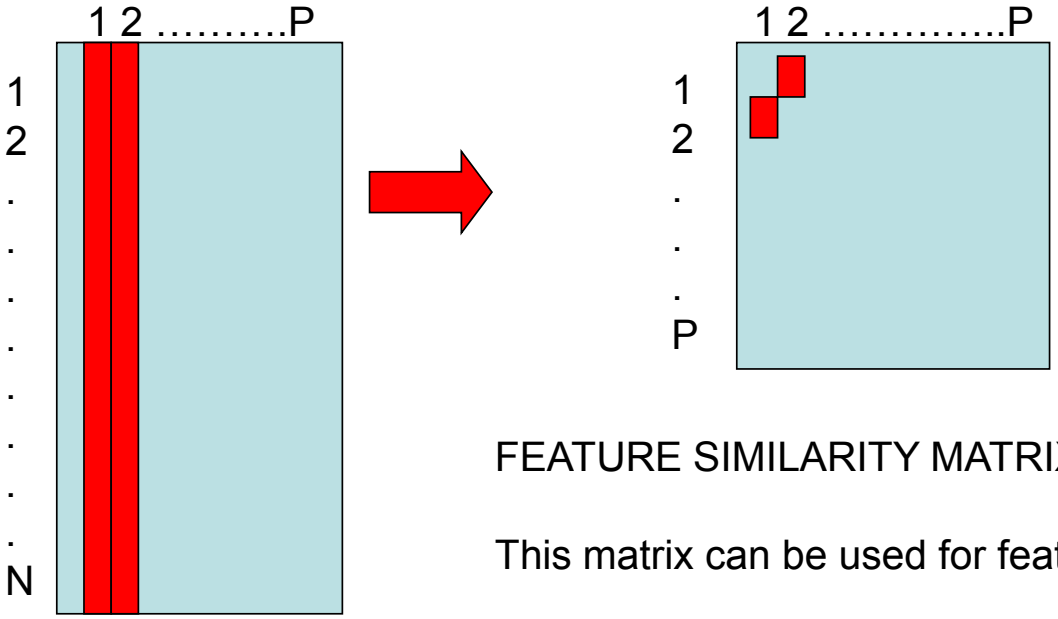
# The similarity/distance matrices




DATA MATRIX

SAMPLE SIMILARITY MATRIX

# The similarity/distance matrices




DATA MATRIX

FEATURE SIMILARITY MATRIX

This matrix can be used for feature selection

# 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 K-均值聚类
  - Partitioning method
  - Requires user to define K = # of clusters a priori
  - No picture to (over) interpret
- Hierarchical clustering 层次聚类
  - 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）根据密度的聚类
  - Identifying points that are in dense regions of the feature space, where many data points are close together.
- 根据网格的聚类.

# 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

---

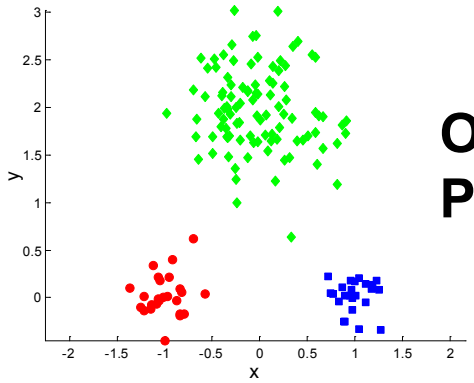
1: Select $K$ points as the initial centroids.
2: **repeat**
3: Form $K$ clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

---

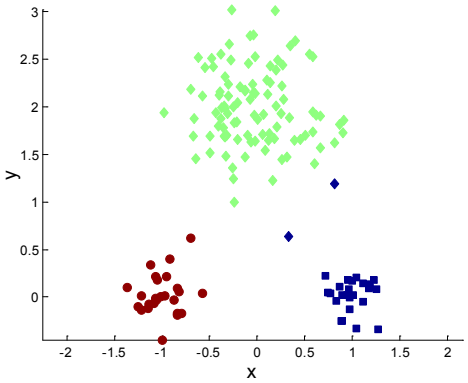
# K-means Clustering – Details

- 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'

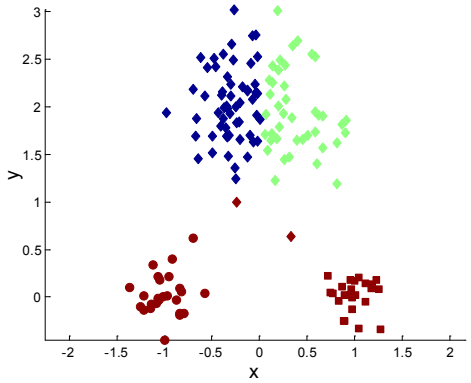
# Two different K-means Clustering




**Original Points**

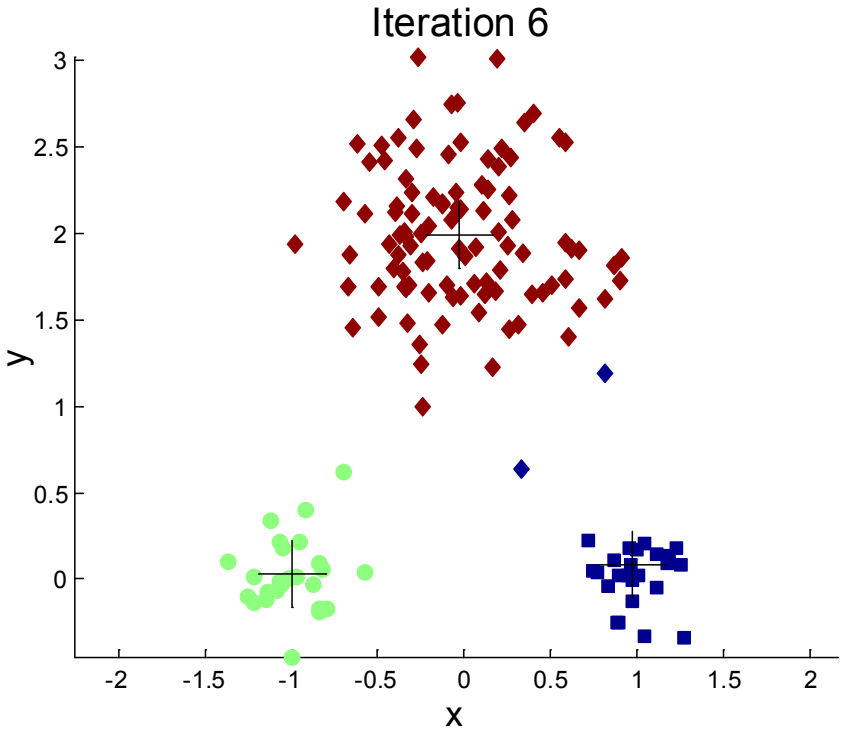



**Optimal Clustering**

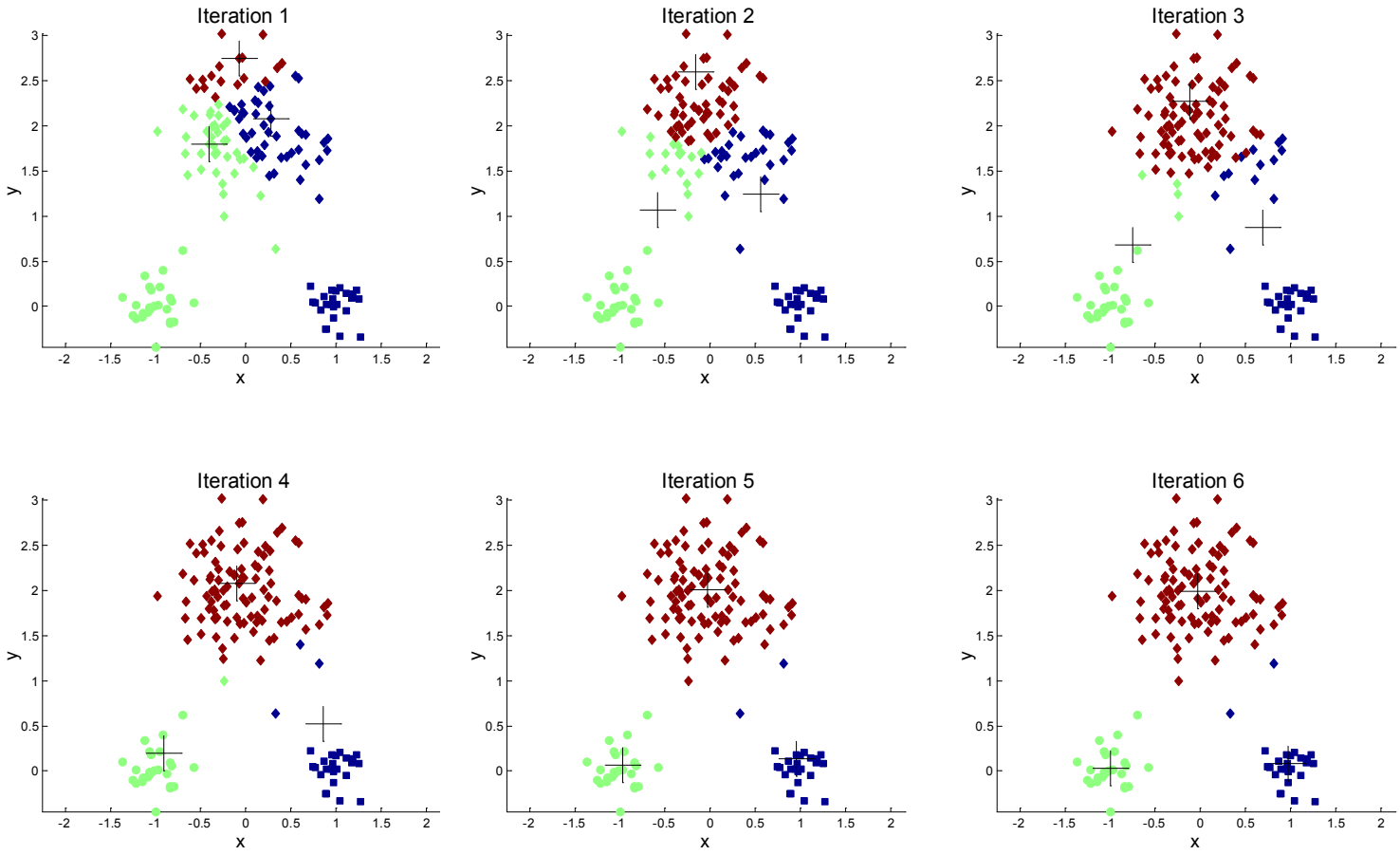



**Sub-optimal Clustering**

# Importance of Choosing Initial Centroids



# Importance of Choosing Initial Centroids



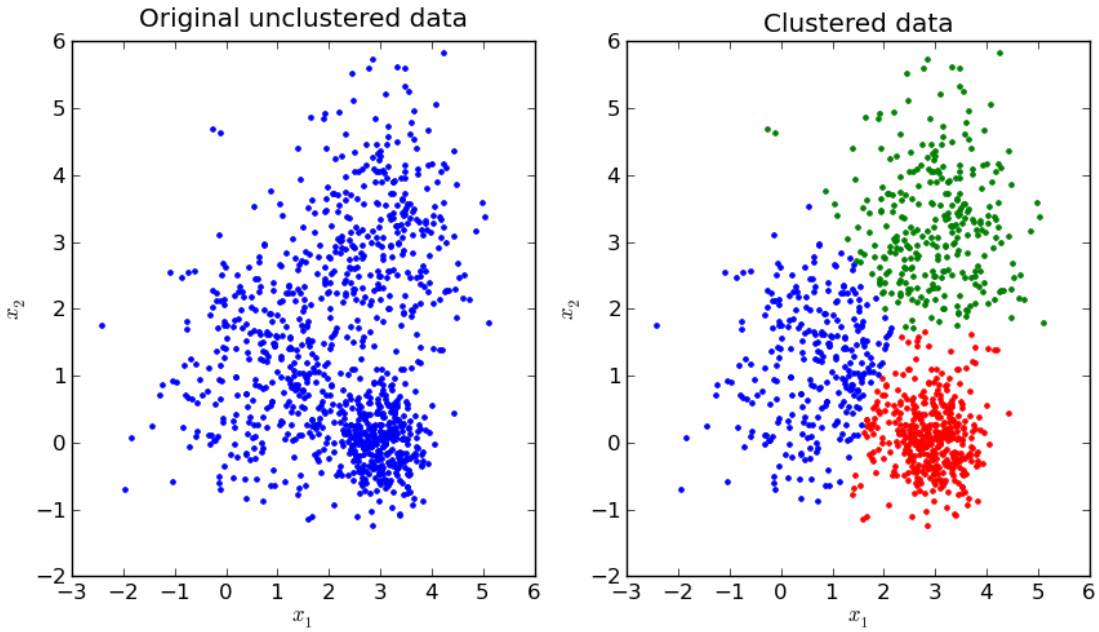
# Evaluating K-means Clusters

- 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?

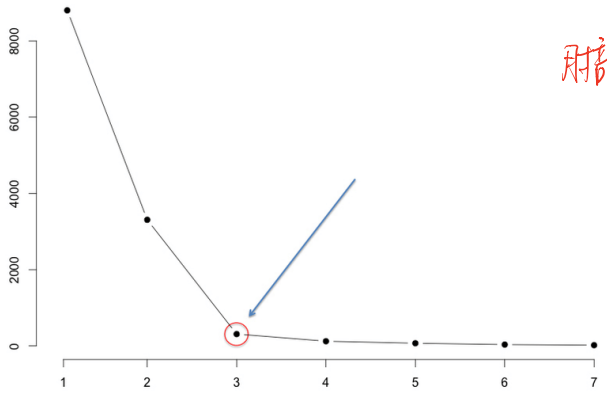



- How do you know K=3?
- Is 3 the optimal K?

不同的初始簇对结果影响非常大.

# 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.

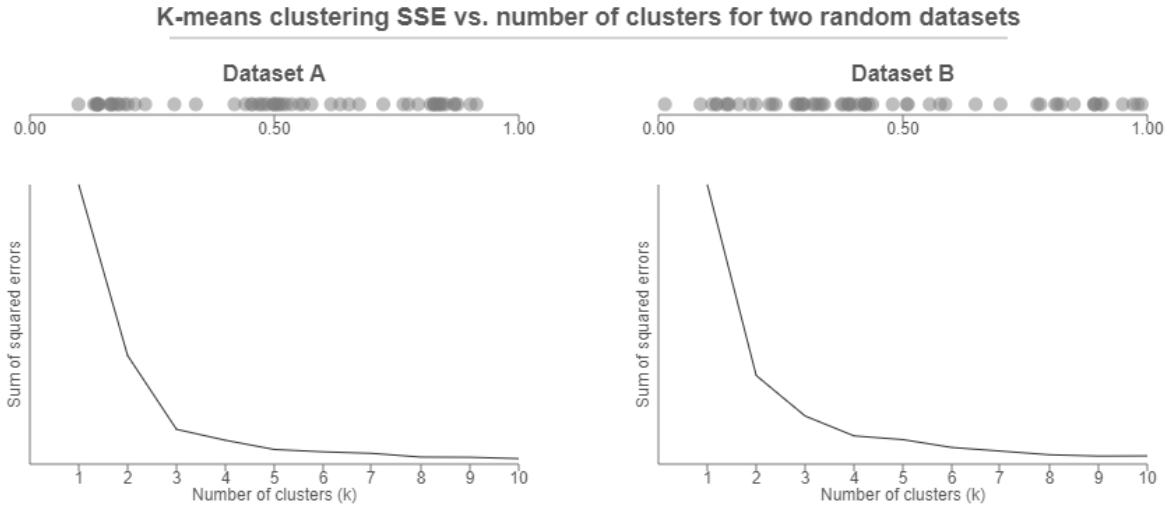



肘部的值开始下降很快，在肘部开始平缓了。

# Determine the Optimal K - Elbow Method

有时候位置不明显 肘部就无法确定k值.

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.

