

人工智能

无监督(聚类)算法

Cluster Analysis

陈川

中山大学 计算机学院

2024年

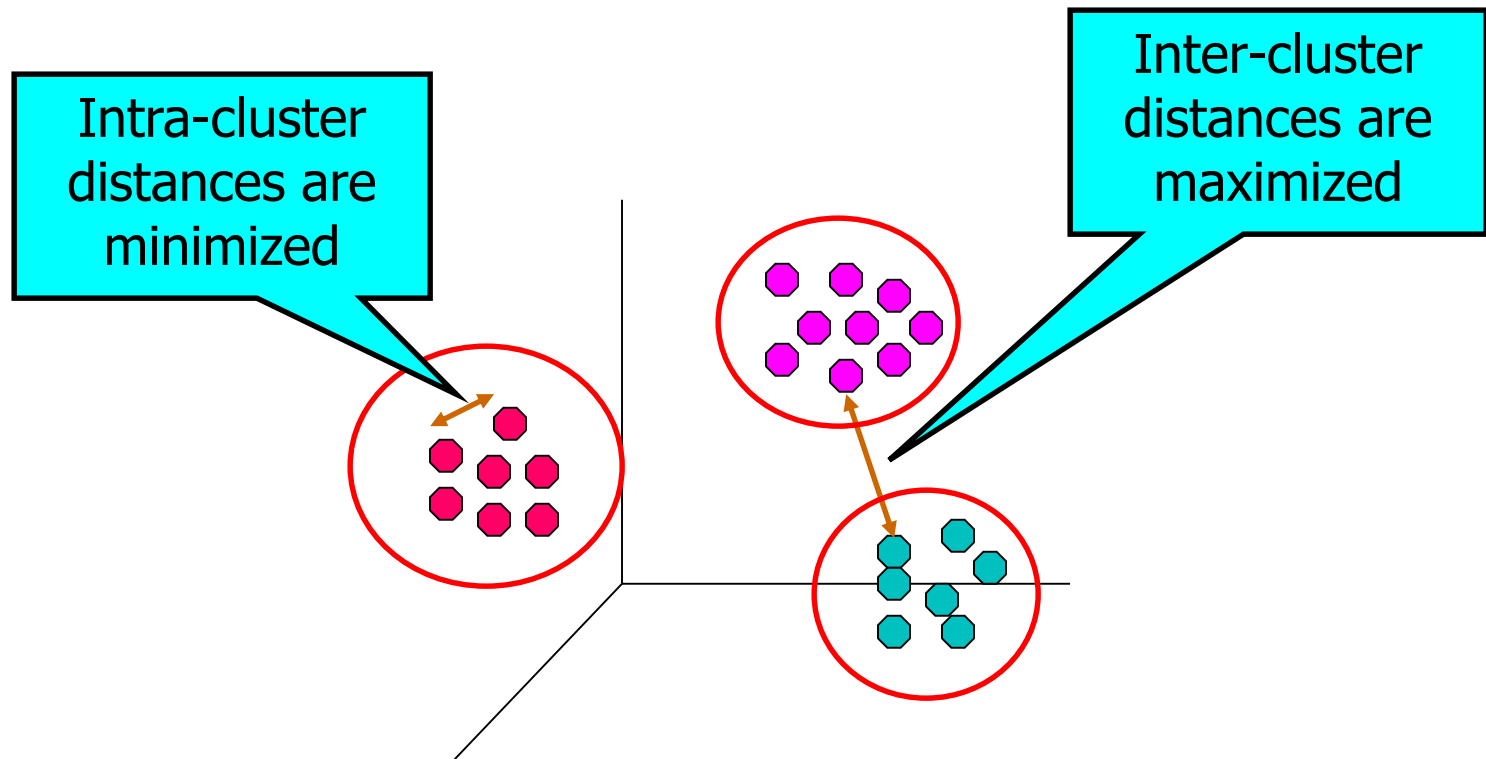


中山大學
SUN YAT-SEN UNIVERSITY

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



Applications of Cluster Analysis



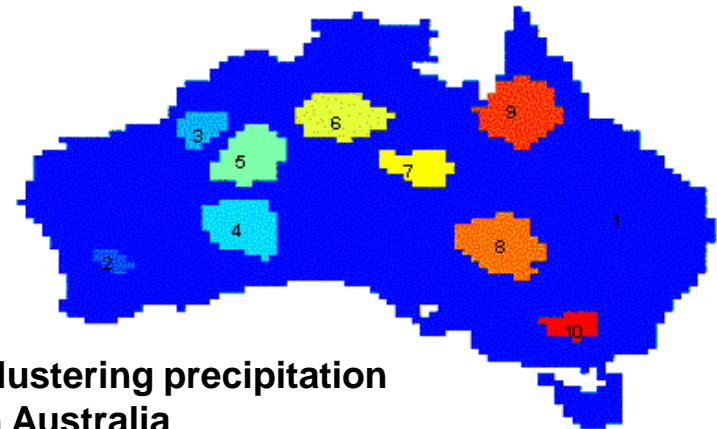
Understanding

- Group related documents for browsing
- Group genes and proteins that have similar functionality
- Group stocks with similar price fluctuations

	<i>Discovered Clusters</i>	<i>Industry Group</i>
1	Applied-Matl-DOWN,Bay-Network-Down,3-COM-DOWN,Cabletron-Sys-DOWN,CISCO-DOWN,HP-DOWN,DSC-Comm-DOWN,INTEL-DOWN,LSI-Logic-DOWN,Micron-Tech-DOWN,Texas-Inst-Down,Tellabs-Inc-Down,Natl-Semiconduct-DOWN,Oracl-DOWN,SGI-DOWN,Sun-DOWN	Technology1-DOWN
2	Apple-Comp-DOWN,Autodesk-DOWN,DEC-DOWN,ADV-Micro-Device-DOWN,Andrew-Corp-DOWN,Computer-Assoc-DOWN,Circuit-City-DOWN,Compaq-DOWN,EMC-Corp-DOWN,Gen-Inst-DOWN,Motorola-DOWN,Microsoft-DOWN,Scientific-Atl-DOWN	Technology2-DOWN
3	Fannie-Mae-DOWN,Fed-Home-Loan-DOWN,MBNA-Corp-DOWN,Morgan-Stanley-DOWN	Financial-DOWN
4	Baker-Hughes-UP,Dresser-Inds-UP,Halliburton-HLD-UP,Louisiana-Land-UP,Phillips-Petro-UP,Unocal-UP,Schlumberger-UP	Oil-UP

Summarization

- Reduce the size of large data sets

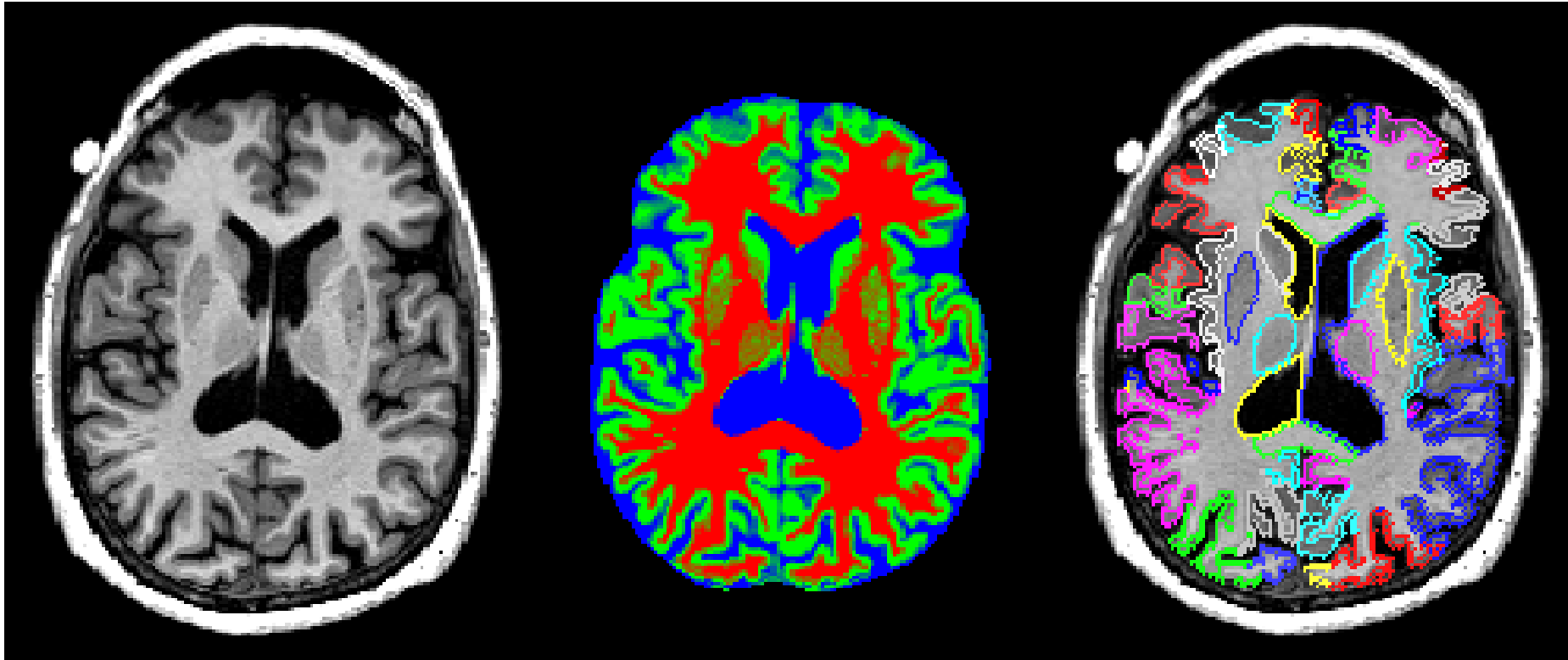


Clustering precipitation
in Australia

Clustering in Social Network



Clustering in Image Segmentation



Clustering in Video Detection



TVCP



Frame #50

F-m.=0.9630



Frame #70

F-m.=0.9752

Nonnegative PARAFAC



Frame #50

F-m.=0.9209



Frame #70

F-m.=0.9512

What is not Cluster Analysis?



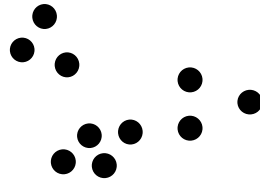
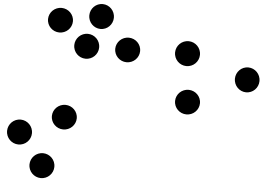
Supervised classification

- Have class label information

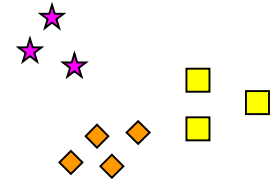
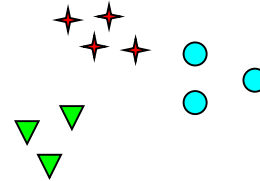
Simple segmentation

- Dividing students into different registration groups alphabetically, by last name

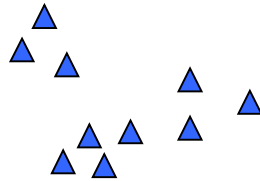
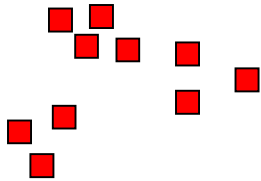
Notion of a Cluster can be Ambiguous



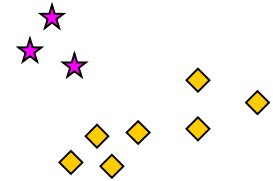
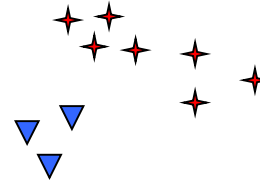
How many clusters?



Six Clusters



Two Clusters

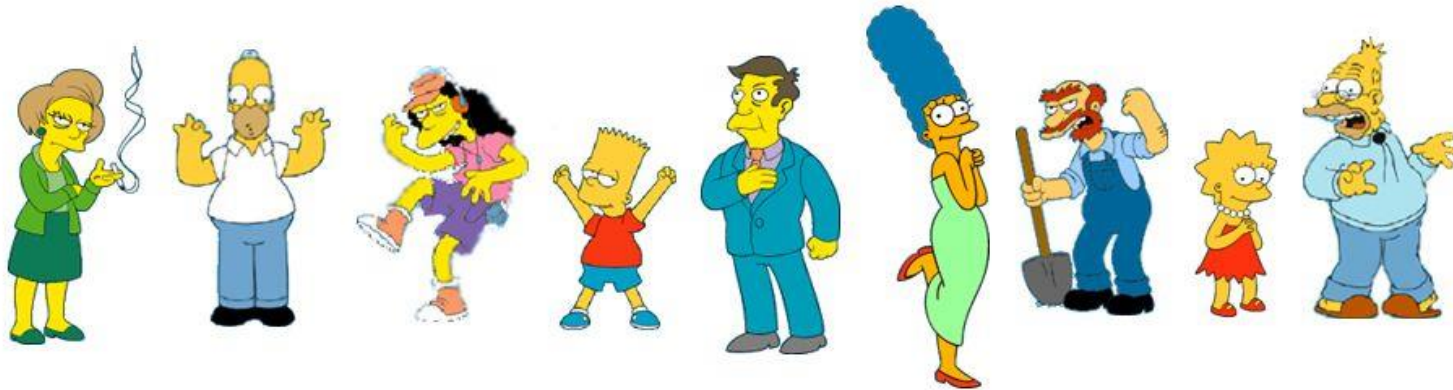


Four Clusters

Notion of a Cluster can be Ambiguous



What is a natural grouping among these objects?