通过画

# Determine the Optimal K - Silhouette Method

轮廓系数方法

- 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

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

si接近1，说明样本i聚类合理

si接近-1，说明样本i更应该分类到别的簇。

若si接近0，则说明样本i在两个簇的边界上。

- ***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. 计算样本i到同簇其它样本的平均距离ai，ai越小，说明样本越来越被聚类到该簇，将ai称为样本i的簇内不相似度。

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. bi越大，说明样本i越不属于其它簇

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

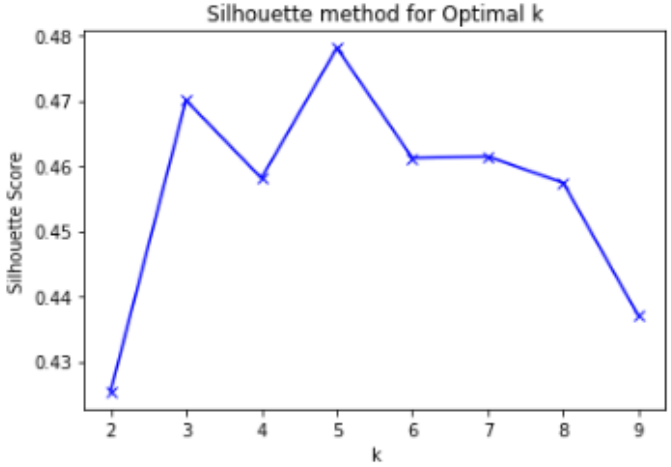
$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j)$$

计算样本i到其它某簇Ck的所有样本的平均距离时，称为样本i与簇Ck的不相似度。定义为样本的簇间不相似度。

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

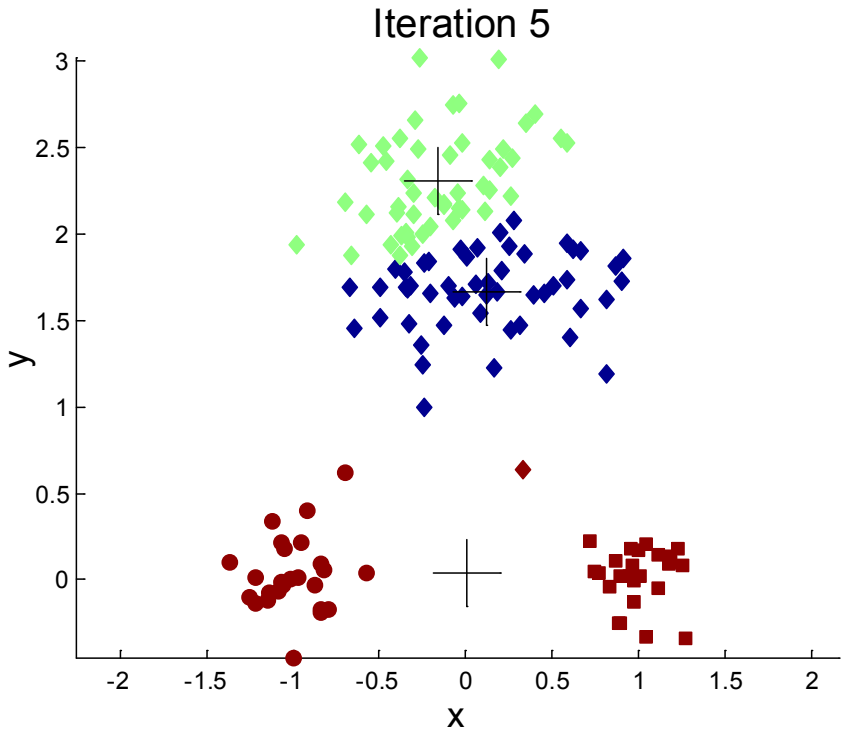
# Determine the Optimal K - Silhouette Method

- 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.

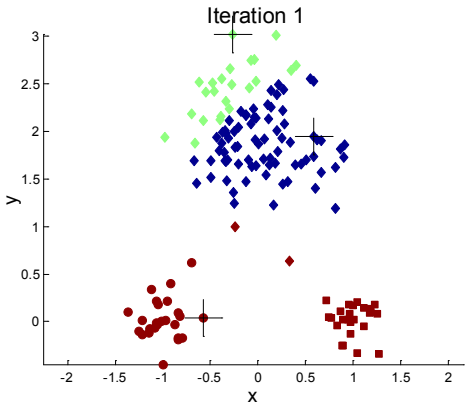


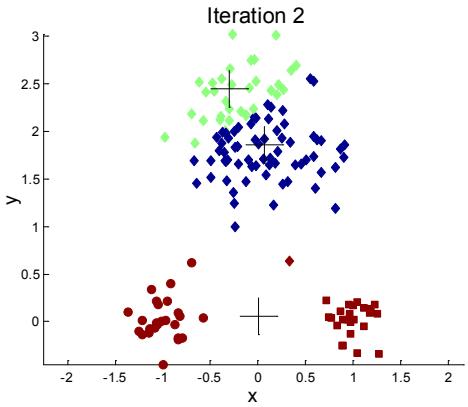

- As per this method k=3 was a local optima, whereas k=5 should be chosen for the number of clusters.

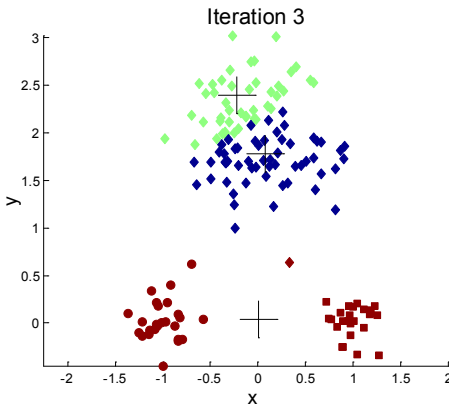
# Importance of Choosing Initial Centroids …

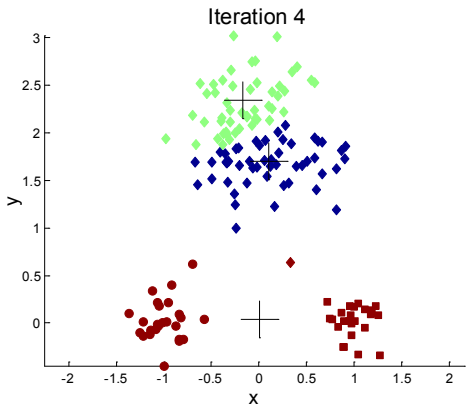


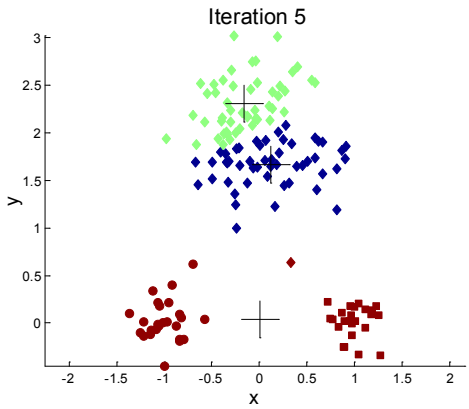
# Importance of Choosing Initial Centroids …



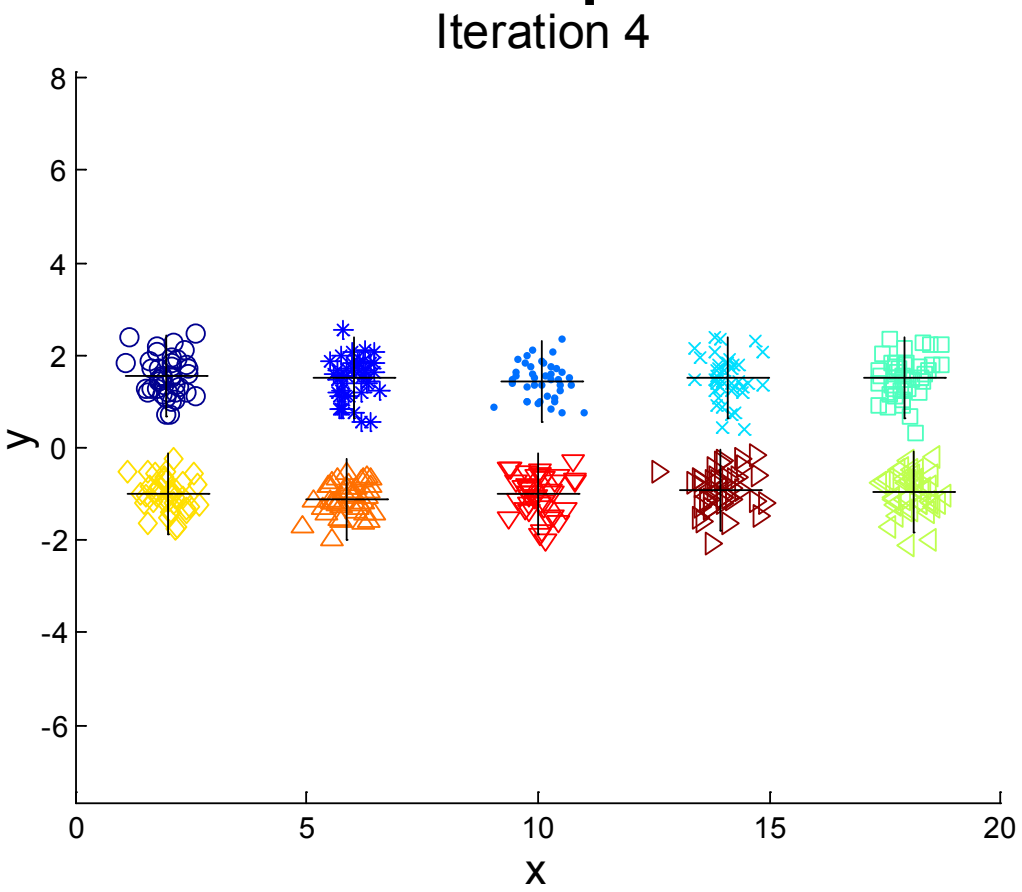
# Problems with Selecting Initial Points

- 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

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

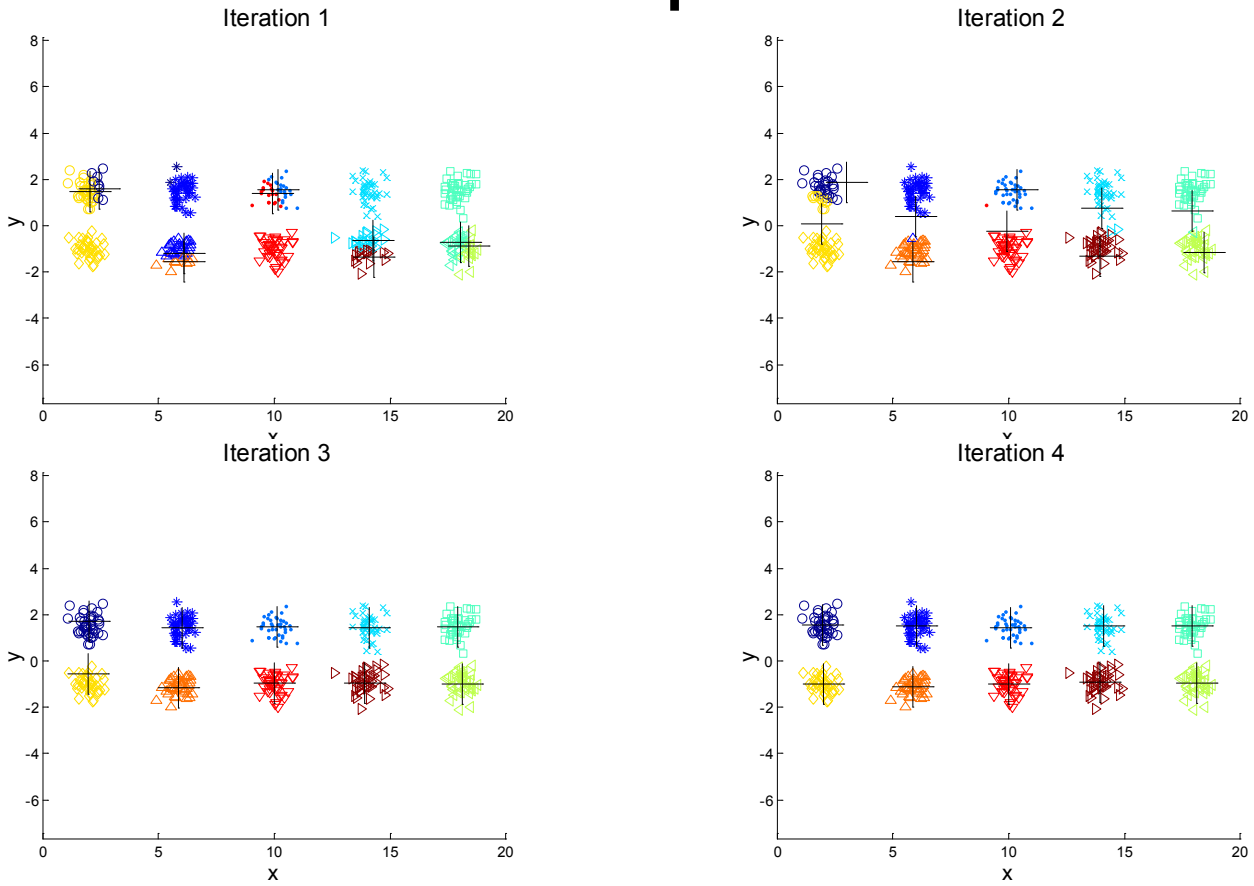
  - For example, if K = 10, then probability = $10!/10^{10}$ = 0.00036
  - Sometimes the initial centroids will readjust themselves in 'right' way, and sometimes they don't
  - Consider an example of five pairs of clusters