Types of Clusterings



A **clustering** is a set of clusters

Important distinction between **hierarchical** and **partitional** sets of clusters

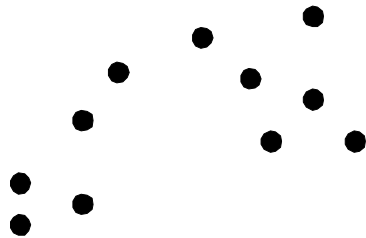
Partitional Clustering (分割式聚类)

- A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- # clusters is needed, kmeans....

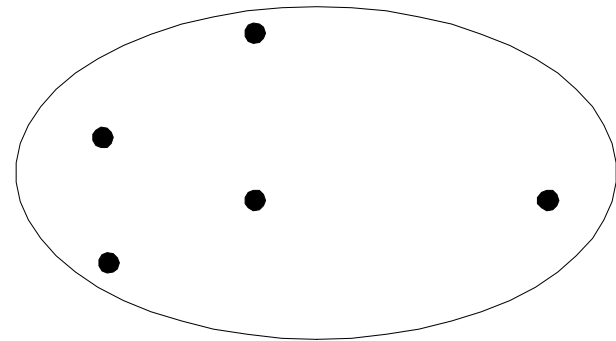
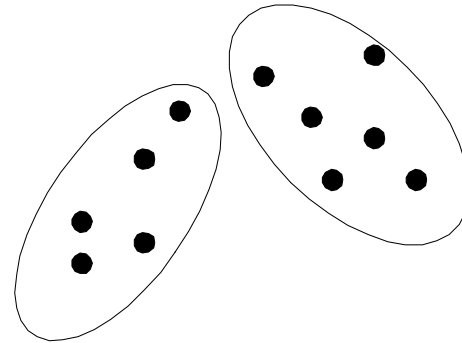
Hierarchical clustering (阶层式聚类)

- A set of nested clusters organized as a hierarchical tree
- # clusters is not needed

Partitional Clustering

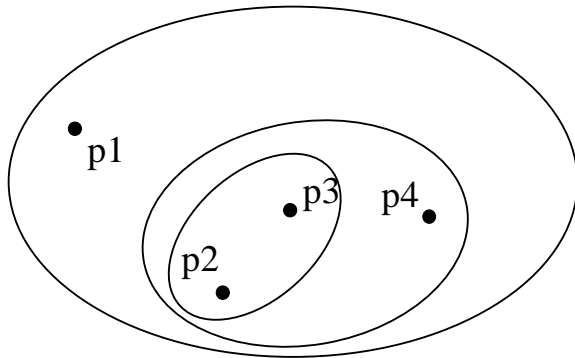


Original Points

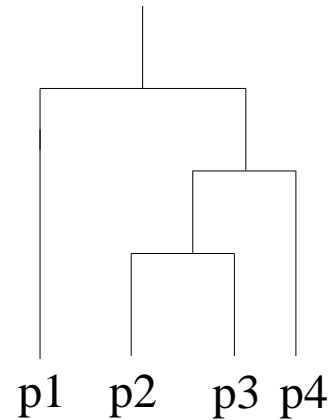


A Partitional Clustering

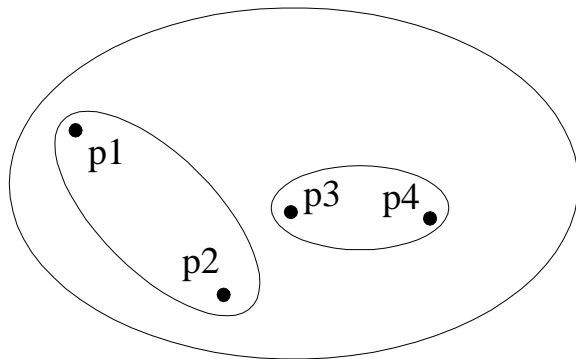
Hierarchical Clustering



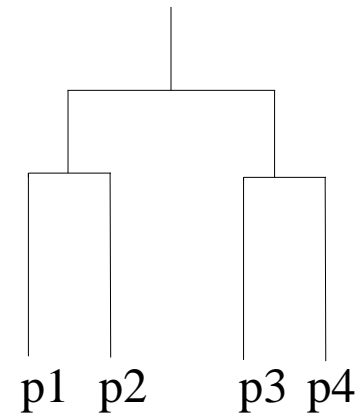
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

Other Distinctions Between Sets of Clusters



Fuzzy versus non-fuzzy 模糊聚类

- In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
- Weights must sum to 1
- Probabilistic clustering has similar characteristics

Partial versus complete

- In some cases, we only want to cluster some of the data

Heterogeneous versus homogeneous 异质vs同质

- Cluster of widely different sizes, shapes, and densities

Types of Clusters



Well-separated clusters

Center-based clusters

Contiguous clusters

Density-based clusters

Property or Conceptual

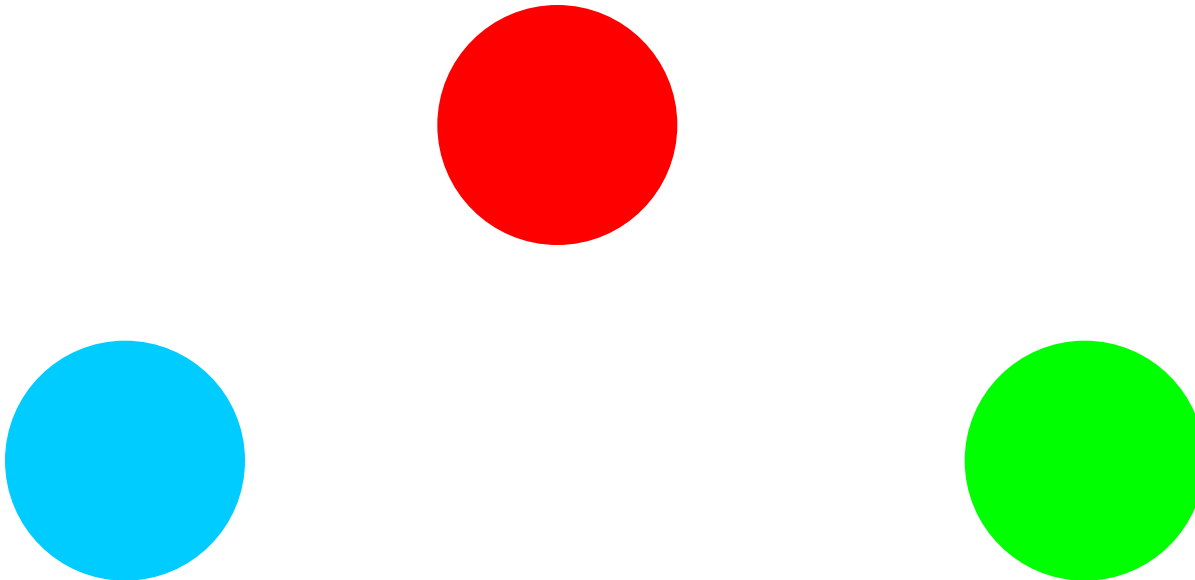
Described by an Objective Function

Types of Clusters: Well-Separated



Well-Separated Clusters:

- A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.



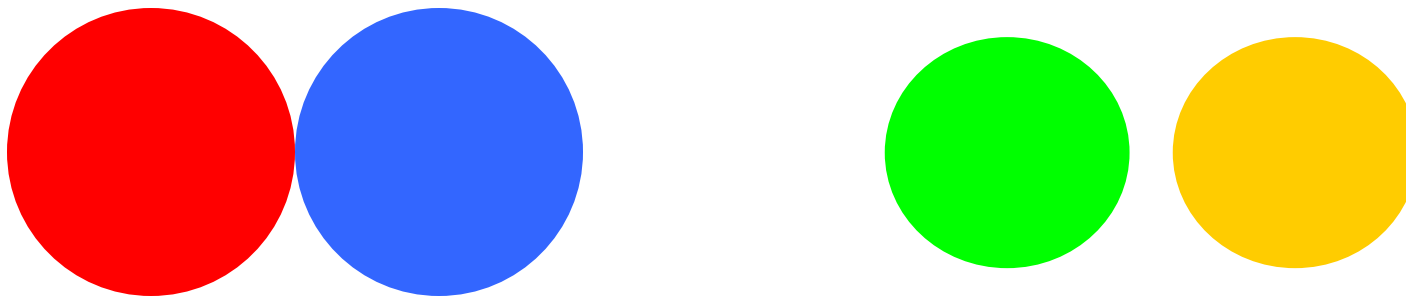
3 well-separated clusters

Types of Clusters: Center-Based



Center-based

- A cluster is a set of objects such that an object in a cluster is closer (more similar) to the “center” of a cluster, than to the center of any other cluster
- The center of a cluster is often a **centroid**, the average of all the points in the cluster, or a **medoid**, the most “representative” point of a cluster



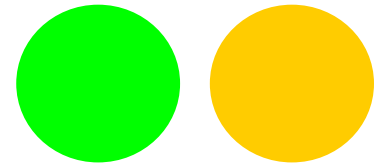
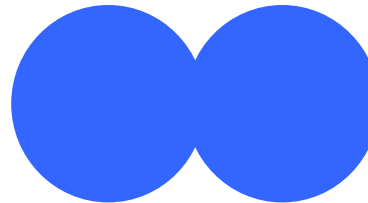
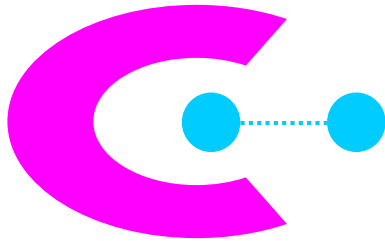
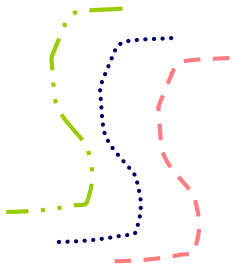
4 center-based clusters

Types of Clusters: Contiguity(邻近)-Based



Contiguous Cluster (Nearest neighbor or Transitive)

- A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.



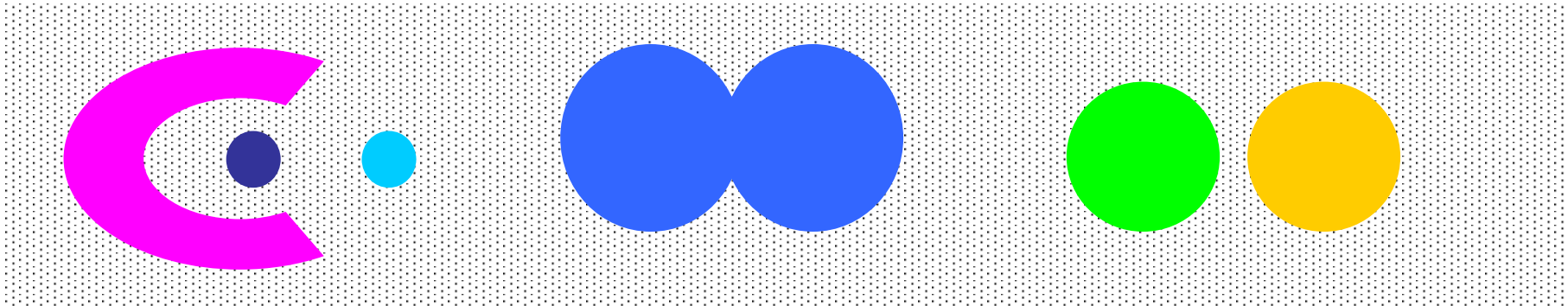
8 contiguous clusters

Types of Clusters: Density-Based



Density-based

- A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
- Used when the clusters are irregular or intertwined(不规则 or 纠缠), and when noise and outliers are present.