# 10 Clusters Example



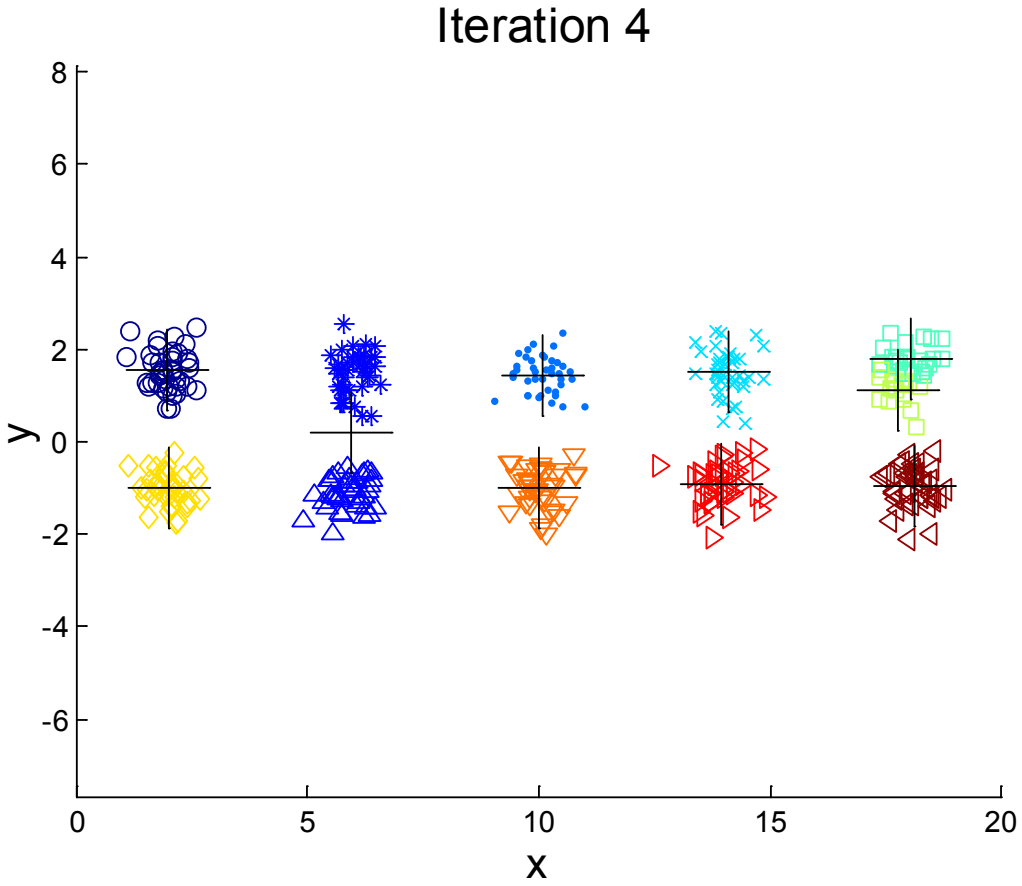

**Starting with two initial centroids in one cluster of each pair of clusters**

# 10 Clusters Example



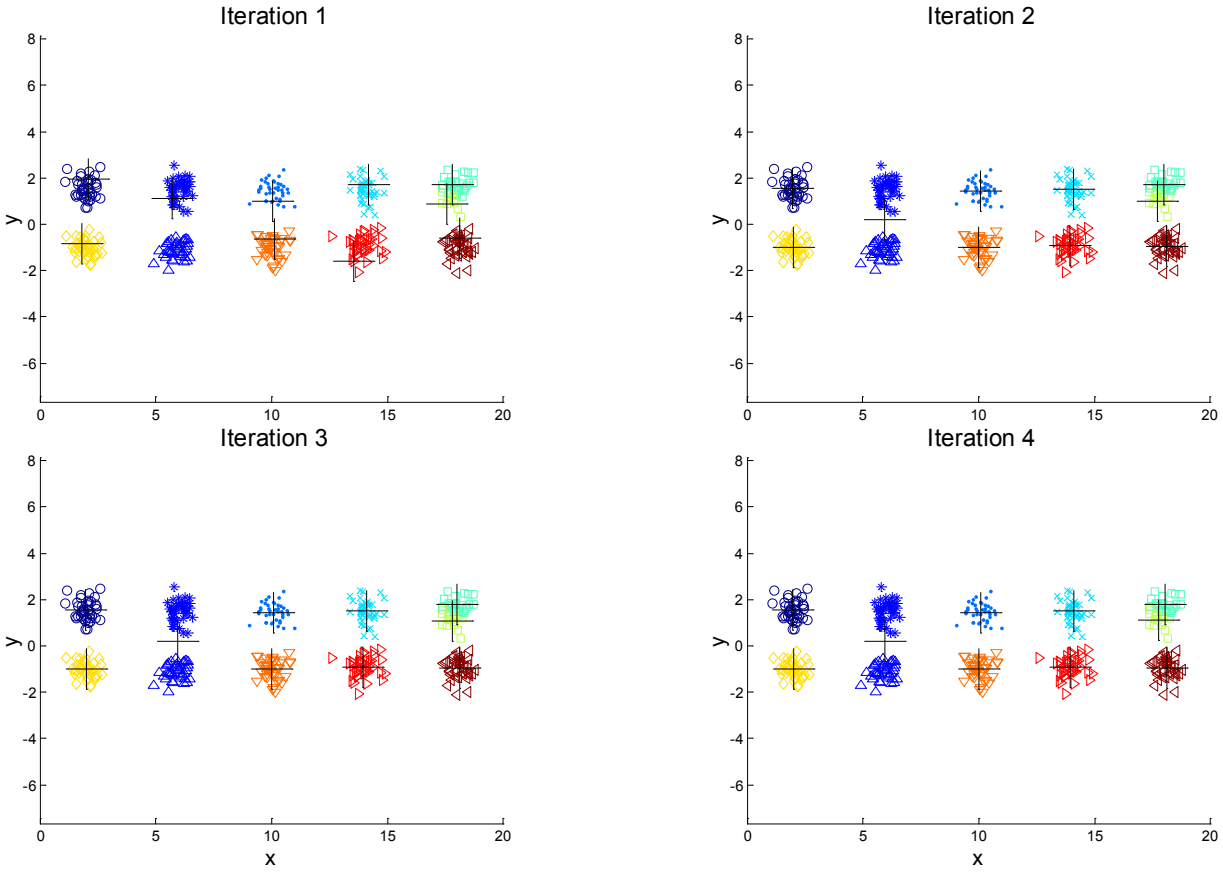

**Starting with two initial centroids in one cluster of each pair of clusters**

# 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

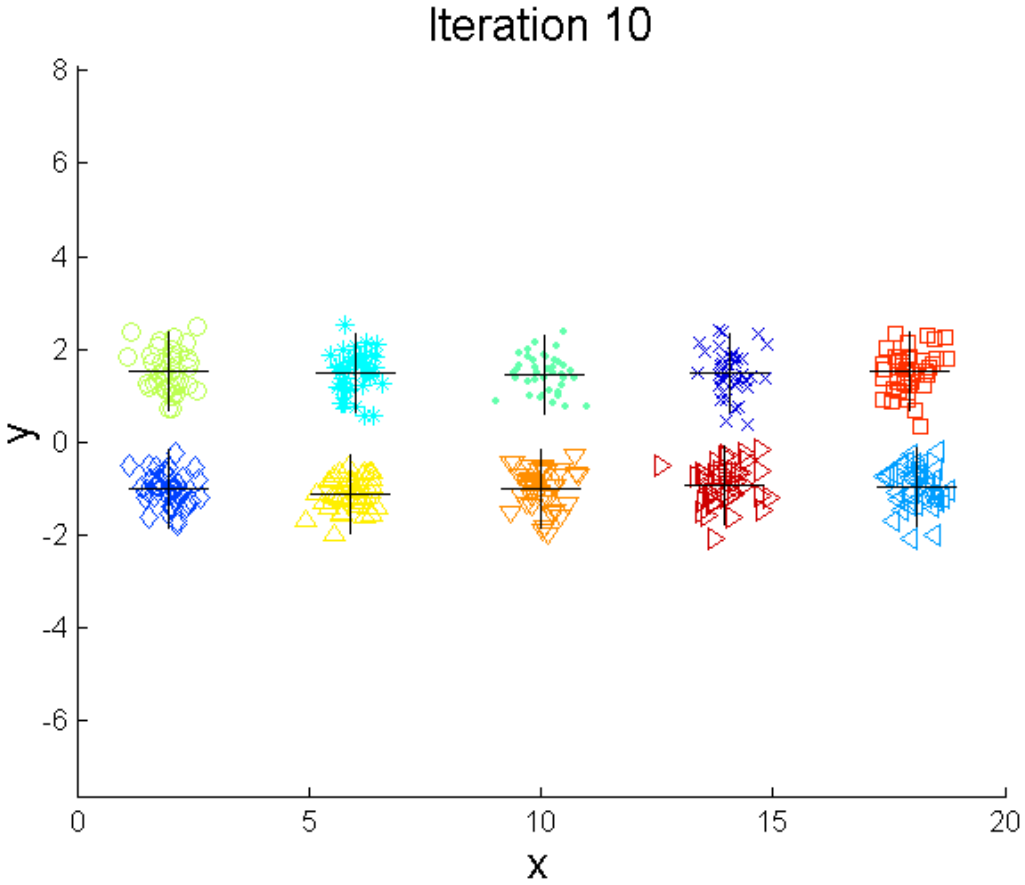
bisecting k-means聚类算法，即二分k均值算法，它是k-means聚类算法的一个变体，主要是为了改进k-means 算法随机选择初始质心的随机性造成聚类结果不确定性的问题，而bisecting k-means算法受随机选择初始质心d 影响比较小

- 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

同k-means算法一样，Bisecting k-means算法不适用于非球形簇的聚类，而且不同尺寸和密度的类型的簇，也不太适合。

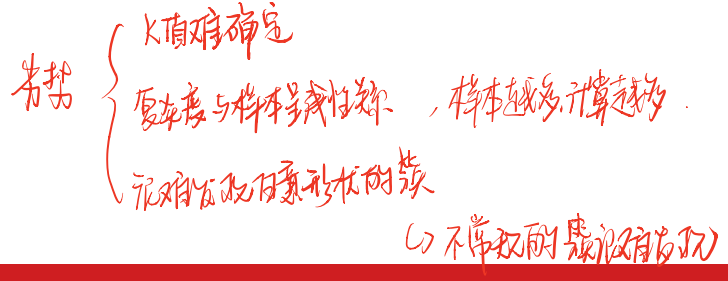
# Bisecting K-means Example



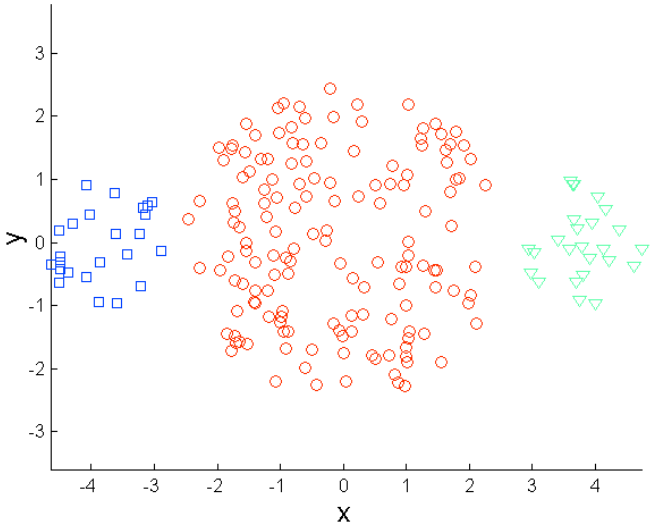
# Limitations of K-means

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

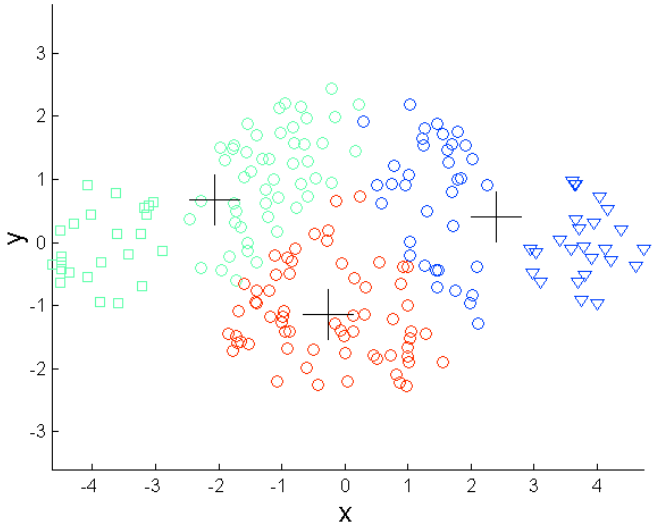
- K-means has problems when the data contains outliers.



# Limitations of K-means: Differing Sizes

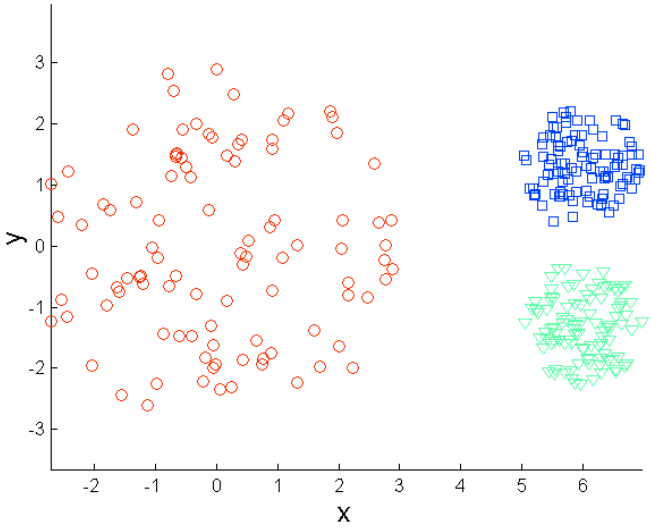



**Original Points**

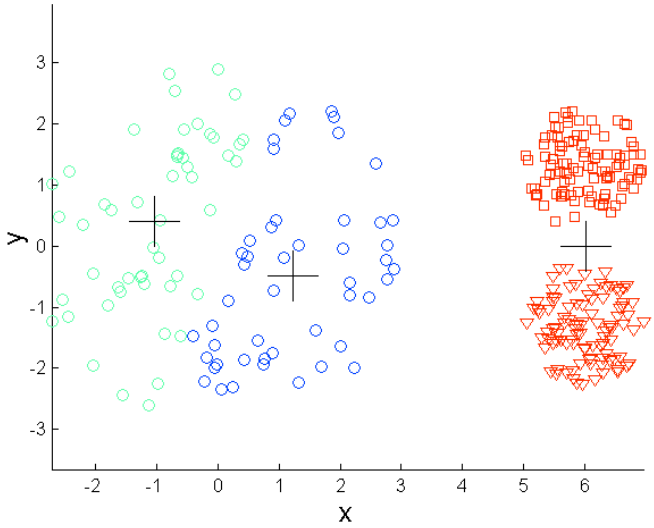



**K-means (3 Clusters)**

# Limitations of K-means: Differing Density

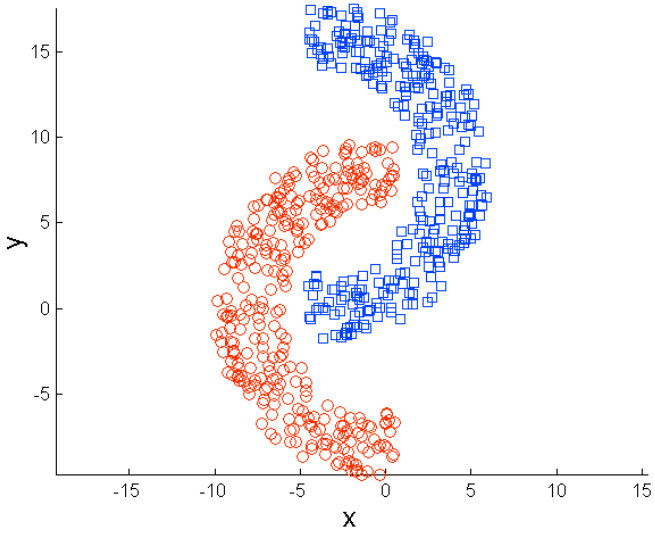



**Original Points**

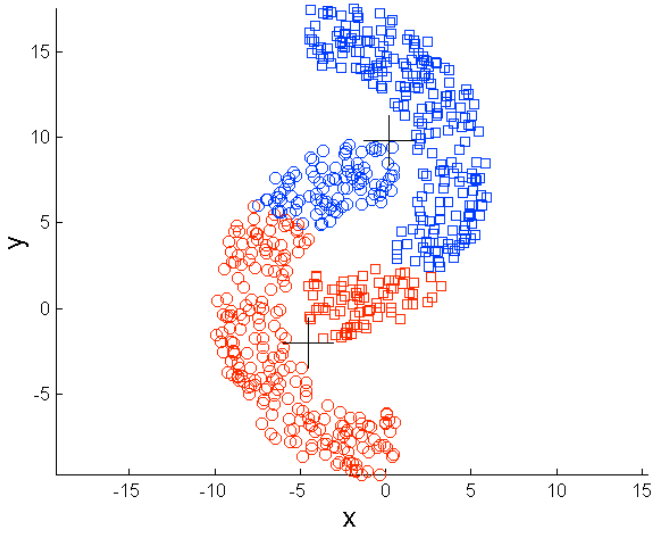



**K-means (3 Clusters)**

# Limitations of K-means: Non-globular Shapes

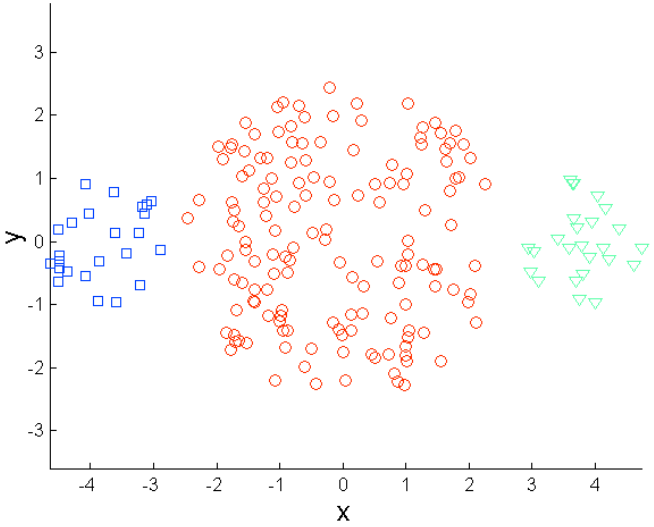



**Original Points**

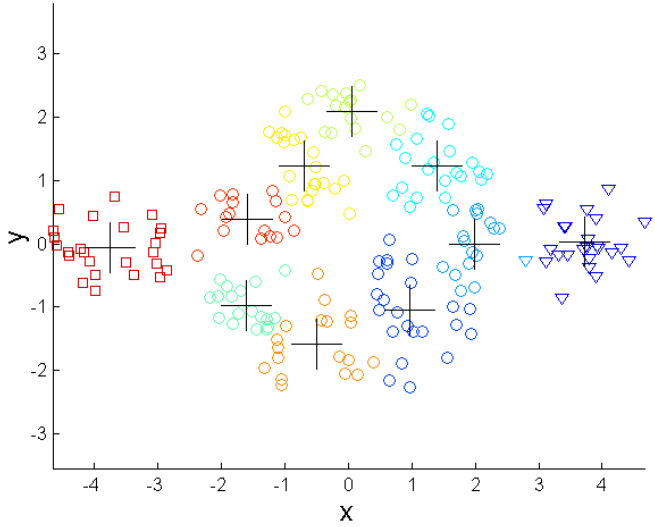



**K-means (2 Clusters)**

# Overcoming K-means Limitations



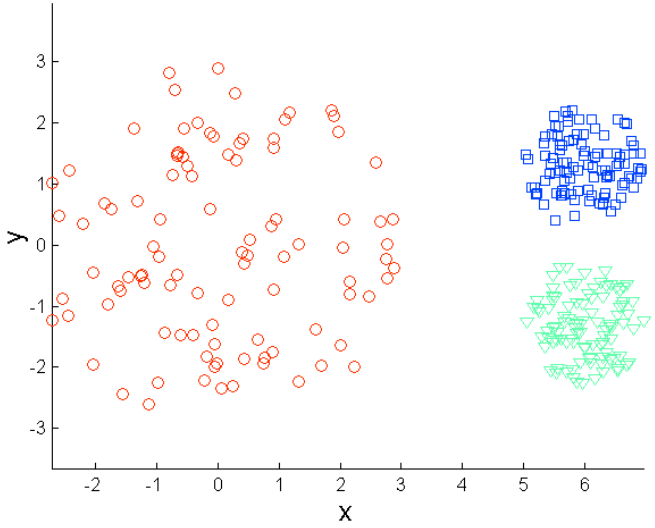

**Original Points**



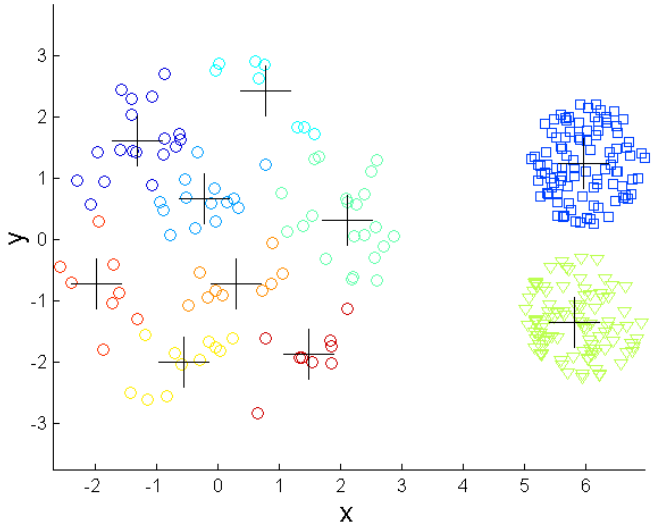

**K-means Clusters**

One solution is to use many clusters.
Find parts of clusters, but need to put together.

# Overcoming K-means Limitations

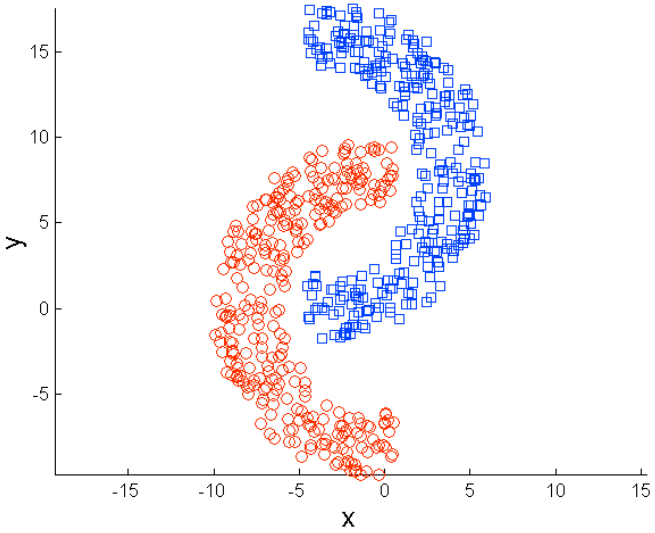


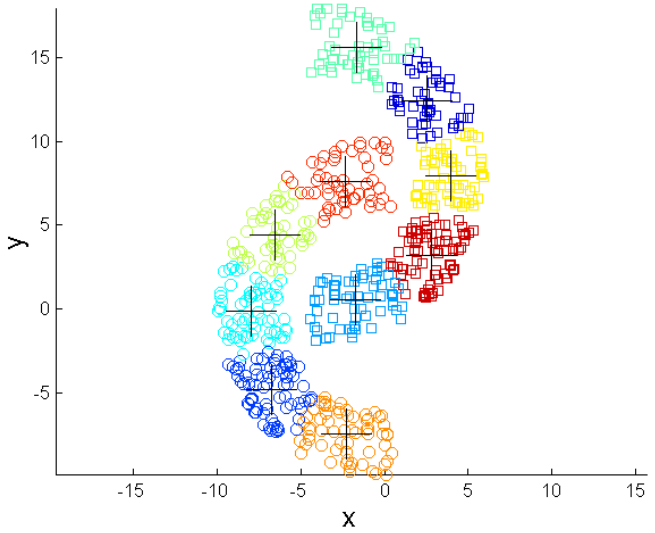

**Original Points**




**K-means Clusters**

# Overcoming K-means Limitations



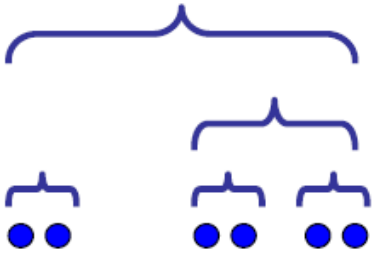

**Original Points** **K-means Clusters**

引入Hierarchical Clustering背景：k-means算法却是一种方便好用的聚类算法，但是始终有K值选择和初始聚类中心点选择的问题，而这些问题也会影响聚类的效果。为了避免这些问题，可以选择另外一种比较实用的聚类算法-层次聚类算法。

2.1 合成聚类合并算法
层次聚类的合并算法通过计算两类数据点间的相似性，对所有数据点中最为相似的两个数据点进行组合，并反复迭代这一过程。简单的说层次聚类的合并算法是通过计算每一个类别的数据点与所有数据点之间的距离来确定它们之间的相似性，距离越小，相似度越高。并将距离最近的两个数据点或类别进行组合，生成聚类树。

# Hierarchical clustering

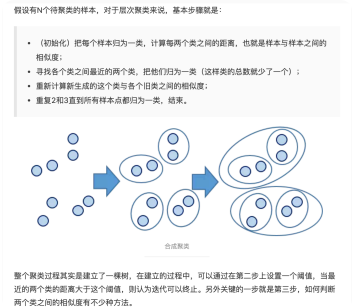
- Probably the most popular clustering algorithm in this area
- First presented in this context by Eisen in 1998



dendrogram

g

- 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



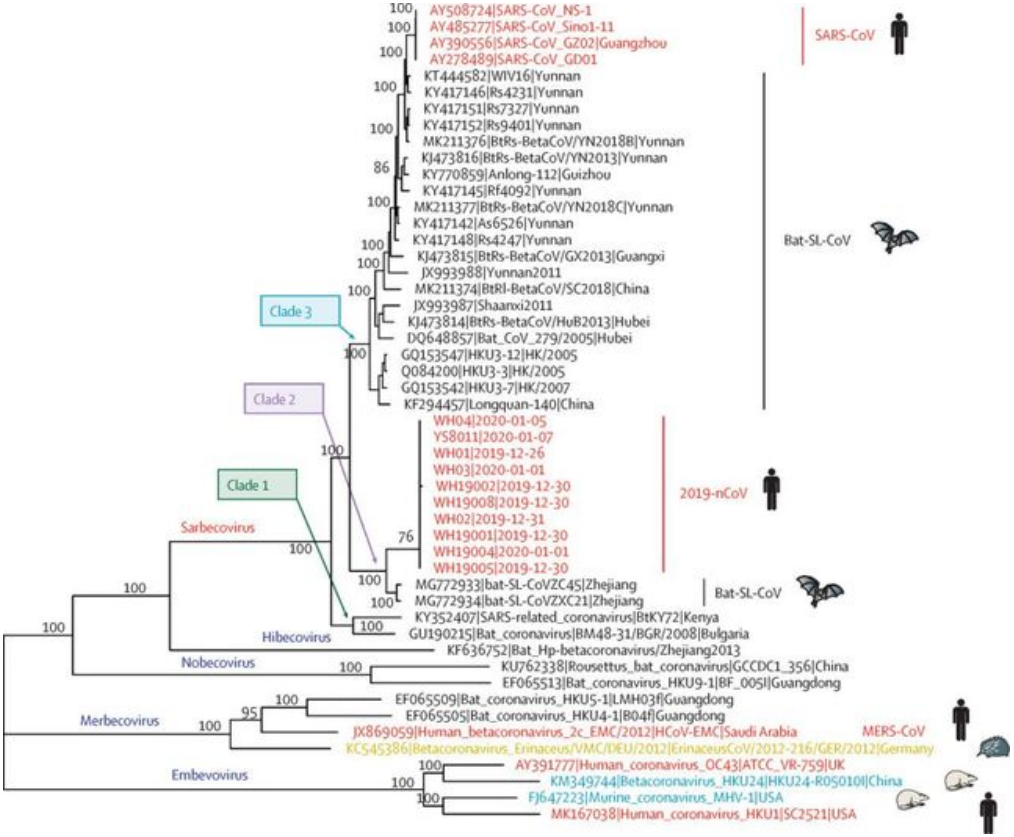
# Strengths of Hierarchical Clustering

- 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

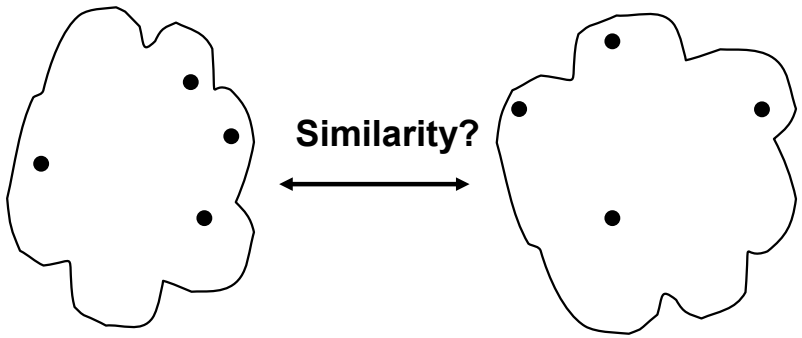
- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)

# 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

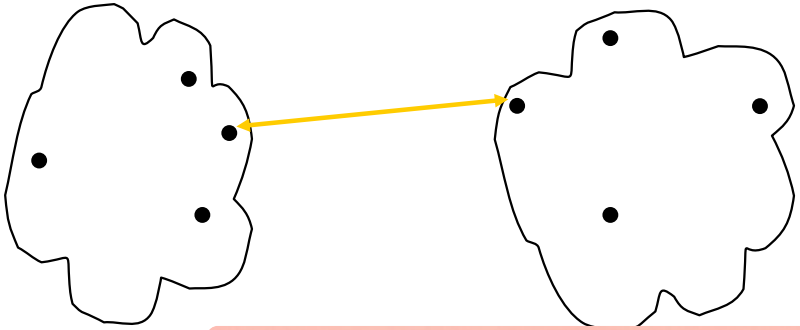



| | p1 | p2 | p3 | p4 | p5 | . . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |
| . | | | | | | |
| . | | | | | | |

**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



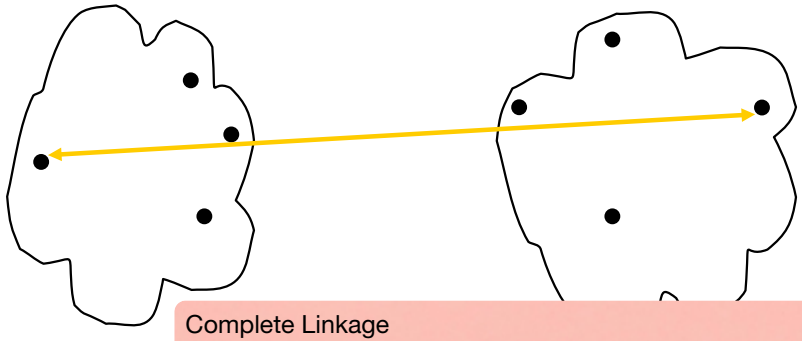
Single Linkage
Single Linkage的计算方法是将两个组合数据点中距离最近的两个数据点间的距离作为这两个组合数据点的距离。这种方法容易受到极端值的影响。两个很相似的组合数据点可能由于其中的某个极端的数据点距离较近而组合在一起。

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

| | p1 | p2 | p3 | p4 | p5 | . . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |
| . | | | | | | |
| . | | | | | | |

**Distance Matrix**

# How to Define Inter-Cluster Similarity



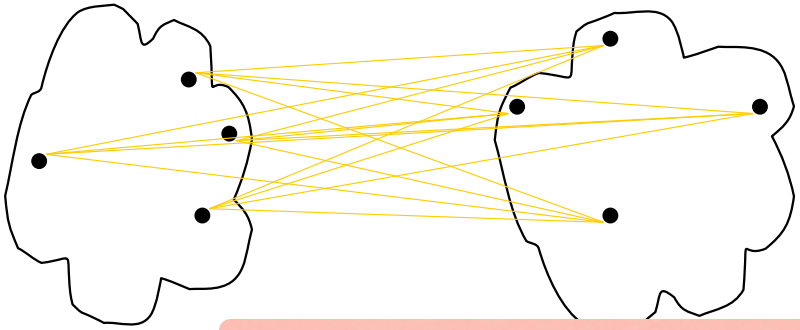
Complete Linkage
Complete Linkage的计算方法与Single Linkage相反，将两个组合数据点中距离最远的两个数据点间的距离作为这两个组合数据点的距离。Complete Linkage的问题也与Single Linkage相反，两个不相似的组合数据点可能由于其中的极端值距离较远而无法组合在一起。

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

| | p1 | p2 | p3 | p4 | p5 | . . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |
| . | | | | | | |
| . | | | | | | |

**Distance Matrix**

# How to Define Inter-Cluster Similarity



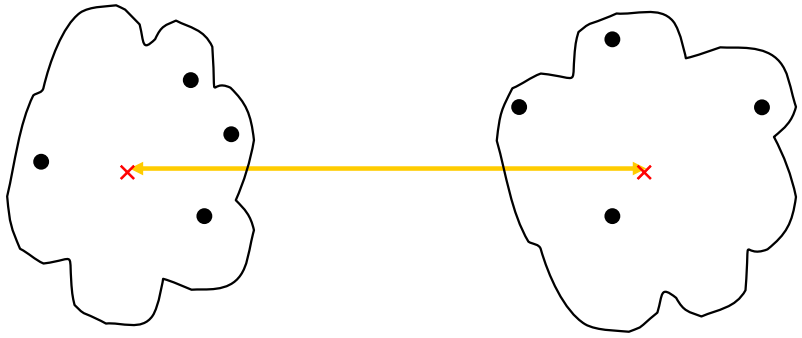
Average Linkage
Average Linkage的计算方法是计算两个组合数据点中的每个数据点与其他所有数据点的距离。将所有距离的均值作为两个组合数据点间的距离。这种方法计算量比较大，但结果比前两种方法更合理。

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

| | p1 | p2 | p3 | p4 | p5 | . . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |
| . | | | | | | |
| . | | | | | | |

**Distance Matrix**

# 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

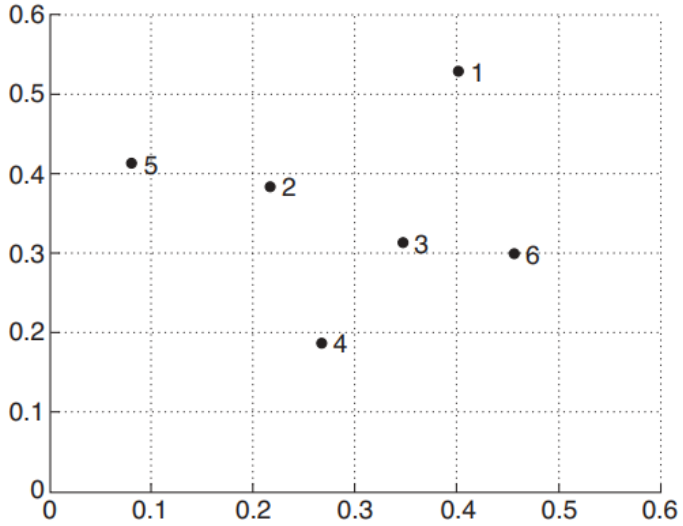
# Cluster Similarity: MIN or Single Link

- 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.