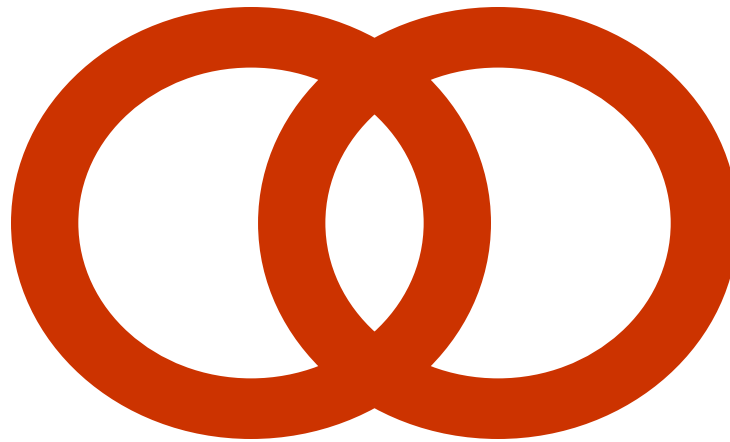
6 density-based clusters

Types of Clusters: Conceptual(概念) Clusters



Shared Property or Conceptual Clusters

- Finds clusters that share some common property or represent a particular concept.



2 Overlapping Circles

Types of Clusters: Objective Function



Clusters Defined by an Objective Function

- Finds clusters that minimize or maximize an objective function.
- **穷举法:** Enumerate all possible ways of dividing the points into clusters and evaluate the 'goodness' of each potential set of clusters by using the given objective function. (NP Hard)
- Can have global or local objectives.
 - ◆ Hierarchical clustering algorithms typically have local objectives
 - ◆ Partitional algorithms typically have global objectives

Types of Clusters: Objective Function ...



Map the clustering problem to a different domain and solve a related problem in that domain

- Proximity (亲近度) matrix defines a weighted graph, where the nodes are the data points, and the weighted edges represent the proximities (similarity or dissimilarity) between data points

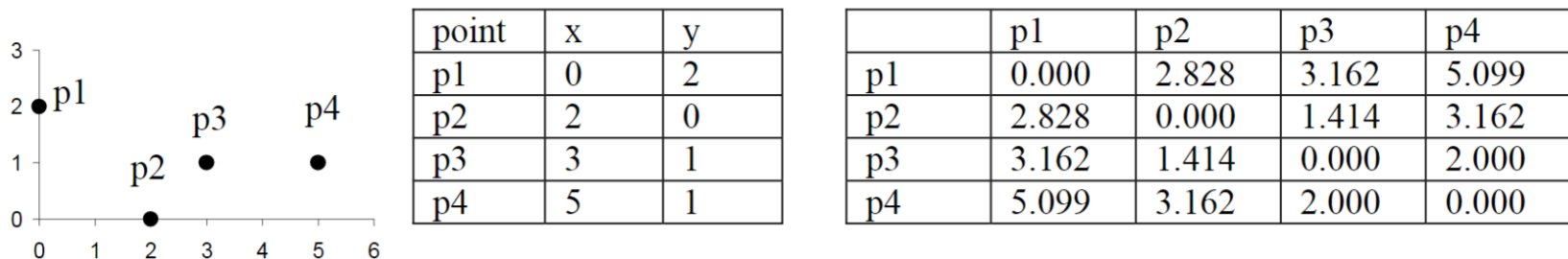


Figure Four points and their corresponding data and proximity (distance) matrices.

- Similarity? Dissimilarity?

Euclidean Distance / Manhattan Distance / Chebyshev Distance /

Cosine Similarity /

$$\text{Similarity} = 1 - \text{Dissimilarity}$$

Bonus: Some Distance Metrics



$$L_p(x_1, x_2) = \left(\sum_{l=1}^n |x_i^{(l)} - x_j^{(l)}|^p \right)^{\frac{1}{p}}$$

$$L_1(x_1, x_2) = \sum_{l=1}^n |x_i^{(l)} - x_j^{(l)}| \quad \text{Manhattan Distance}$$

$$L_2(x_1, x_2) = \sum_{l=1}^n \sqrt{(x_i^{(l)} - x_j^{(l)})^2} \quad \text{Euclidean Distance}$$

$$L_\infty(x_1, x_2) = \sqrt[n]{\sum_{l=1}^n |x_i^{(l)} - x_j^{(l)}|^\infty} = \max(|x_i - x_j|)$$

Chebyshev Distance

$$L_{-\infty}(x_1, x_2) = \sqrt[n]{\sum_{l=1}^n |x_i^{(l)} - x_j^{(l)}|^{-\infty}} = \min(|x_i - x_j|)$$

$$\text{Cosine Distance: } D(x_1, x_2) = 1 - \cos(x_1, x_2) = 1 - \frac{x_1 \cdot x_2}{\|x_1\|_2 \|x_2\|_2}$$

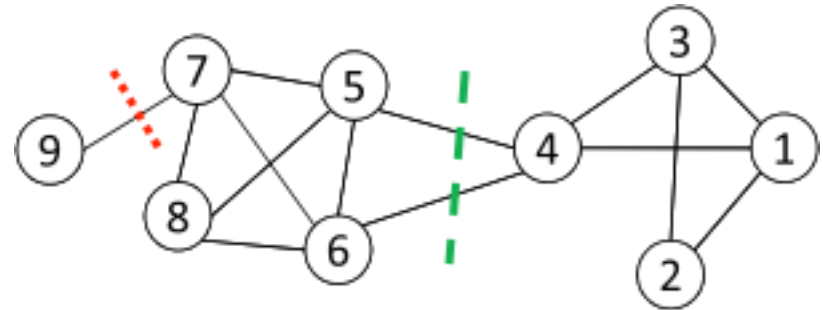
Types of Clusters: Objective Function ...



- Clustering is equivalent to breaking the graph into connected components, one for each cluster.
- Want to minimize the edge weight between clusters and maximize the edge weight within clusters

$$\text{Ratio Cut}(\pi) = \frac{1}{k} \sum_{i=1}^k \frac{\text{cut}(C_i, \bar{C}_i)}{|C_i|},$$

$$\text{Normalized Cut}(\pi) = \frac{1}{k} \sum_{i=1}^k \frac{\text{cut}(C_i, \bar{C}_i)}{\text{vol}(C_i)}$$



C_i : a community
 $|C_i|$: # nodes in C_i
 $\text{vol}(C_i)$: sum of degrees in C_i

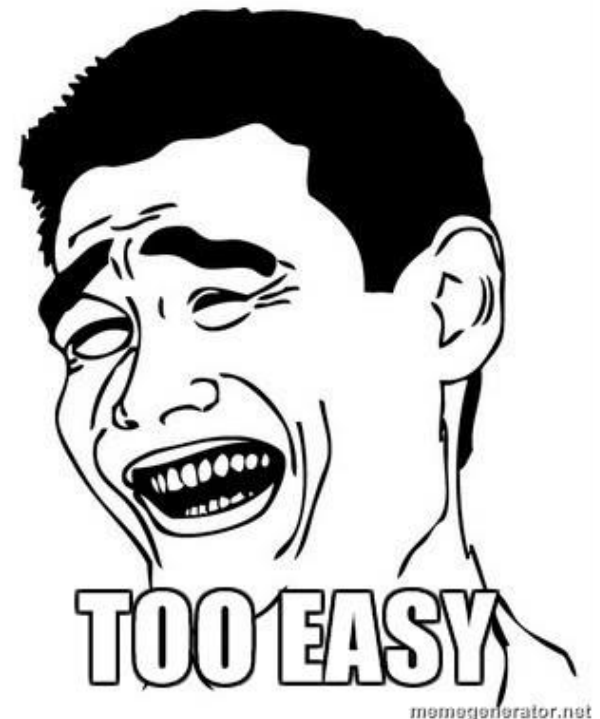
Clustering Algorithms



Hierarchical clustering (Omit, because...)

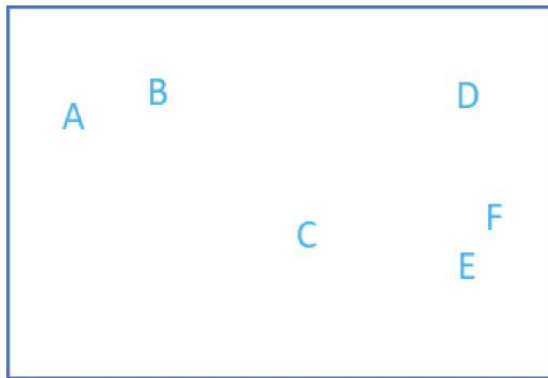
K-means and its variants

Density-based clustering

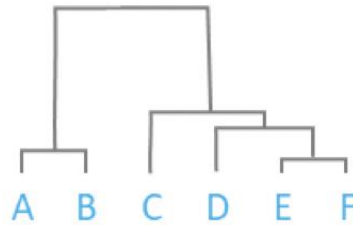


Clustering Algorithms

Hierarchical clustering



Dendrogram



Hierarchical clustering starts by treating each observation as a separate cluster.

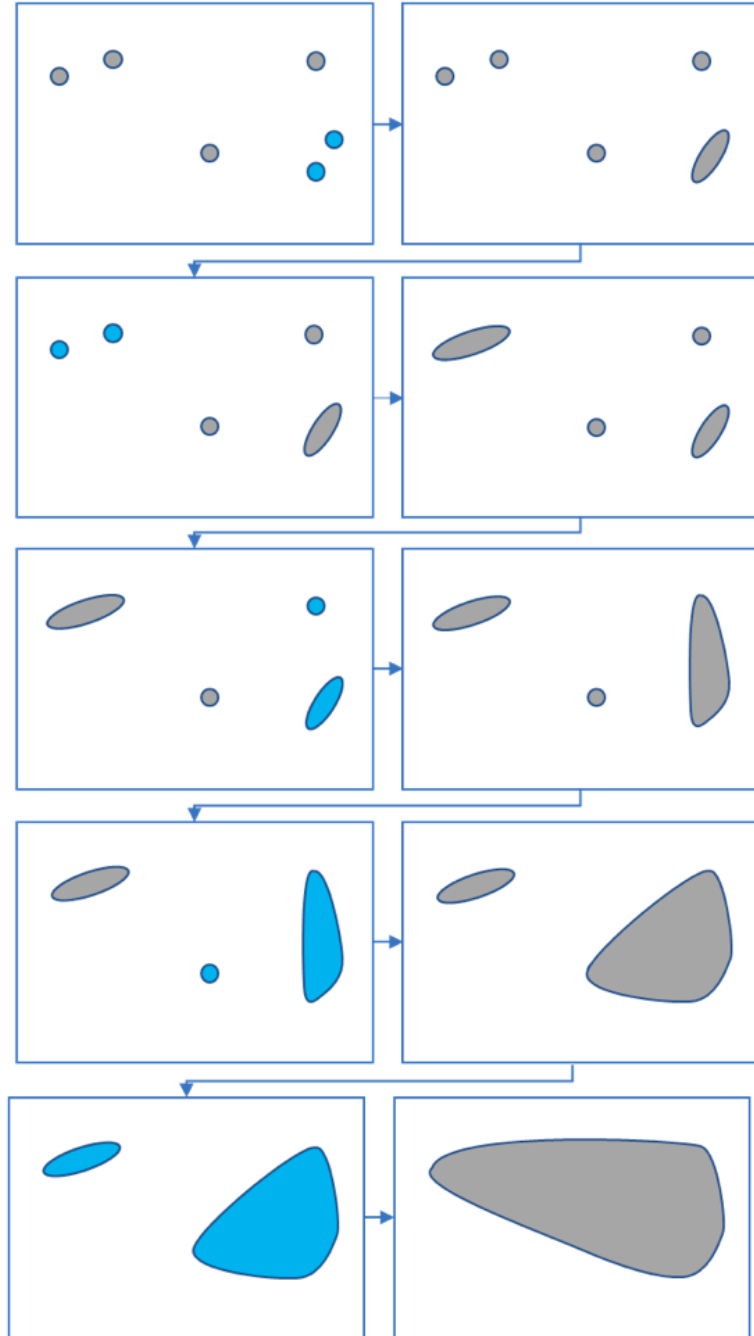
Then, it repeatedly executes the following two steps:

- (1) identify the two clusters that are closest together,
- (2) merge the two most similar clusters.

This iterative process continues until all the clusters are merged together.