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

# 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 |



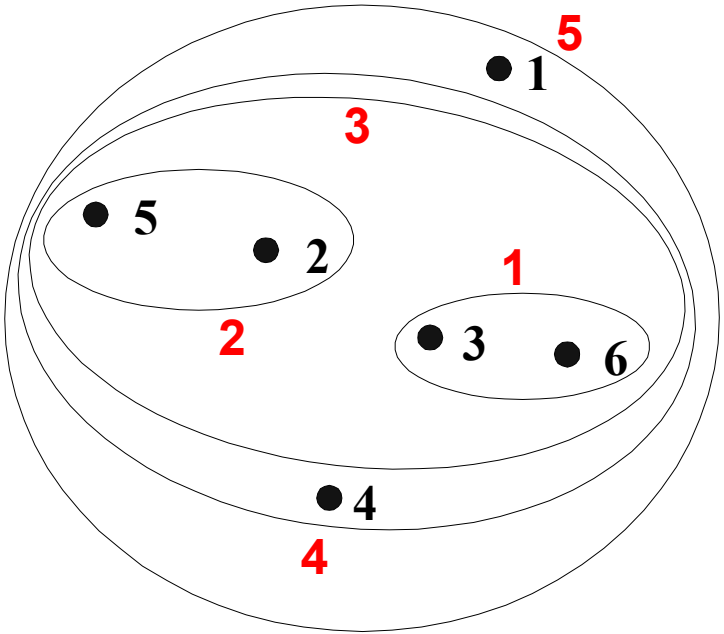
# Hierarchical Clustering: MIN

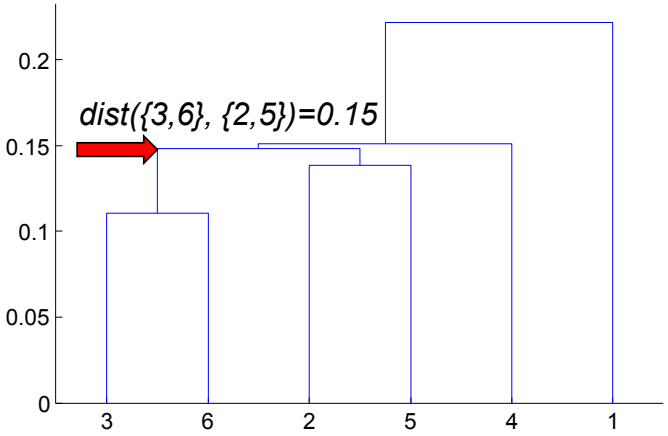
## Euclidian Distance Matrix

| | 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

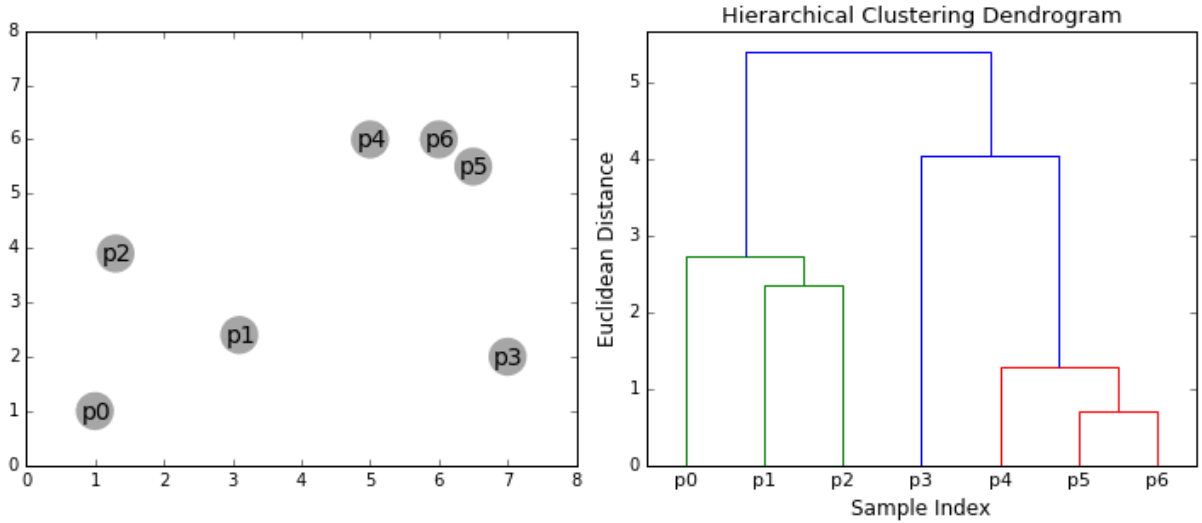



**Nested Clusters**

**Dendrogram**

# Example: MIN


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

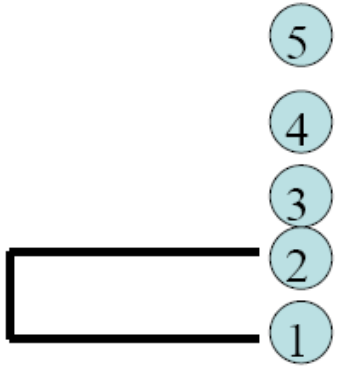
# A Complete Example: MIN

| | 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 |

| | (1,2) | 3 | 4 | 5 |
|---|---|---|---|---|
| (1,2) | 0 | | | |
| 3 | 3 | 0 | | |
| 4 | 9 | 7 | 0 | |
| 5 | 8 | 5 | 4 | 0 |

The minimum distance indicates to merge the two clusters

$$d_{(1,2),3} = \min\{d_{1,3}, d_{2,3}\} = \min\{6,3\} = 3$$
$$d_{(1,2),4} = \min\{d_{1,4}, d_{2,4}\} = \min\{10,9\} = 9$$
$$d_{(1,2),5} = \min\{d_{1,5}, d_{2,5}\} = \min\{9,8\} = 8$$



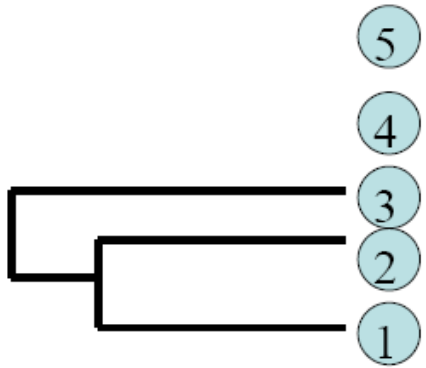
# A Complete Example: MIN

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

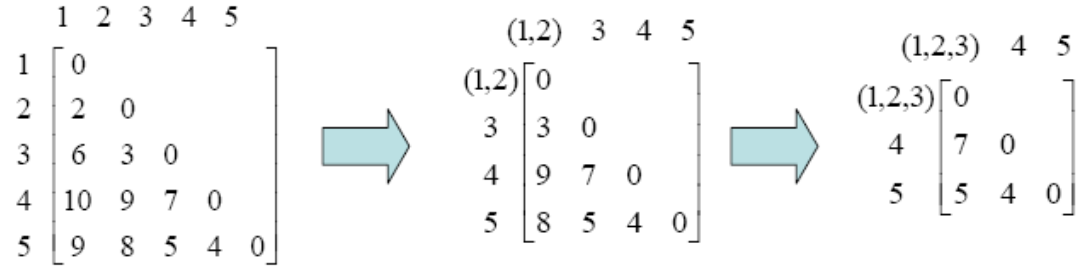
The minimum distance indicates to merge the two clusters

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

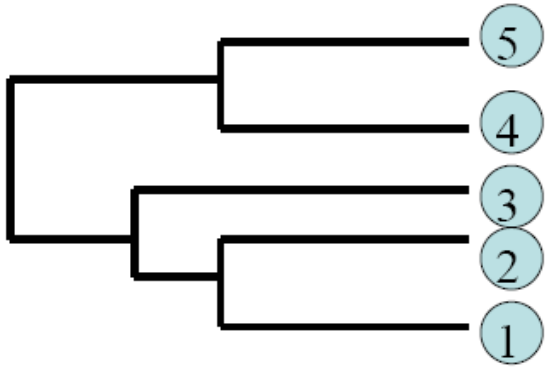
$$d_{(1,2,3),5} = \min\{d_{(1,2),5}, d_{3,5}\} = \min\{8,5\} = 5$$



# A Complete Example: MIN

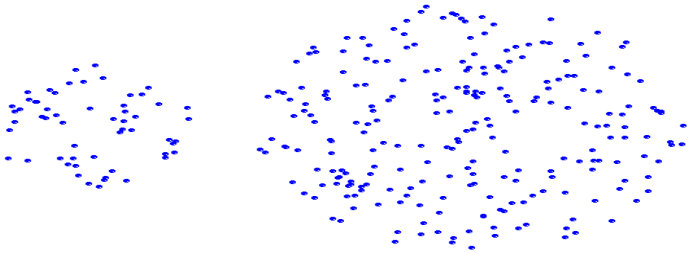



$$d_{(1,2,3),(4,5)} = \min\{d_{(1,2,3),4}, d_{(1,2,3),5}\} = 5$$

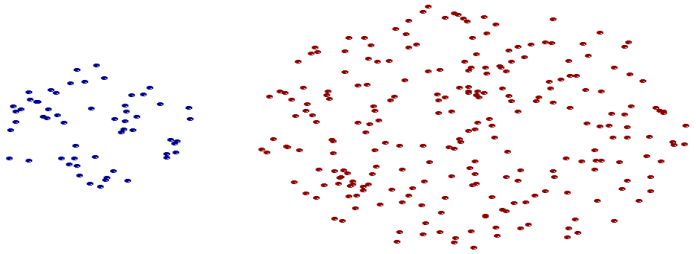


# Strength of MIN

- **Can handle non-elliptical shapes**



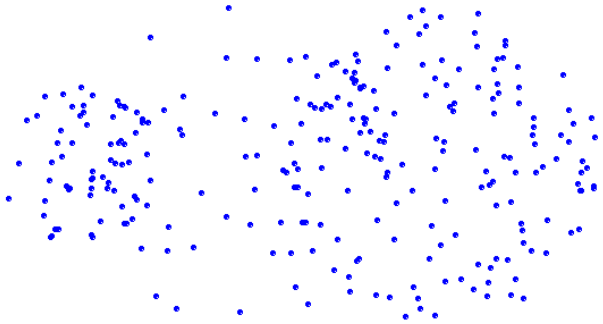
**Original Points**



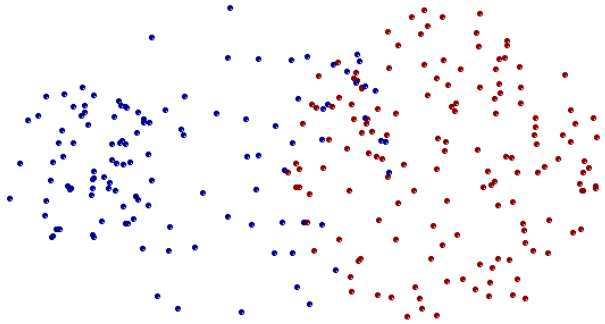
**Two Clusters**


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

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

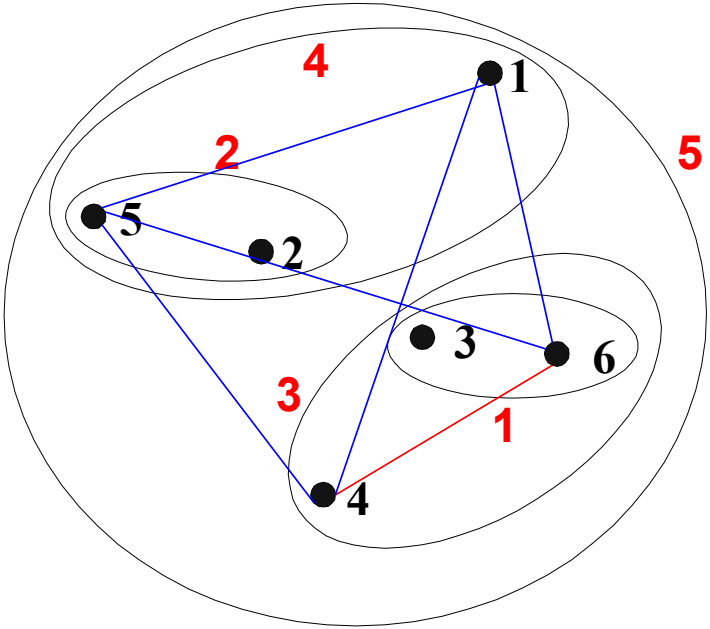
# Hierarchical Clustering: MAX

## Euclidian Distance Matrix

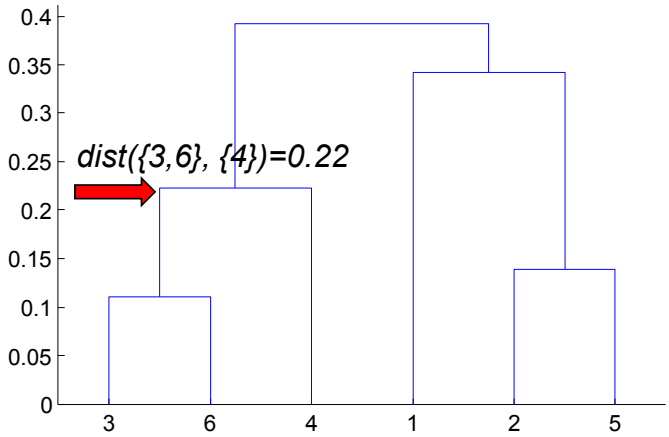
| | P1 | P2 | P3 | P4 | P5 | P6 |
|---|---|---|---|---|---|---|
| P1 | 0 | 0.2357 | 0.2218 | 0.3688 | 0.3421 | 0.2347 |
| P2 | 0.2357 | 0 | 0.1483 | 0.2042 | 0.1388 | 0.2540 |
| P3 | 0.2218 | 0.1483 | 0 | 0.1513 | 0.2843 | 0.1100 |
| P4 | 0.3688 | 0.2042 | 0.1513 | 0 | 0.2932 | 0.2216 |
| P5 | 0.3421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