Identify the two clusters that are **closest** together

Merge the two most similar clusters



K-means Clustering



Partitional clustering approach

- Each cluster is associated with a **centroid** (center point)
- Each point is assigned to the cluster with the **closest centroid**

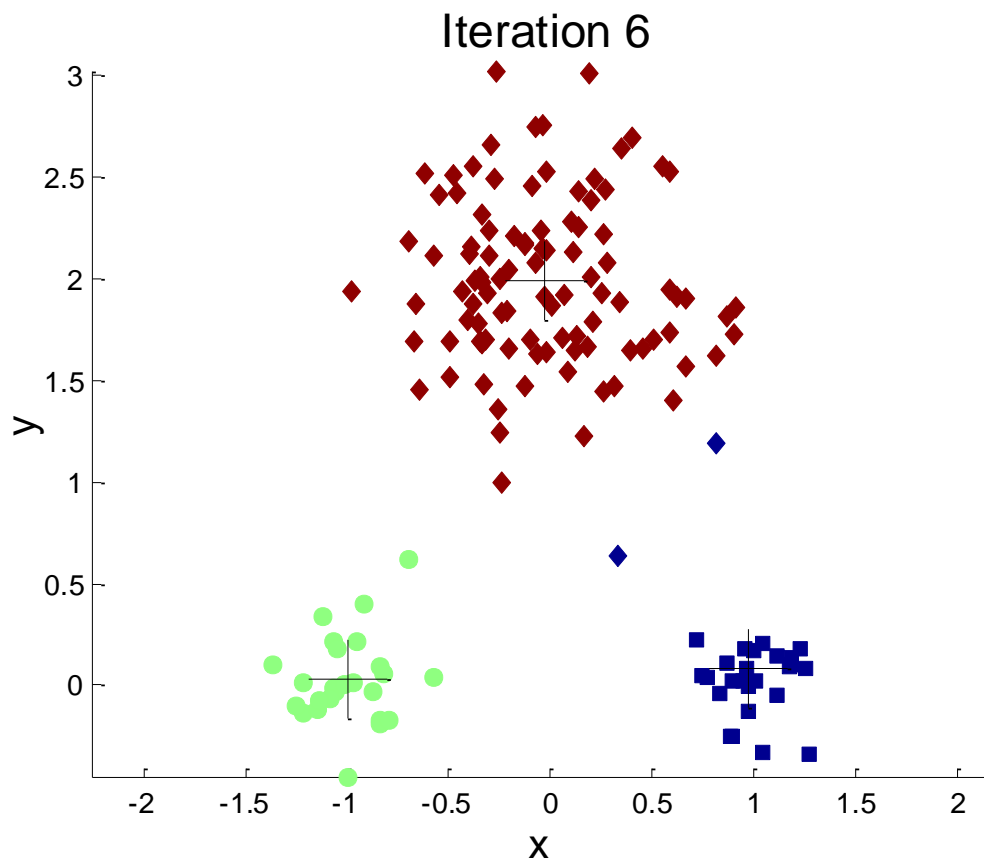
Number of clusters, K , must be specified

The basic algorithm is very simple

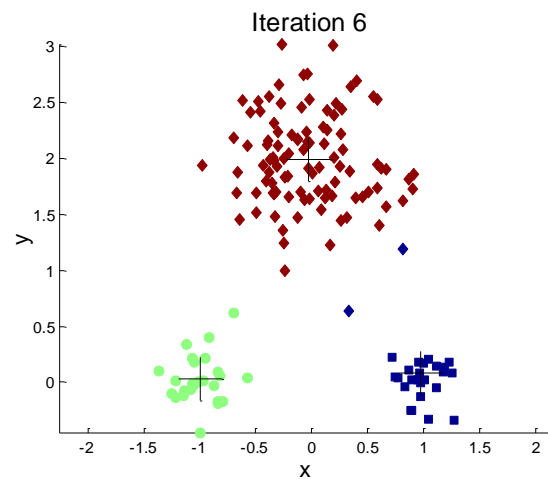
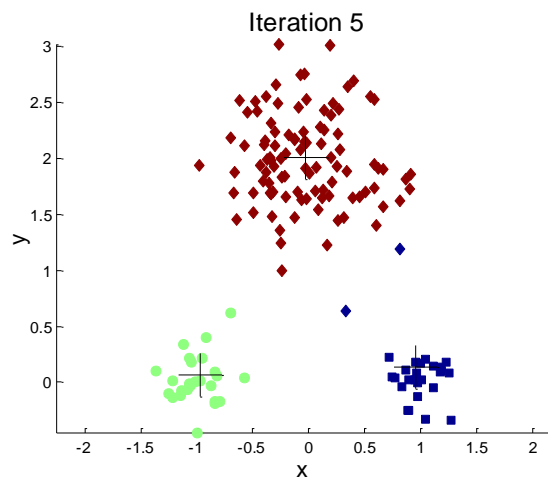
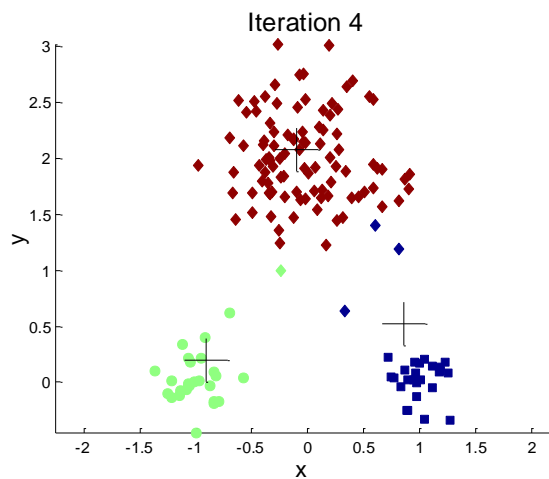
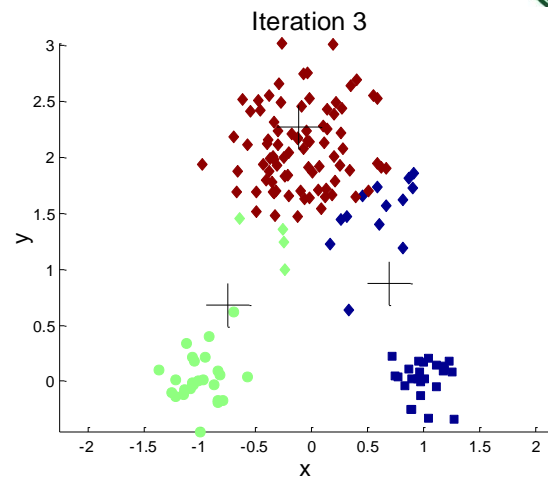
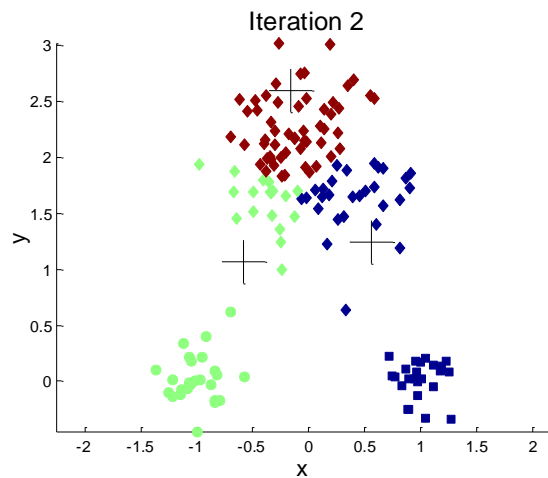
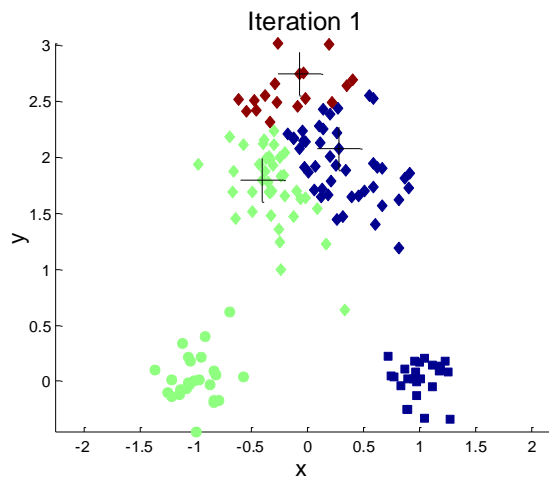
-
- 1: Select K points as the initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning all points to the closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** The centroids don't change
-

<https://www.naftaliharris.com/blog/visualizing-k-means-clustering/>

标准Kmeans流程



标准Kmeans流程



K-means Clustering – Details



Initial centroids are often chosen randomly.

- Clusters produced vary from one run to another.

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

‘Closeness’ is measured by Euclidean distance, cosine similarity, correlation, etc.

K-means will converge for common similarity measures mentioned above.

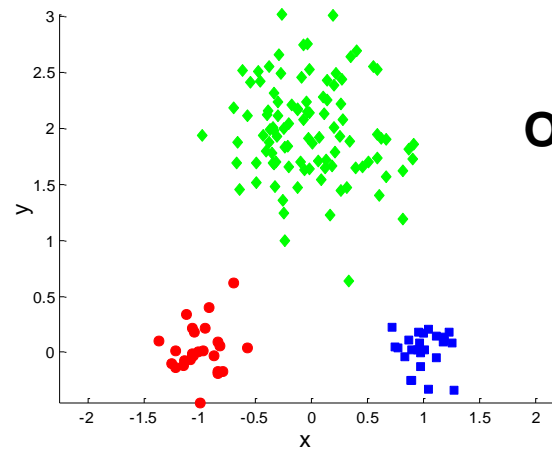
Most of the convergence happens in the first few iterations.

- Often the stopping condition is changed to ‘Until relatively few points change clusters’

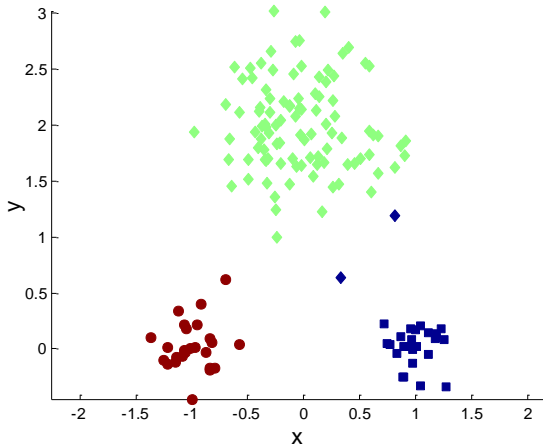
Complexity is $O(n * K * I * d)$

- n = number of points, K = number of clusters,
 I = number of iterations, d = number of attributes

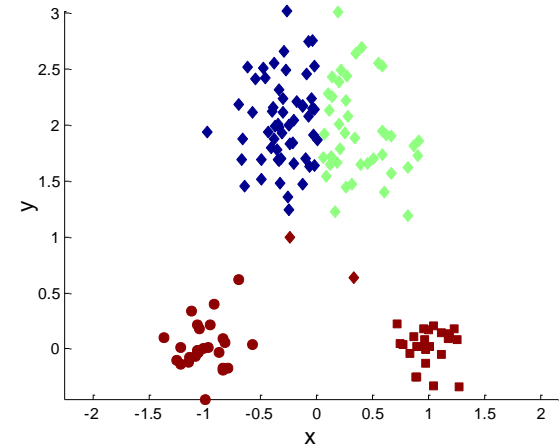
Two different K-means Clusterings



Original Points

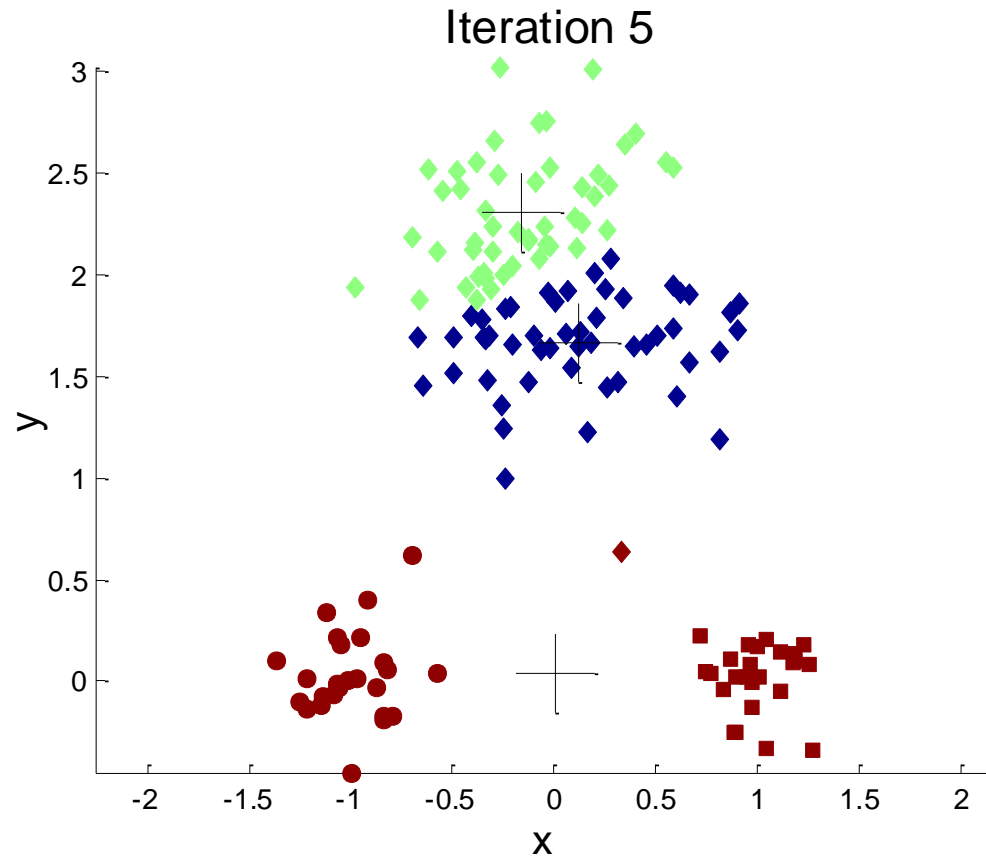


Optimal Clustering

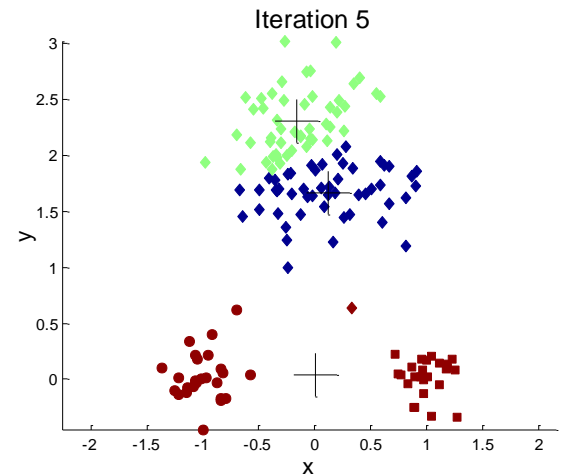
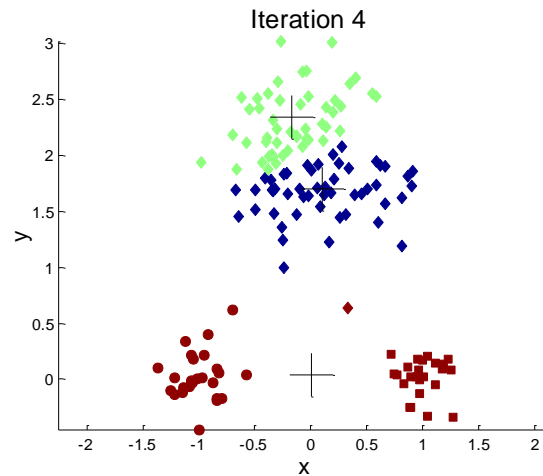
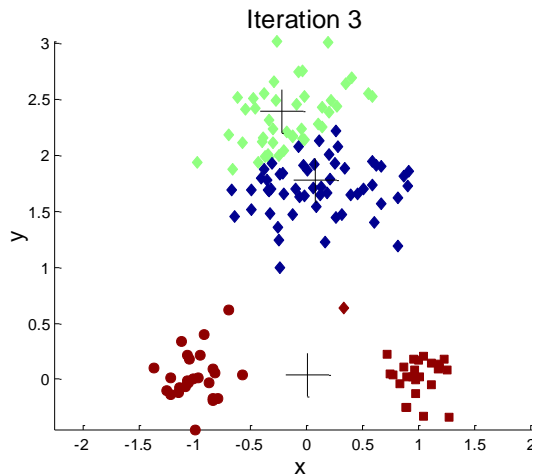
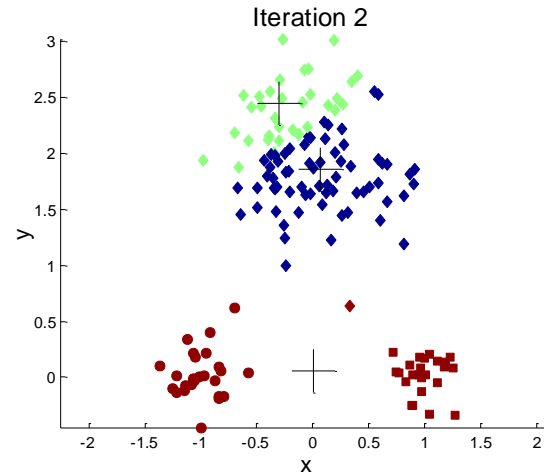
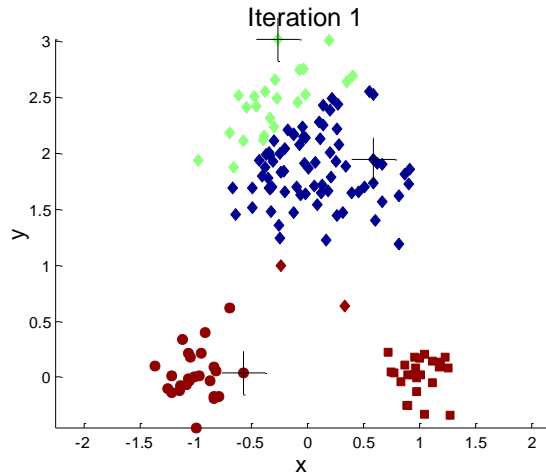


Sub-optimal Clustering

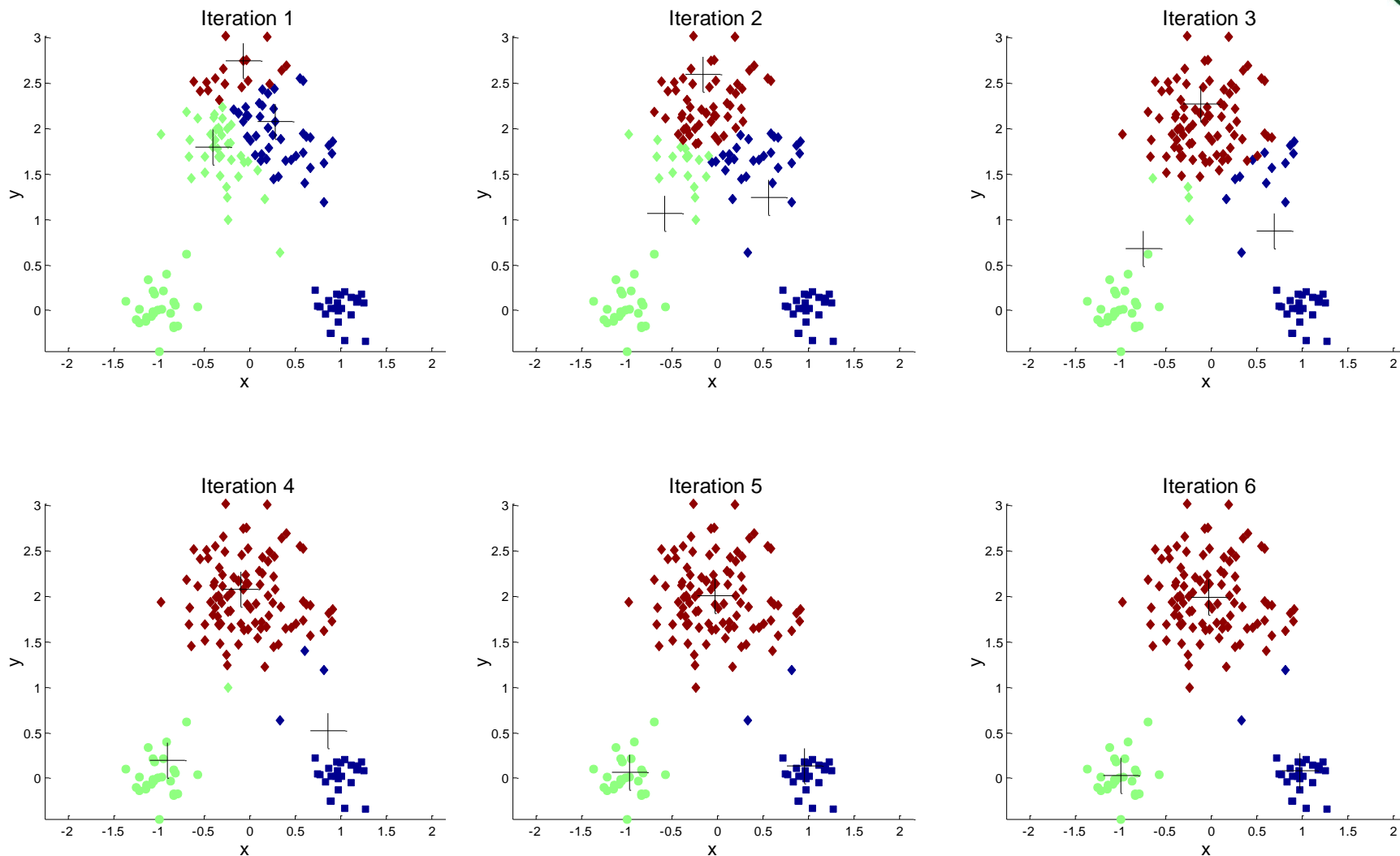
Importance of Choosing Initial Centroids ...



Importance of Choosing Initial Centroids ...



Importance of Choosing Initial Centroids



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.

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

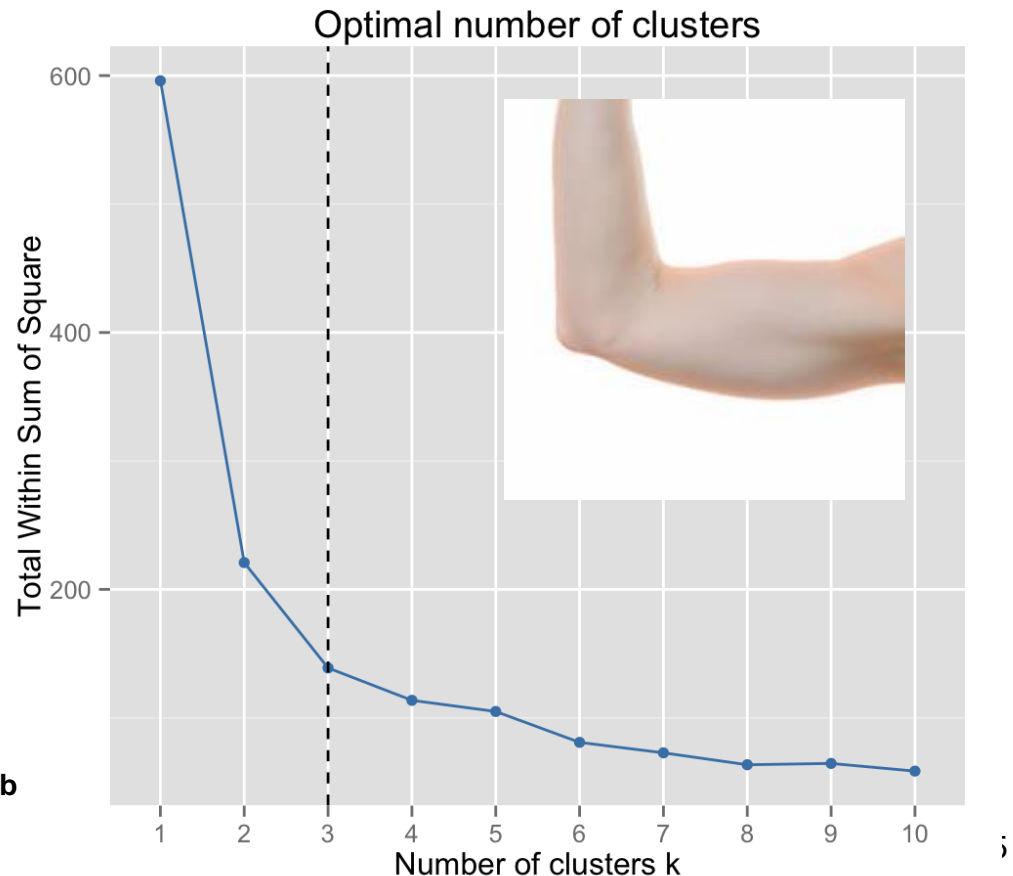
- x is a data point in cluster C_i and m_i is the representative point for cluster C_i
 - ◆ can show that m_i corresponds to the center (mean) of the cluster
- Given two clusters, we can choose the one with the smallest error
- One easy way to reduce SSE is to increase K , the number of clusters
 - ◆ A good clustering with smaller K can have a lower SSE than a poor clustering with higher K

Bonus: how to choose a lovely "K"



- **Elbow method:** plot a line chart of the SSE for each value of k . If the line chart looks like an arm, then the "elbow" on the arm is the value of K that is the best.