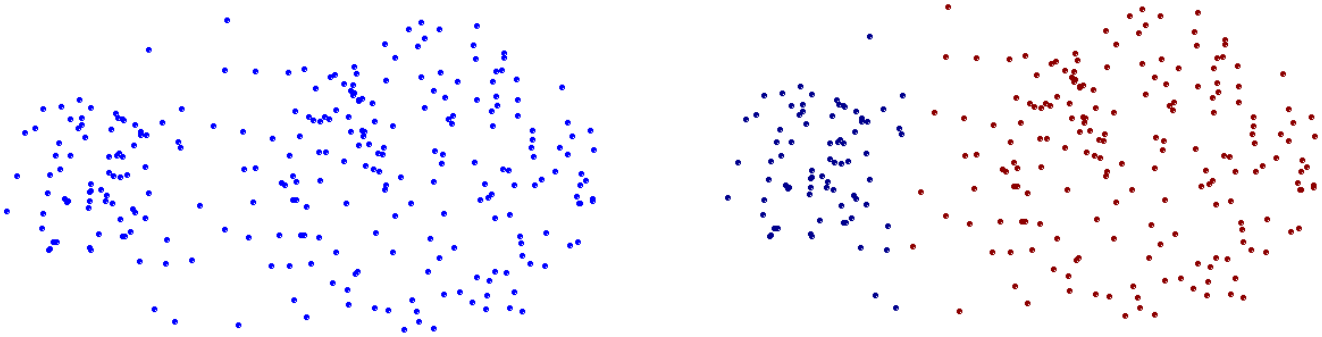



Nested Clusters




Dendrogram

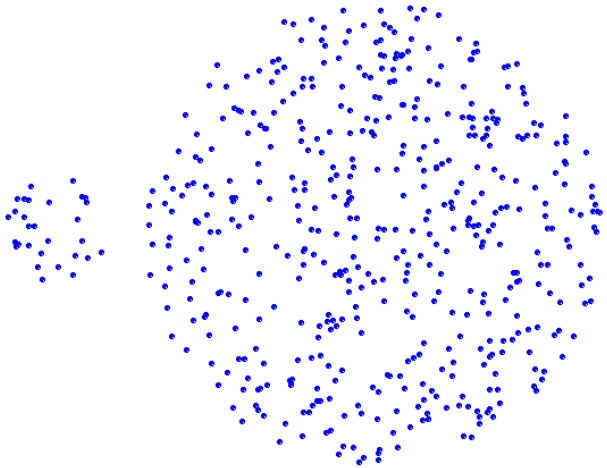
# Strength of MAX



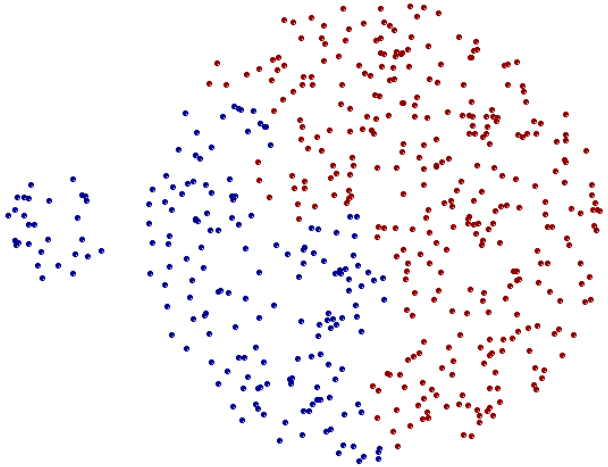
**Original Points** **Two Clusters**

- **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

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

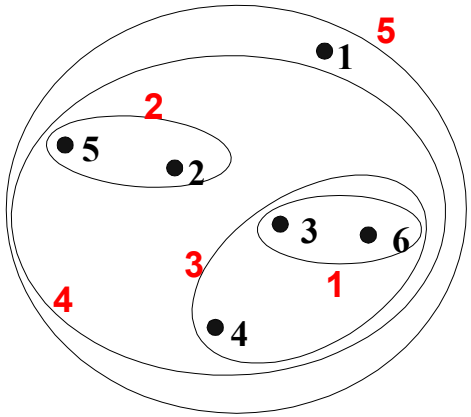
# Hierarchical Clustering: Group Average

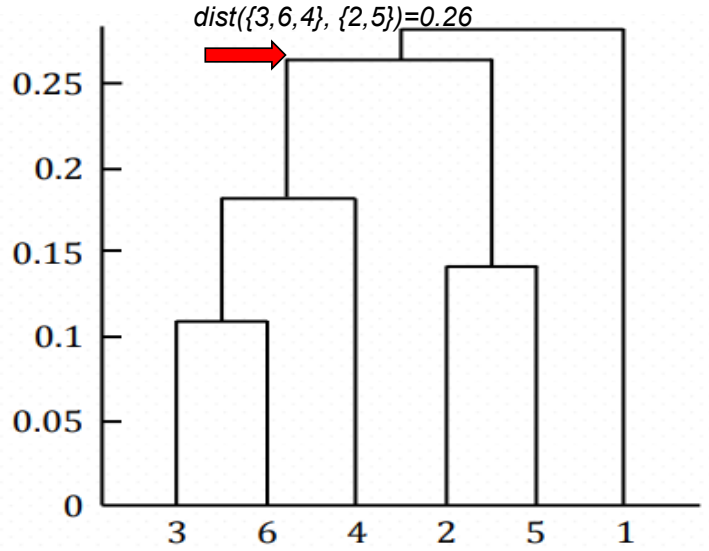
## Euclidian Distance Matrix

| | 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



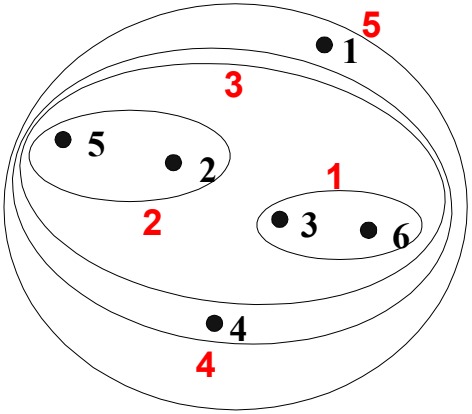

**Nested Clusters**

**Dendrogram**

# Hierarchical Clustering: Group Average

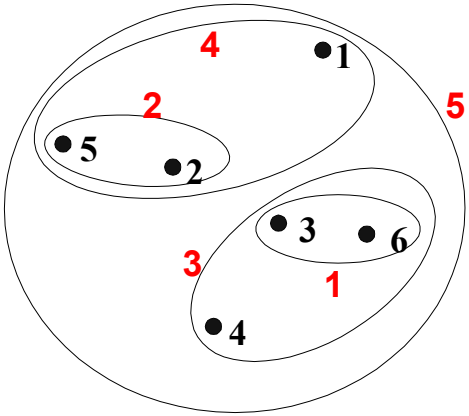
- Compromise between Single and Complete Link
- Strengths
  - Less susceptible to noise and outliers
- Limitations
  - Biased towards globular clusters

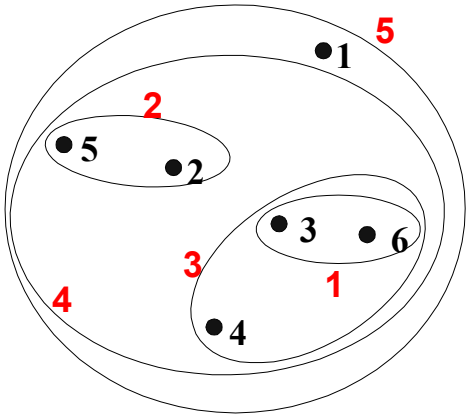
# Hierarchical Clustering: Comparison




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是基于密度空间的聚类算法，与KMeans算法不同，它不需要确定聚类的数量，而是基于数据推测聚类的数目，它能够针对任意形状产生聚类。

- 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

DBSCAN算法需要首先确定两个参数：
（1）epsilon:在一个点周围邻近区域的半径
（2）minPts:邻近区域内至少包含点的个数
根据以上两个参数，结合epsilon-neighborhood的特征，可以把样本中的点分成三类：

1. 核点（core point）：满足NBHD(p,epsilon)>=minPts，则为核样本点
2. 边缘点（border point）：NBHD(p,epsilon)<minPts，但是该点可由一些核点获得（*density-reachable*或者*directly-reachable*）
3. 离群点（Outlier）：既不是核点也不是边缘点，则是不属于这一类的点

- 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.

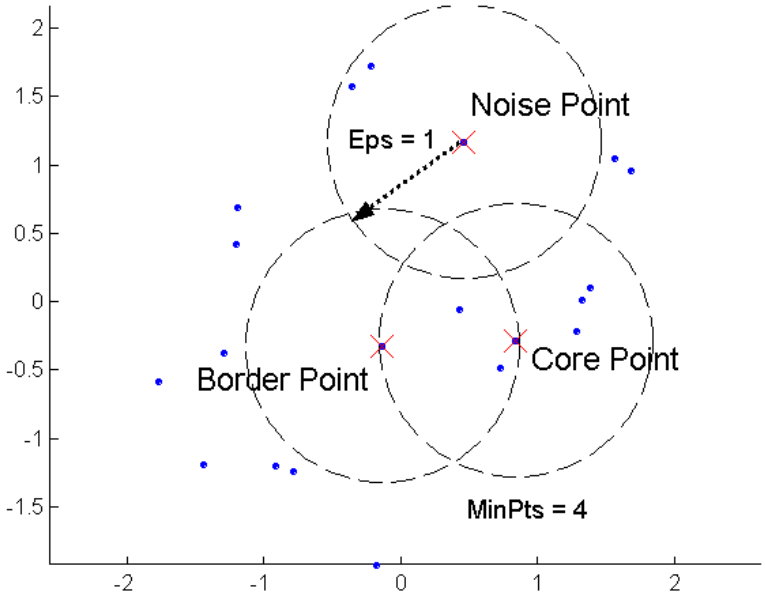
# DBSCAN: Core, Border, and Noise Points

1、根据 eps 邻域和密度阈值 MinPts，判断一个点是核心点、边界点或者离群点。并将离群点删除

2、如果核心点之间的距离小于 MinPts，就将两个核心点连接在一起。这样就形成了若干组簇

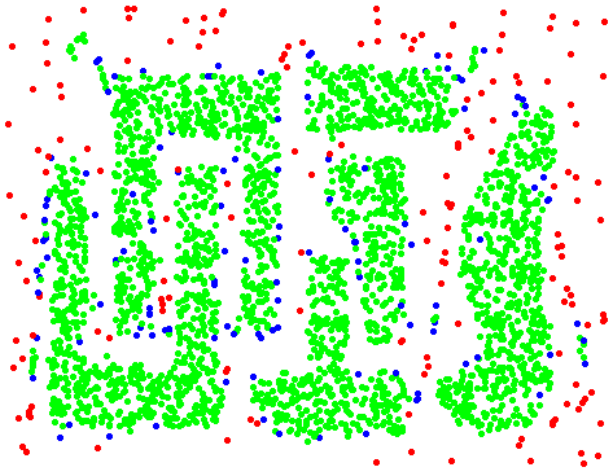
3、将边界点分配到距离它最近的核心点范围内

4、形成最终的聚类结果



# DBSCAN: Core, Border and Noise Points

Original Points

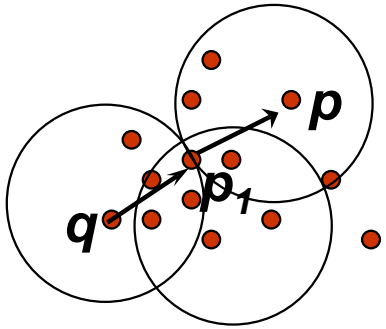


border and noise

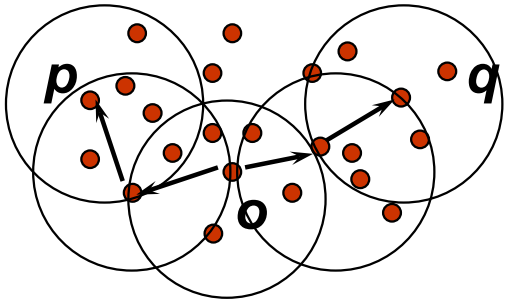
Eps = 10, MinPts = 4

# 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

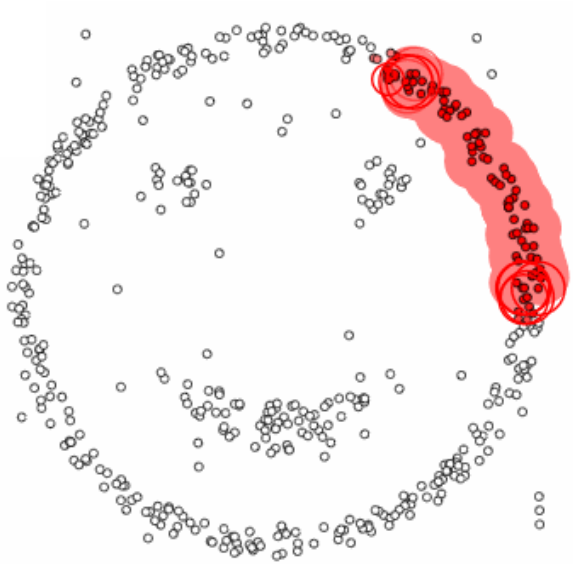
- Label points as core, border and noise
- Eliminate noise points
- For every core point p that has not been assigned to a cluster
  - Create a new cluster with the point p and all the points that are density-connected to p.
- Assign border points to the cluster of the closest core point.
- (very similar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

# DBSCAN Algorithm

对于算法的实现，首先我们概要地描述一下实现的过程：

1. 解析样本数据文件
2. 计算每个点与其他所有点之间的欧几里德距离
3. 计算每个点的k-距离值，并对所有点的k-距离集合进行升序排序，输出的排序后的k-距离值
4. 将所有点的k-距离值，在Excel中用散点图显示k-距离变化趋势
5. 根据散点图确定半径Eps的值
6. 根据给定MinPts=4，以及半径Eps的值，计算所有核心点，并建立核心点与到核心点距离小于半径Eps的点的映射
7. 根据得到的核心点集合，以及半径Eps的值，计算能够连通的核心点，并得到离群点
8. 将能够连通的每一组核心点，以及到核心点距离小于半径Eps的点，都放到一起，形成一个簇
9. 选择不同的半径Eps，使用DBSCAN算法聚类得到的一组簇及其离群点，使用散点图对比聚类效果

然后，再详细描述聚类过程的具体实现。



epsilon = 1.00
minPoints = 4

Restart

Pause

根据经验计算半径Eps：根据得到的所有点的k-距离集合E，对集合E进行升序排序后得到k-距离集合E'，需要拟合一条排序后的E'集合中k-距离的变化曲线图，然后绘出曲线，通过观察，将急剧发生变化的位置所对应的k-距离的值，确定为半径Eps的值。

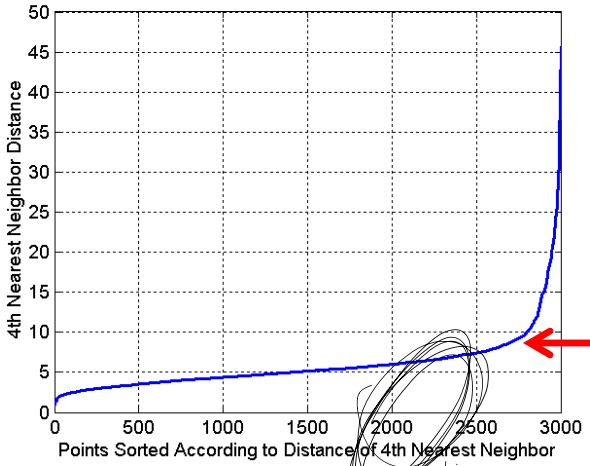
根据经验计算最少点的数量MinPts：确定MinPts的大小，实际上也是确定k-距离中k的值，DBSCAN算法取k=4，则MinPts=4。

另外，如果觉得经验值聚类的结果不满意，可以适当调整Eps和MinPts的值，经过多次迭代计算对比，选择最合适的参数值。可以看出，如果MinPts不变，Eps取得值过大，会导致大多数点都聚到同一个簇中，Eps过小，会导致一个簇的分裂；如果Eps不变，MinPts的值取得过大，会导致同一个簇中点被标记为离群点，MinPts过小，会导致发现大量的核心点。

# 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

DBSCAN聚类使用到一个k-距离的概念，k-距离是指：给定数据集P={p(i); i=0,1,…n}，对于任意点P(i)，计算点P(i)到集合D的子集S={p(1), p(2), …, p(i-1), p(i+1), …, p(n)}中所有点之间的距离，距离按照从小到大的顺序排序，假设排序后的距离集合为D={d(1), d(2), …, d(k-1), d(k), d(k+1), …,d(n)}，则d(k)就被称为k-距离。也就是说，k-距离是点p(i)到所有点（除了p(i)点）之间距离第k近的距离。对待聚类集合中每个点p(i)都计算k-距离，最后得到所有点的k-距离集合E={e(1), e(2), …, e(n)}。