手臂曲线的肘关节



<https://bl.ocks.org/rpgove/0060ff3b656618e9136b>

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

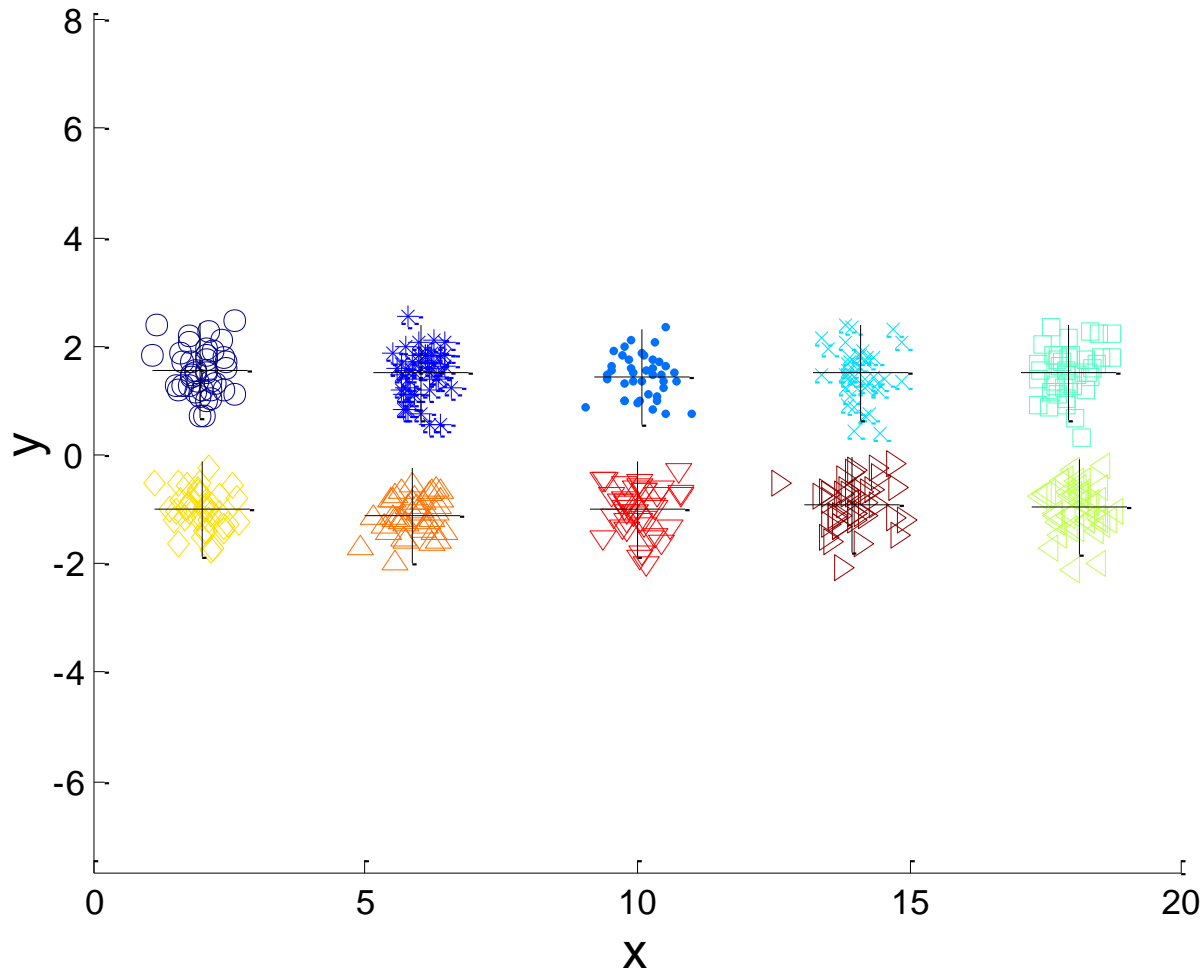
$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K}$$

- 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

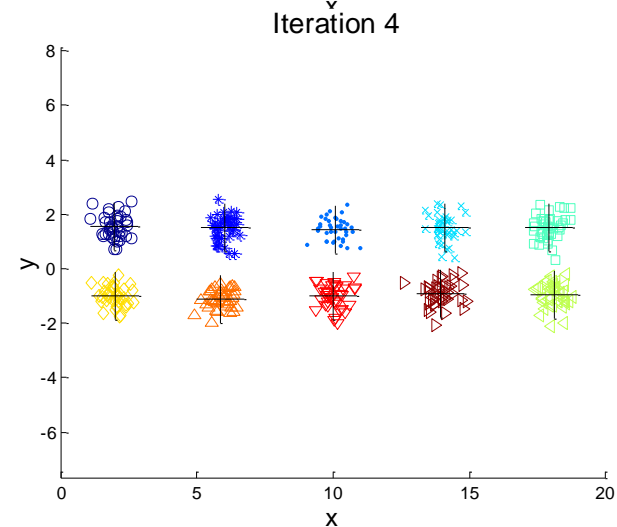
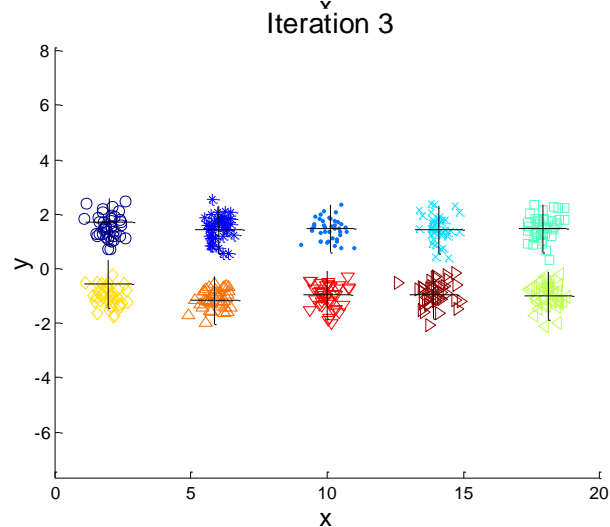
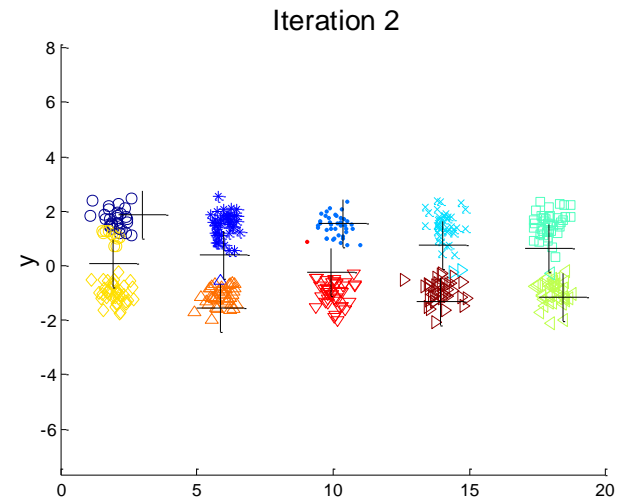
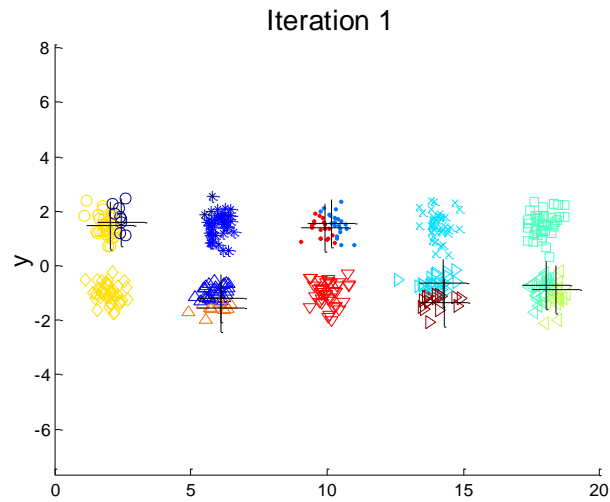