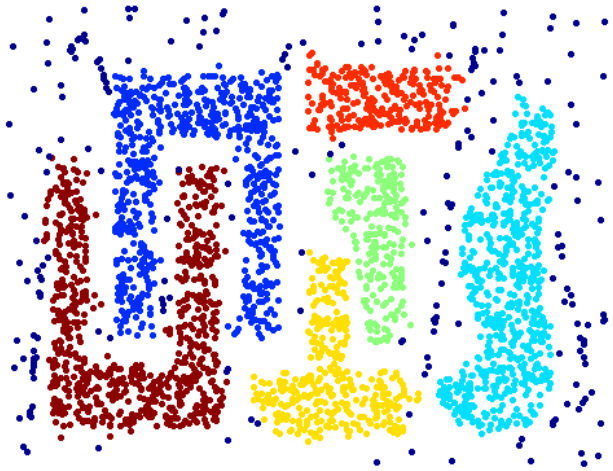



It is the radius crossing the border between dense region and sparse region

Eps ~ 7-10
MinPts = 4

# When DBSCAN Works Well

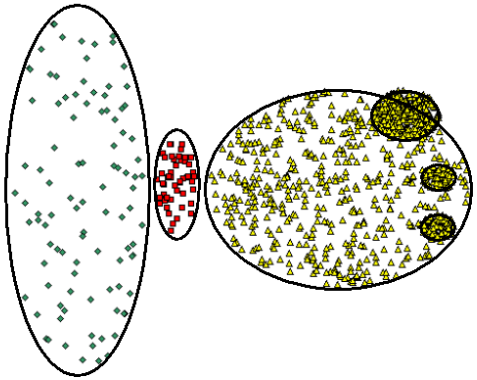
Original Points



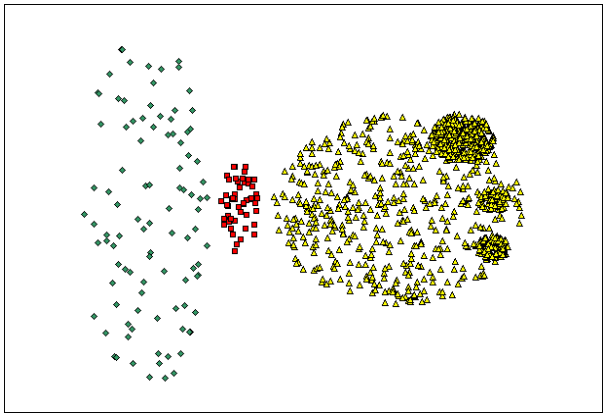
Clusters

- 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

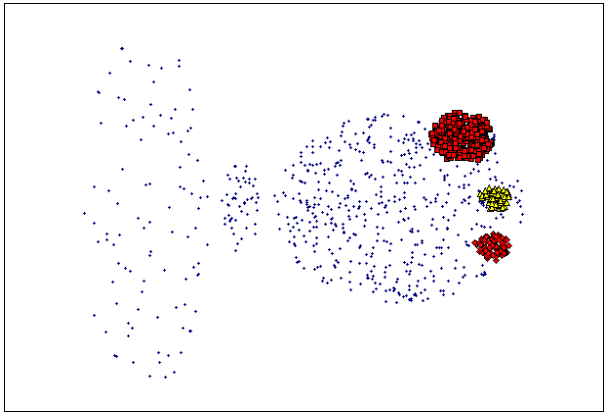
# When DBSCAN Does NOT Work Well



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

根据经验计算半径Eps：根据得到的所有点的k-距离集合E，对集合E进行升序排序后得到k-距离集合E'，需要拟合一条排序后的E'集合中k-距离的变化曲线图，然后绘出曲线，通过观察，将急剧发生变化的位置所对应的k-距离的值，确定为半径Eps的值。
根据经验计算最少点的数量MinPts：确定MinPts的大小，实际上也是确定k-距离中k的值，DBSCAN算法取k=4，则MinPts=4。
另外，如果觉得经验值聚类的结果不满意，可以适当调整Eps和MinPts的值，经过多次迭代计算对比，选择最合适的参数值。可以看出，如果MinPts不变，Eps取得值过大，会导致大多数点都聚到同一个簇中，Eps过小，会导致一个簇的分裂；如果Eps不变，MinPts的值取得过大，会导致同一个簇中点被标记为离群点，MinPts过小，会导致发现大量的核心点。

DBSCAN优点：

1、对噪声不敏感。这是因为该算法能够较好地判断离群点，并且即使错判离群点，对最终的聚类结果也没什么影响

2、能发现任意形状的簇。这是因为DBSCAN 是靠不断连接邻域此高密度点来发现簇的，只需要定义邻域大小和密度阈值，因此可以发现不同形状，不同大小的簇

DBSCAN 缺点：

1、对两个参数的设置敏感，即圈的半径 eps 、阈值 MinPts。

2、DBSCAN 使用固定的参数识别聚类。显然，当聚类的稀疏程度不同，聚类效果也有很大不同。即数据密度不均匀时，很难使用该算法

# 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

# The rest of the semester

- Linear Models for Regression
- Linear Models for Classification
- Logistic Regression Model
- Evaluating Predictors
- Feature and Model Selection
- Regularization
- Support Vector Machines
- Decision Tree/Random Forest
- K-nearest neighbors
- Neural Network
- We may cover deep learning if we have time

# Final Project

- **House Price Prediction**
  - https://www.kaggle.com/c/house-prices-advanced-regression-techniques
- Goals
  - We are not competing with teams on kaggle
  - Group project, 1-4 people per team
  - You can use R or Python
  - Build one prediction model using the ML algorithms of this course
  - Evaluate your prediction model
  - Try different ways to improve your model and show the improvements.
  - Submit code and results in Jupyter in a presentation style on canvas
  - Deadline July 31st midnight.

# 机器学习—聚类系列-层次聚类（Hierarchical Clustering）



引入Hierarchical Clustering背景：k-means算法却是一种方便好用的聚类算法，但是始终有K值选择和初始聚类中心点选择的问题，而这些问题也会影响聚类的效果。为了避免这些问题，可以选择另外一种比较实用的聚类算法-层次聚类算法。

**1、Hierarchical Clustering的作用及类别**

**1.1Hierarchical Clustering**：一如其字面意思，是层次化的聚类，得出来的是树形结构（计算机科学的树是一棵根在最上的树），HC不需要指定具体类别数目的，其得到的是一颗树，聚类完成之后，可在任意层次横切一刀，得到指定数目的 cluster，具体有两种方式一种是在下而上凝聚方法（agglomerative：先将所有样本的每个点都看成一个簇，然后找出距离最小的两个簇进行合并，不断重复到预期簇或者其他终止条件），另一种自上而下分裂方法（divisive：先将所有样本当作一整个簇，然后找出簇中距离最远的两个簇进行分裂，不断重复到预期簇或者其他终止条件）。

**2、Agglomerative HC运行的4个步骤详解**

2.1、把每个样本归为一类，计算每两个类之间的距离，也就是样本与样本之间的相似度；

2.2、寻找各个类之间最近的两个类，把他们归为一类（这样类的总数就少了一个）.

2.3、重新计算新生成的这个类与各个旧类之间的相似度.

2.4、重复2和3直到所有样本点都归为一类，结束。

**3、Dendrogram工作机制解密：**

3.1 计算两个点之间的距离 ，欧式距离（平面几何）。

3.2 计算两个簇之间的距离（Distance between two Cluster），有下面四种方式：

1.Closest Points ：两个簇最近点的距离

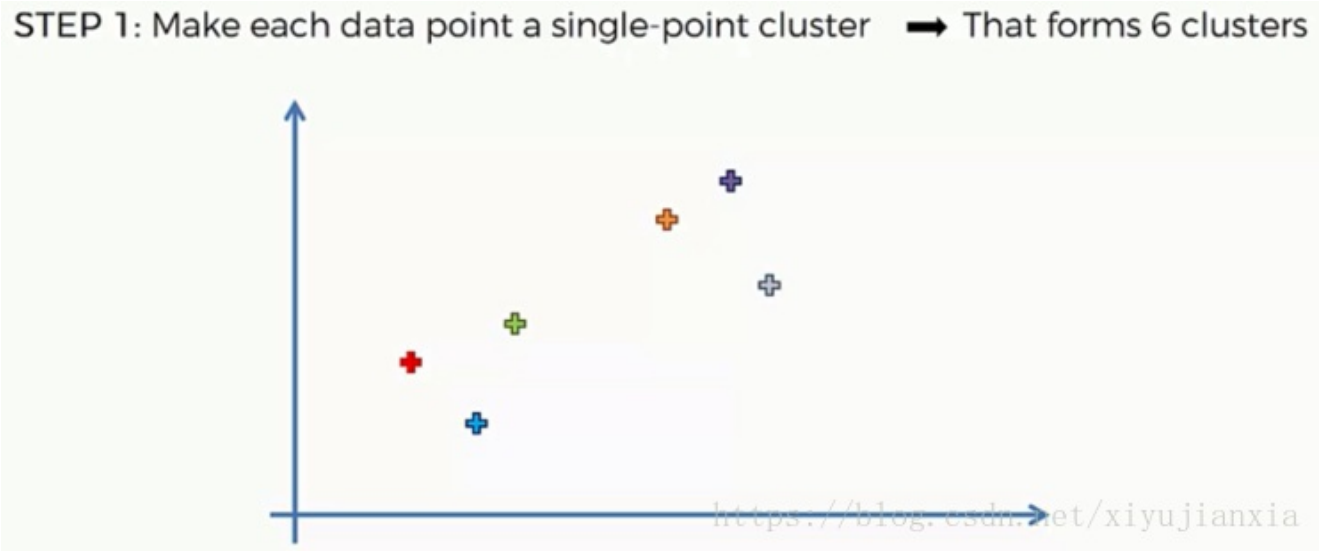
2.Furthest Points ：两个簇最远的点的距离

3.Average Points ：两个簇所有点两两距离平均值。

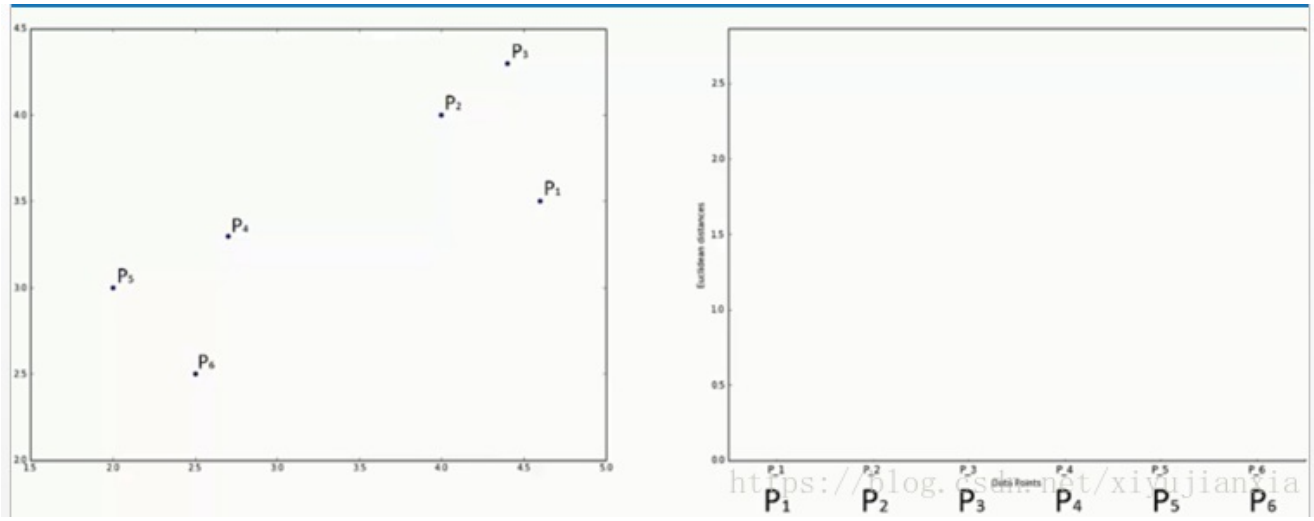
3.Distance Between Centroids 两个簇 簇中心点之间的距离

**4、 Hierarchical Clustering在内部是如何使用Dendrogram进行Clustering的？**

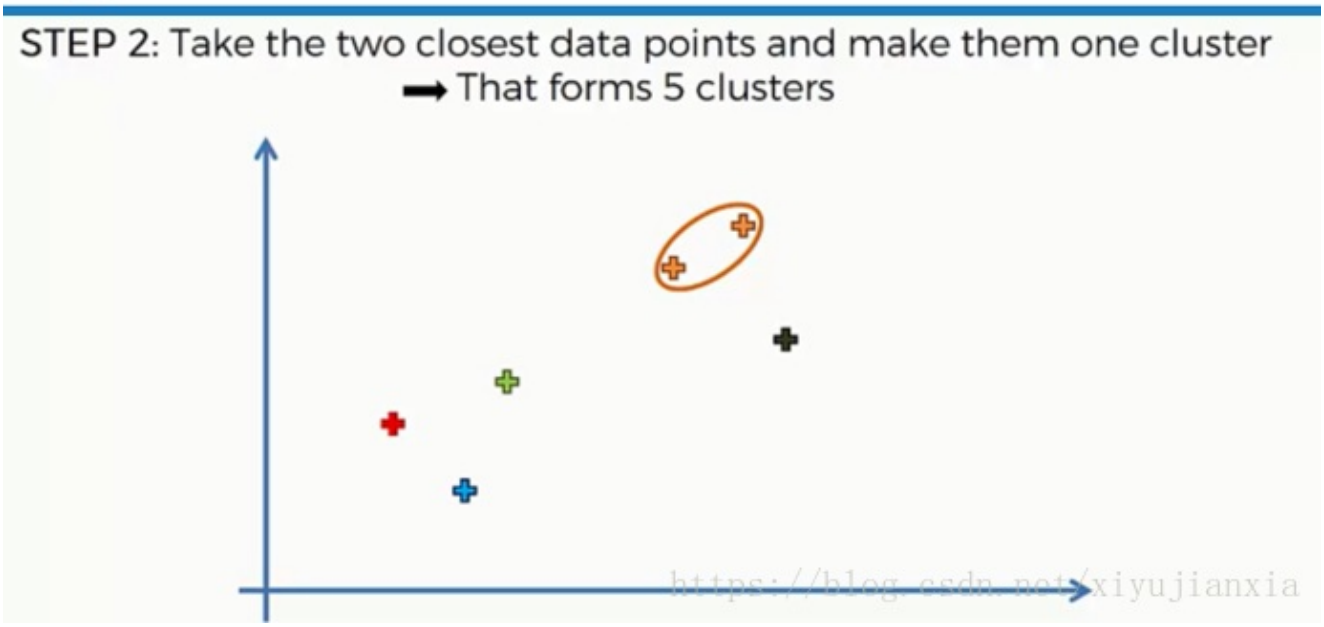
4.1把每一个点当作一个簇（组），例子是6个点



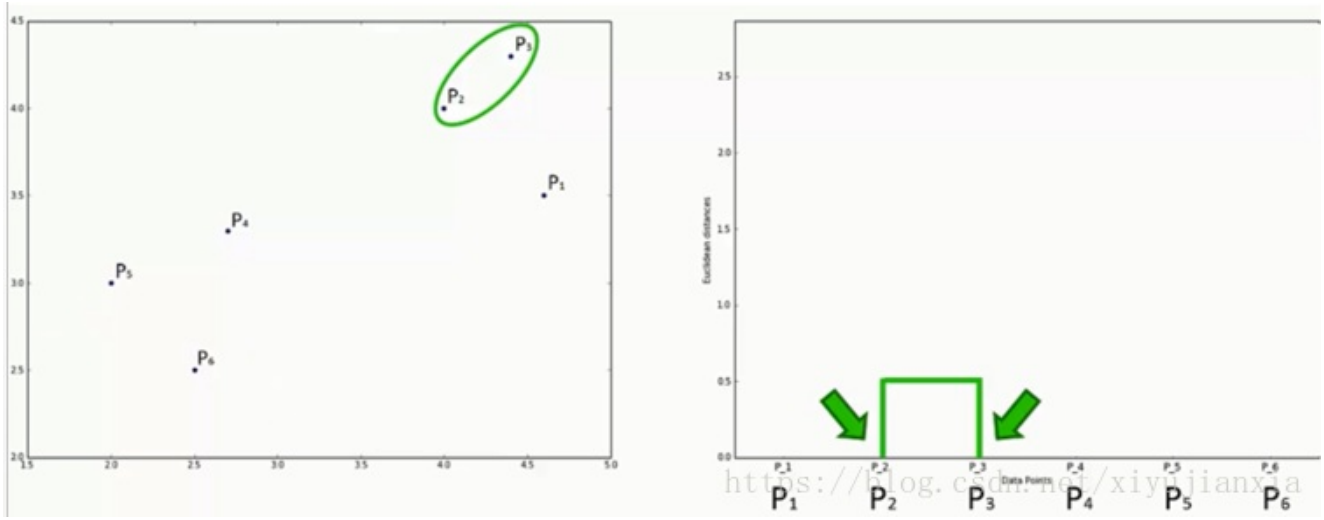

**工作过程：开始在P1-P6共6个点**




4.2 找到两个最近的点合并为一簇（组），就变成5簇

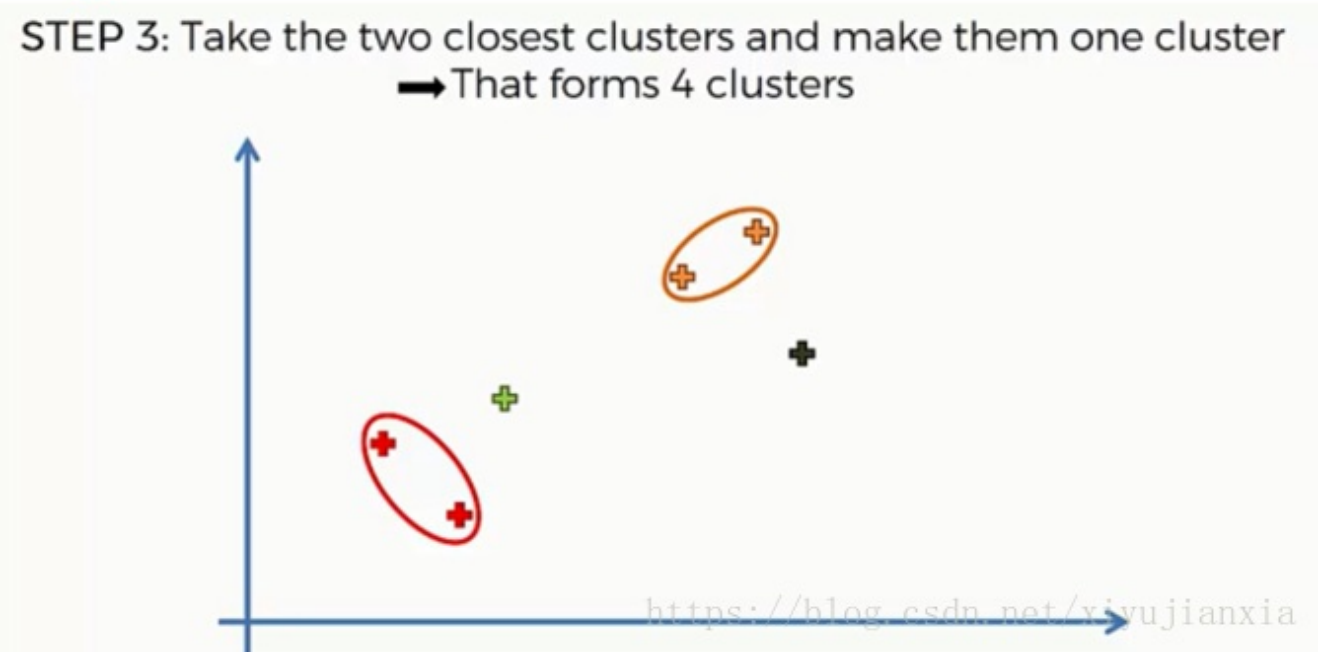


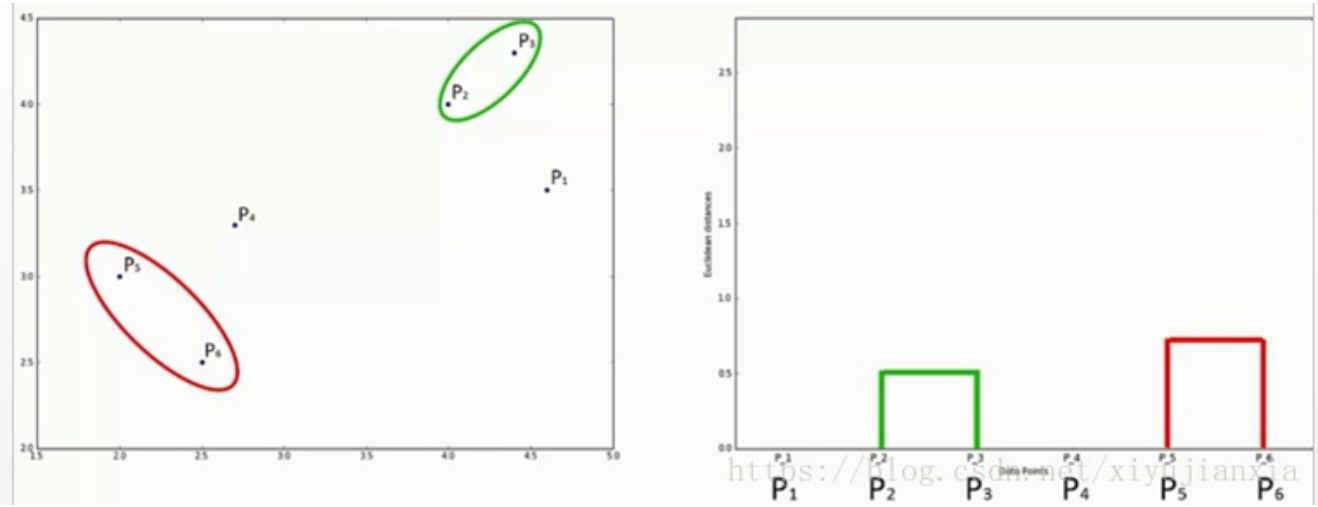

合并最近的两个点，绘制一个矩形，高度代表两个簇之间的距离



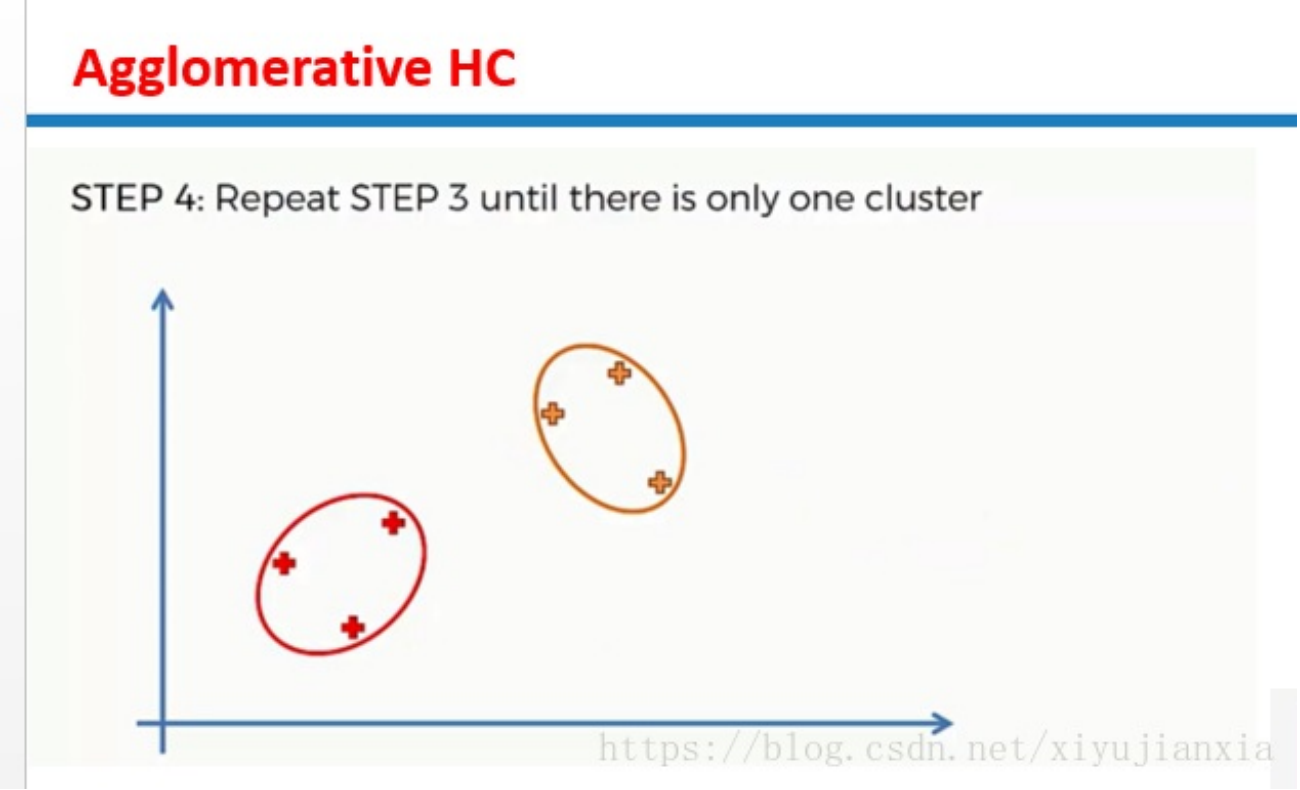

4.3 继续上面的步骤，找到两个最近的点合并为一簇（组），就变成4簇：

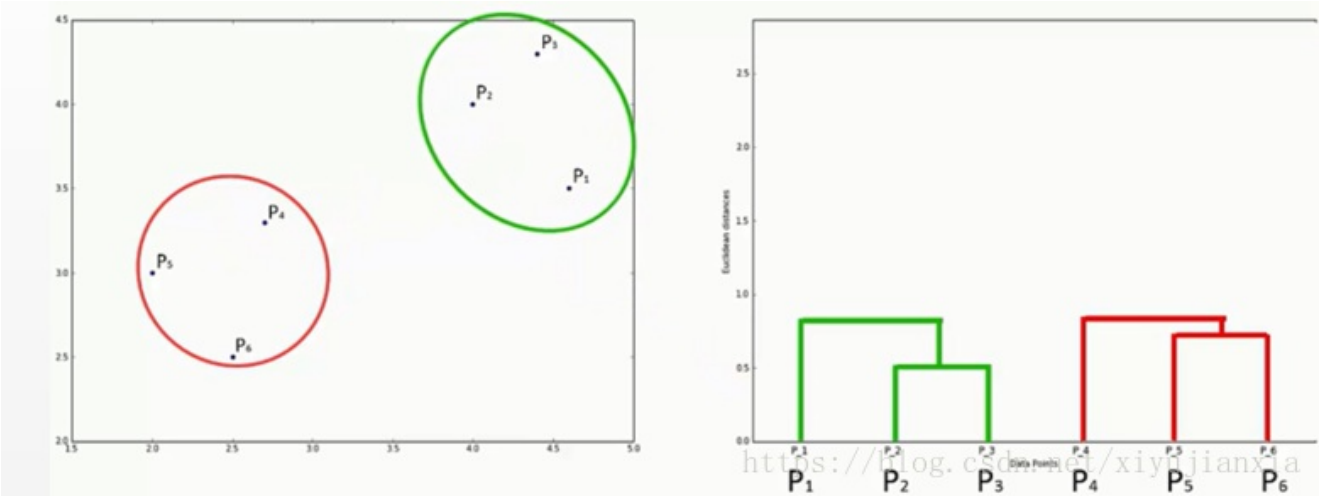
## Agglomerative HC



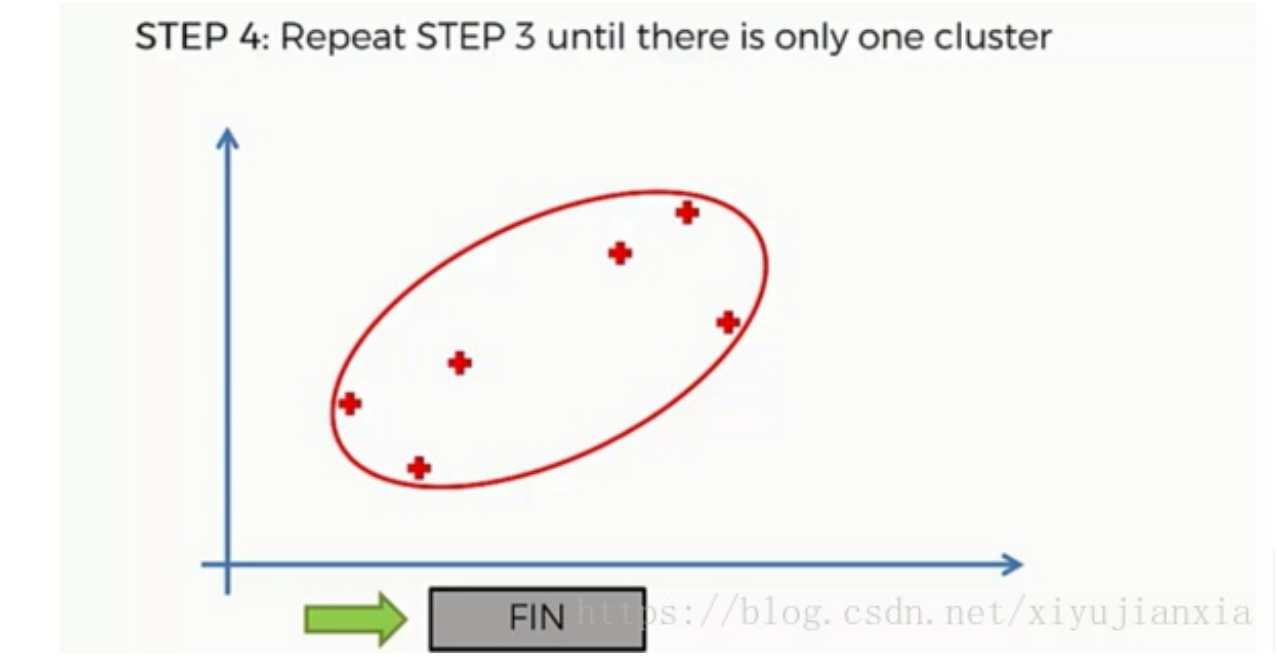



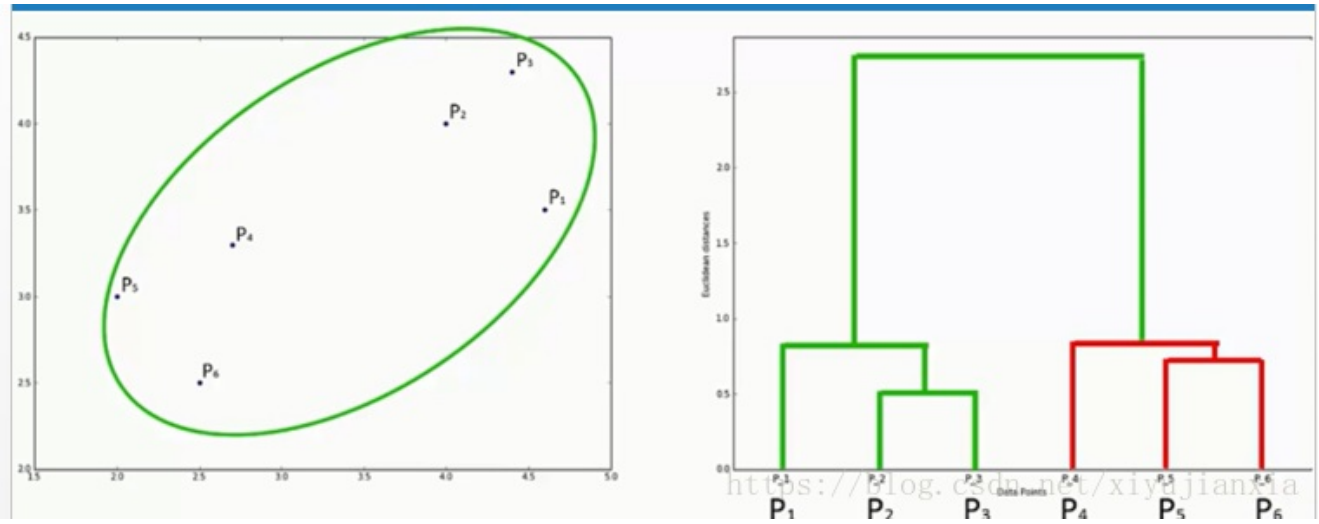

4.4 继续上面的步骤，找到两个最近的点合并为一簇（组），就变成2簇：




4.5 继续上面的步骤，直到变成一个簇（组），结束：




## 5、寻求最优的Clusters的个数

采用什么标准来决定Clusters的个数？ 最大距离，横向切过一条线，所有关联的的簇（组）。

# Optimal numbers of clusters