Iteration 4



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

10 Clusters Example

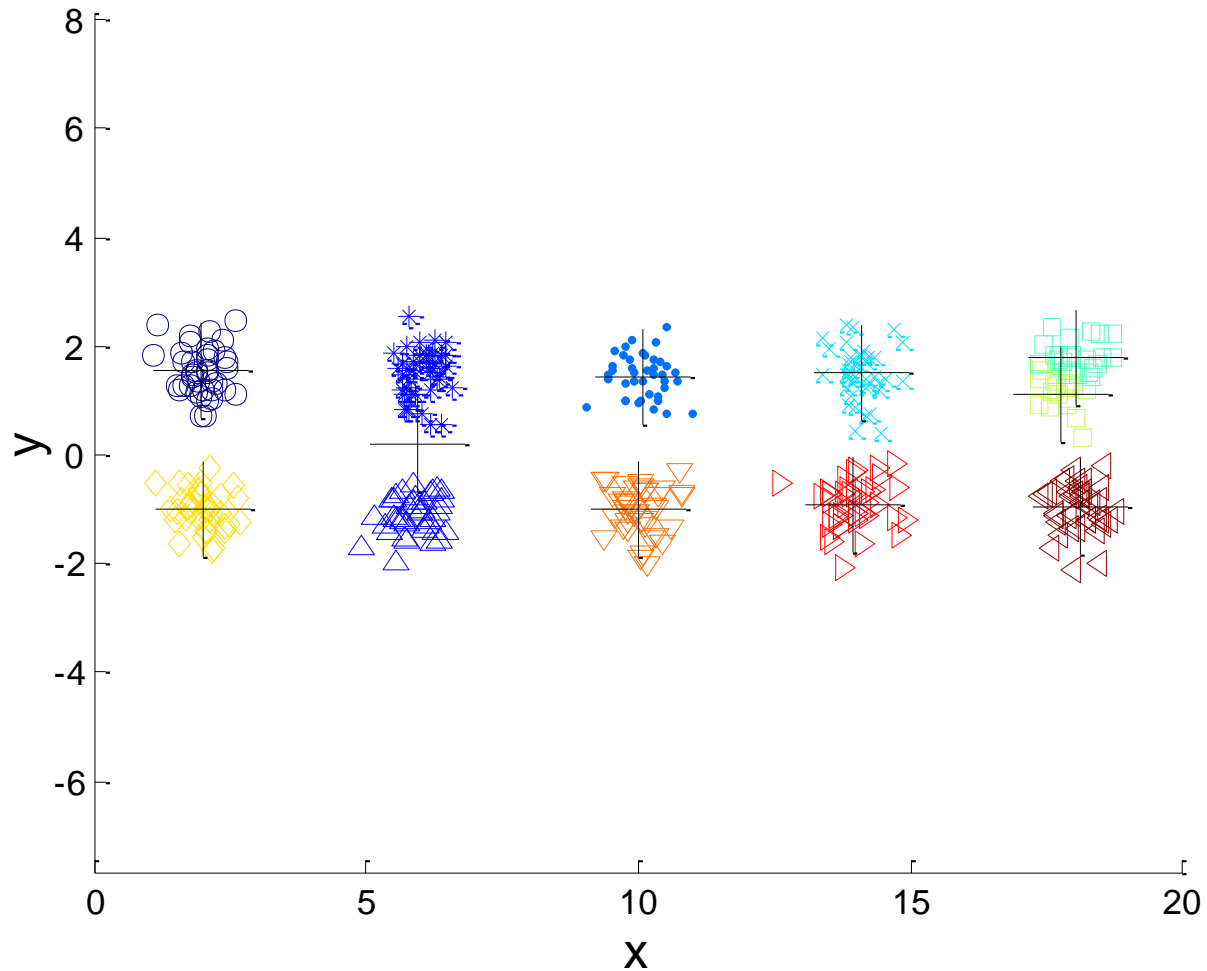


Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example

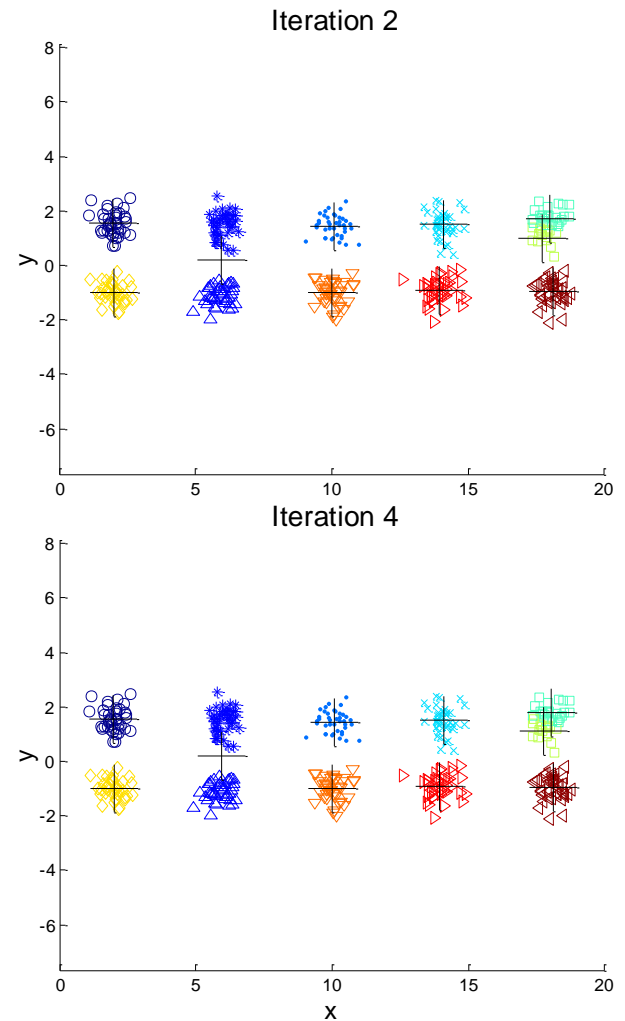
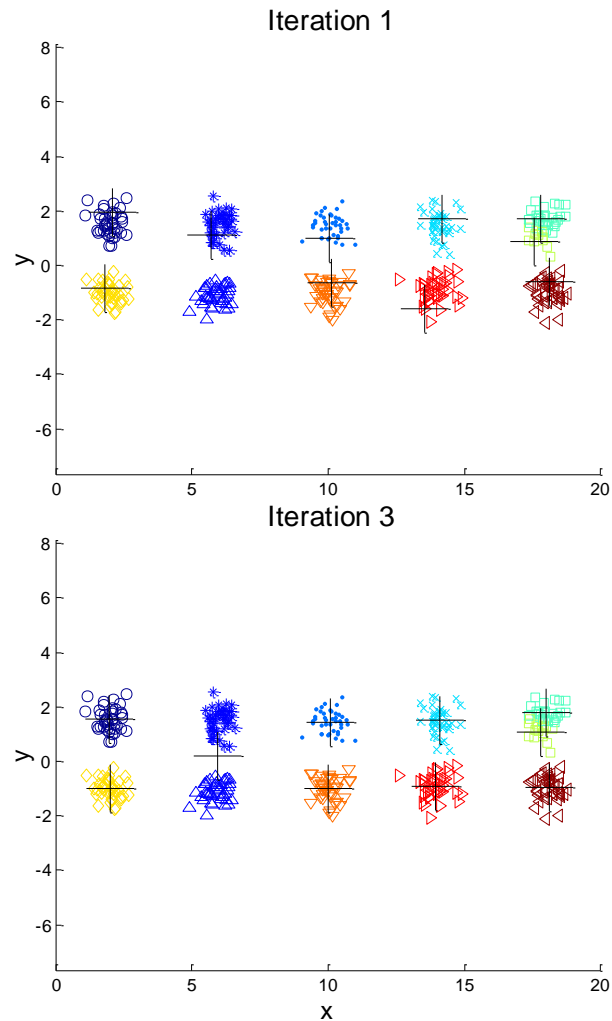


Iteration 4



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 (二分kmeans)

- 不容易受到初始化问题的影响

Bisecting K-means algorithm

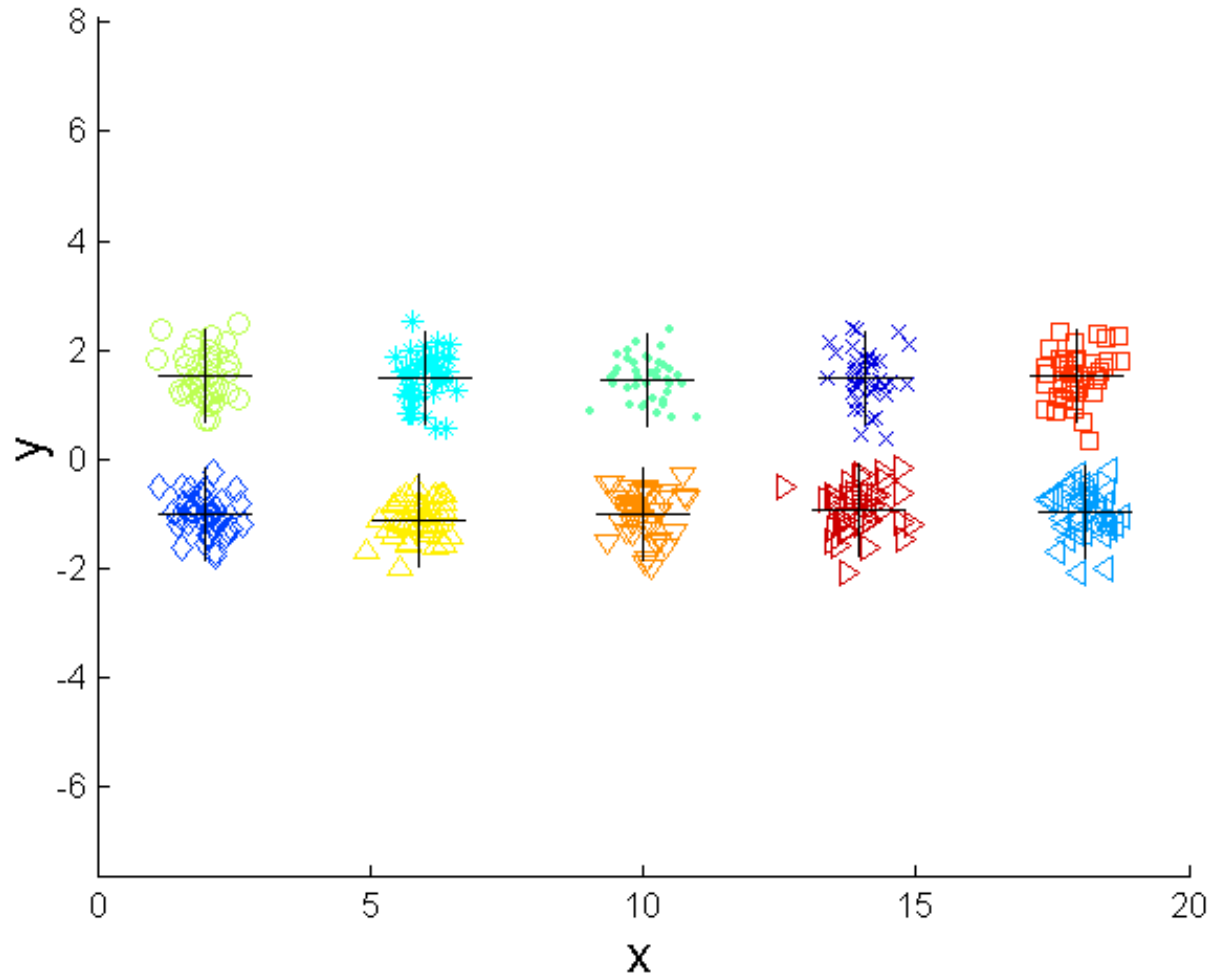
- Variant of K-means that can produce a partitional or a hierarchical clustering

```
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
```

Bisecting K-means Example



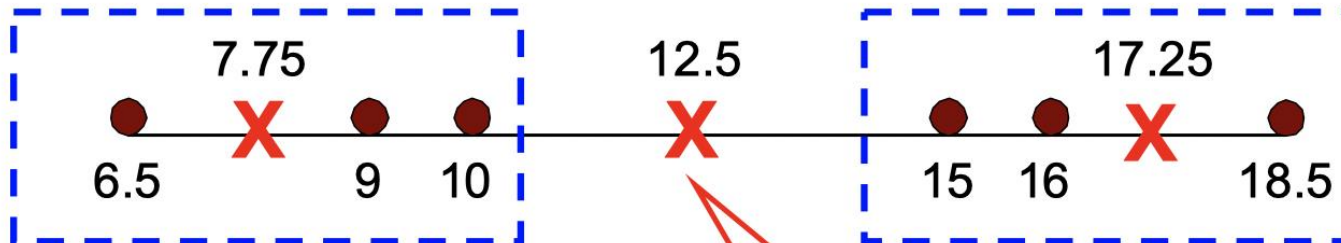
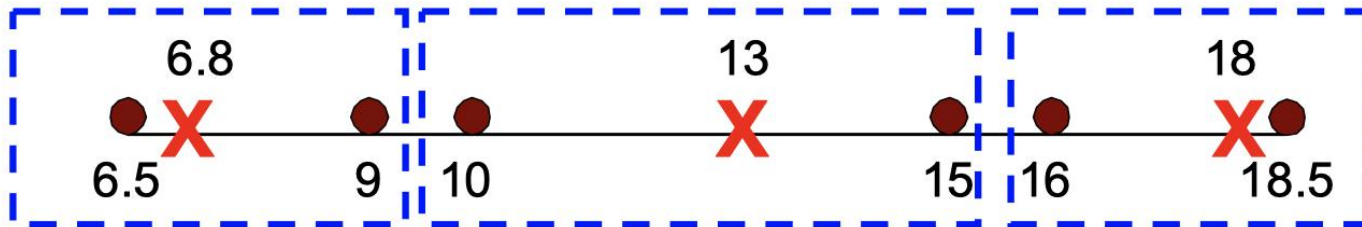
Iteration 10



Handling Empty Clusters



Basic K-means algorithm can yield **empty clusters**



**Empty
Cluster**

Handling Empty Clusters



Basic K-means algorithm can yield **empty clusters**

If happens, find a replacement centroid to replace the centroid of the empty cluster, several strategies:

- Choose the point that contributes most to SSE
- Choose a point from the cluster with the highest SSE
- If there are several empty clusters, the above can be repeated several times.

-
- 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
-

Pre-processing and Post-processing



Pre-processing

- Normalize the data
- Eliminate outliers

Post-processing

- Eliminate small clusters that may represent outliers
- Split 'loose' clusters, i.e., clusters with relatively high SSE
- Merge clusters that are 'close' and that have relatively low SSE

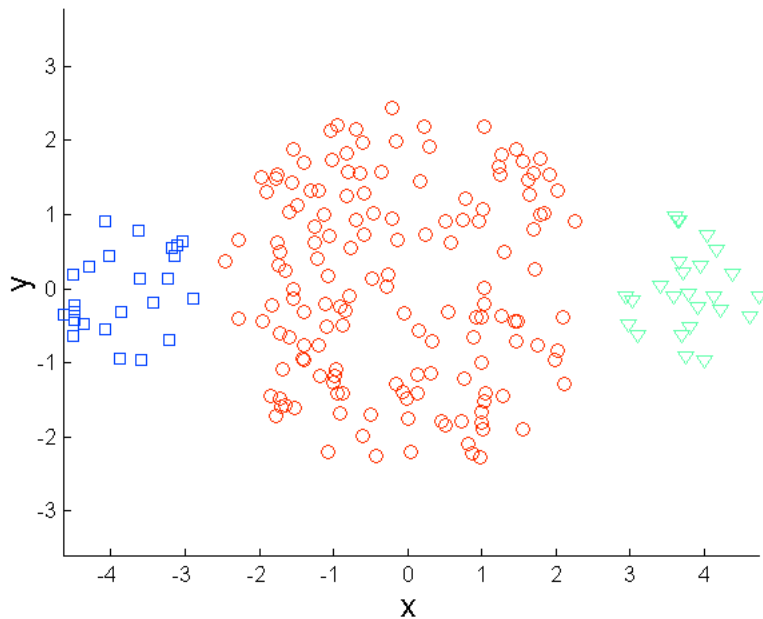


K-means has problems when clusters are of differing

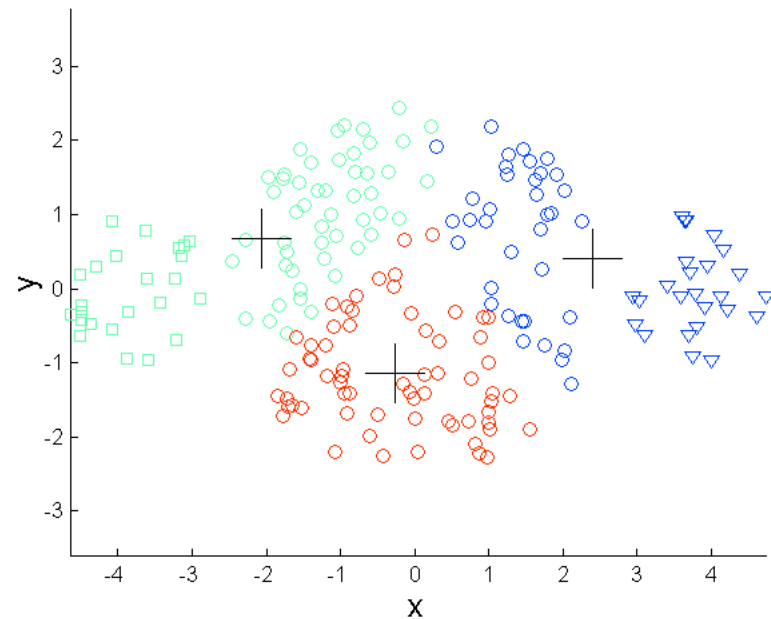
- Sizes
- Densities
- Non-globular shapes

K-means has problems when the data contains outliers.

Limitations of K-means: Differing Sizes

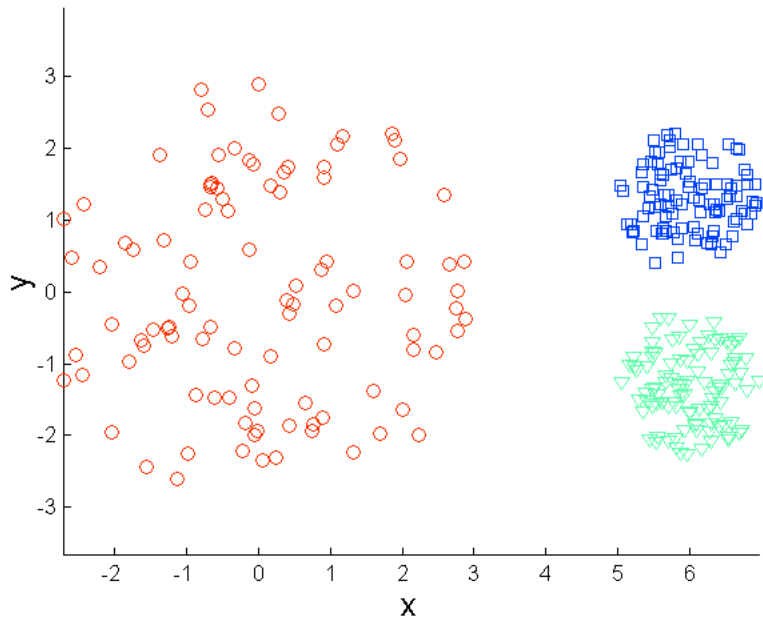


Original Points

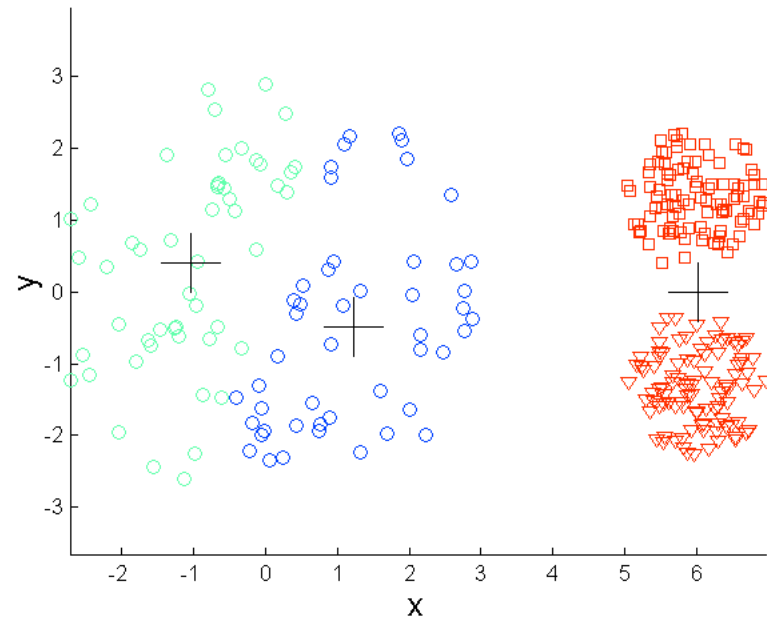


K-means (3 Clusters)

Limitations of K-means: Differing Density

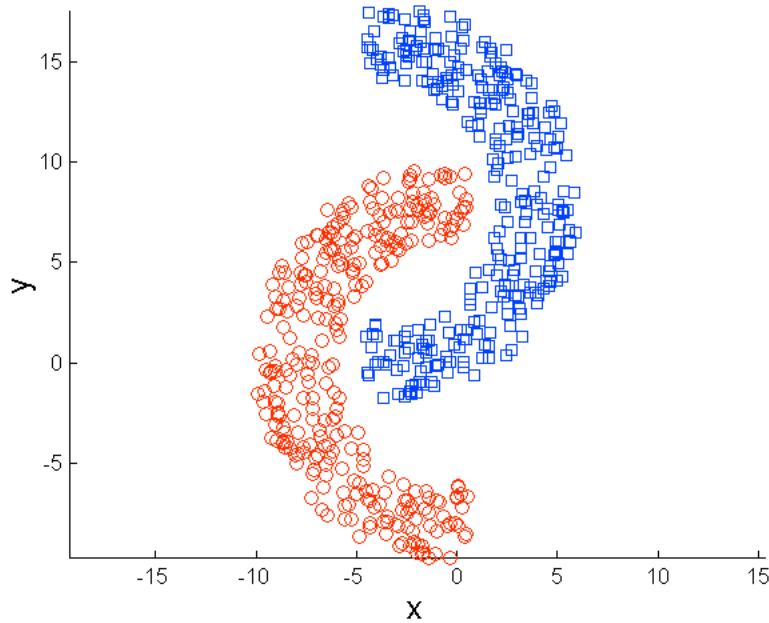


Original Points

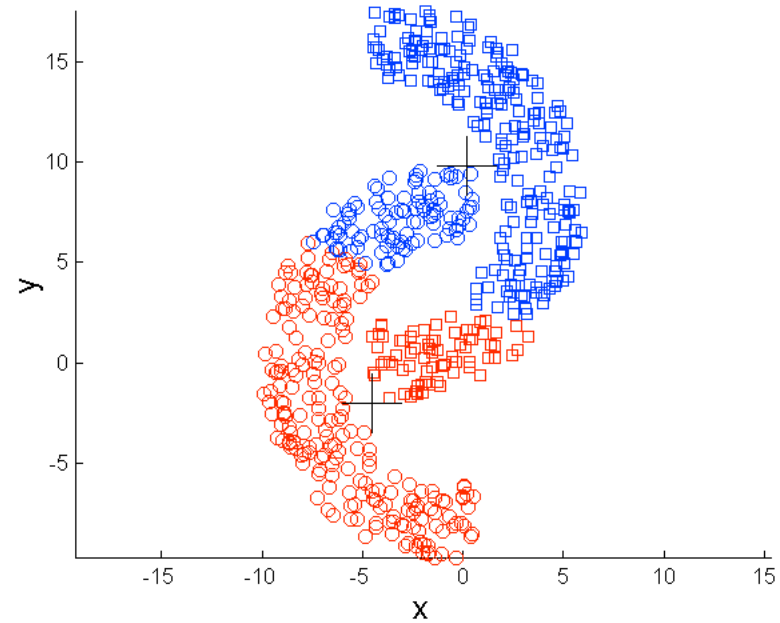


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes

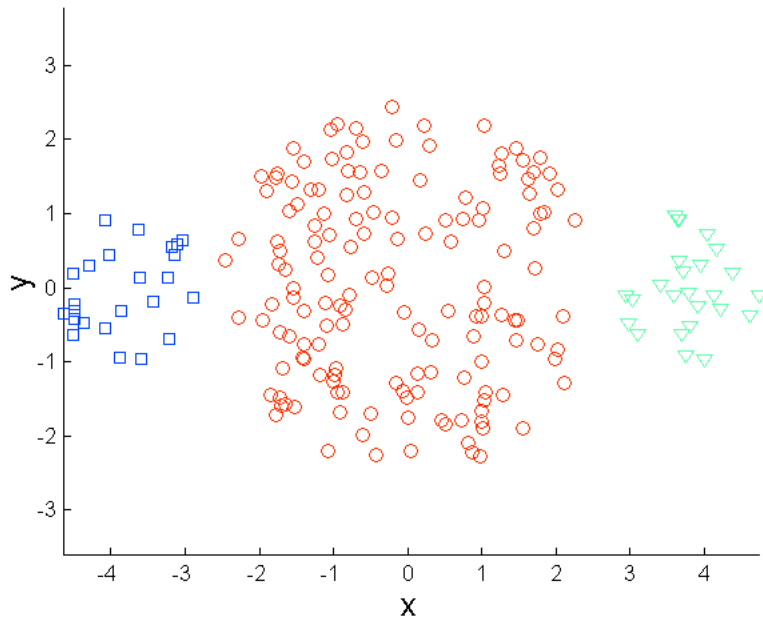


Original Points

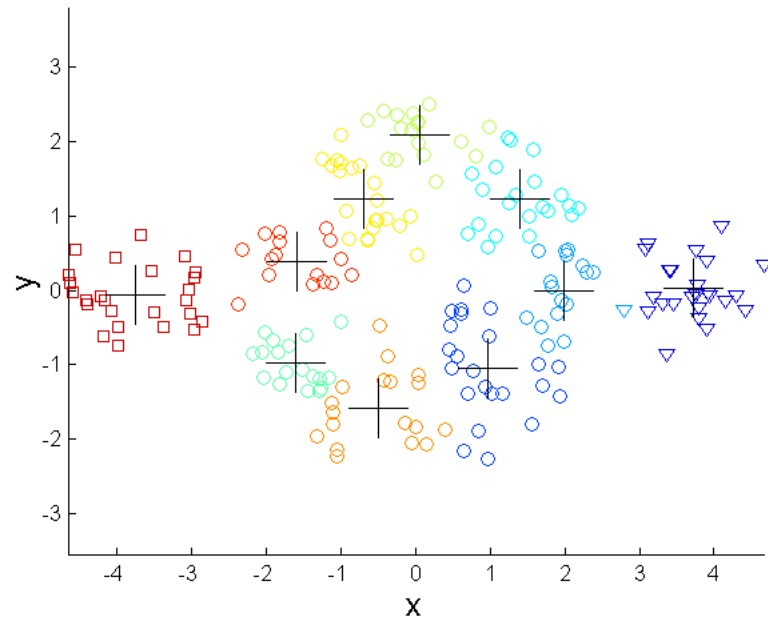


K-means (2 Clusters)

Overcoming K-means Limitations



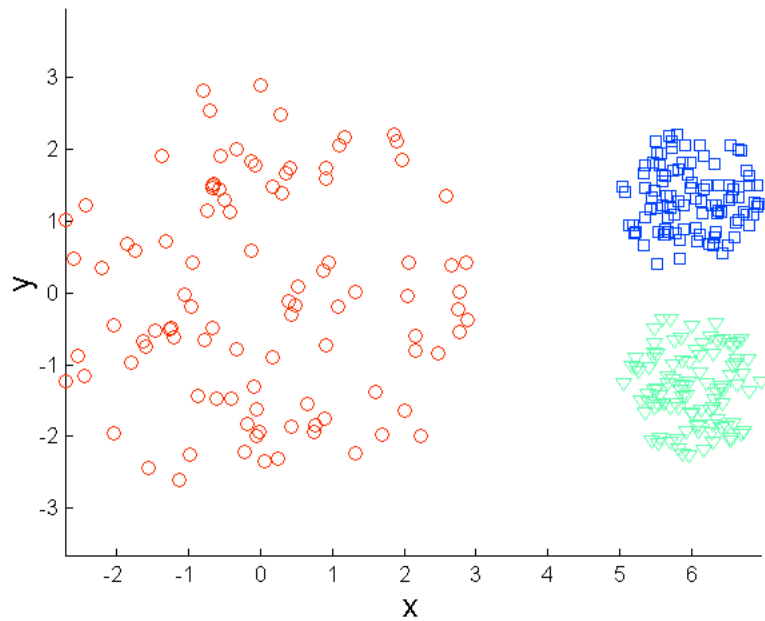
Original Points



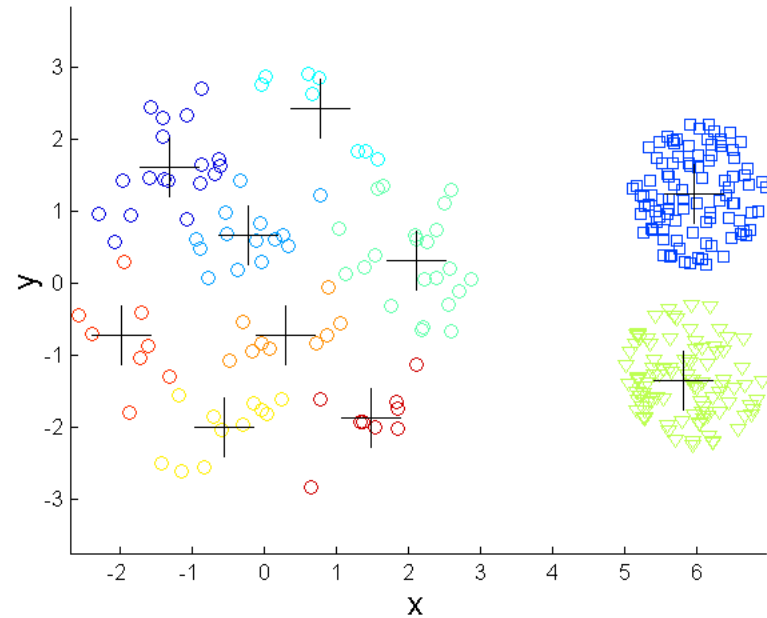
K-means Clusters

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

Overcoming K-means Limitations

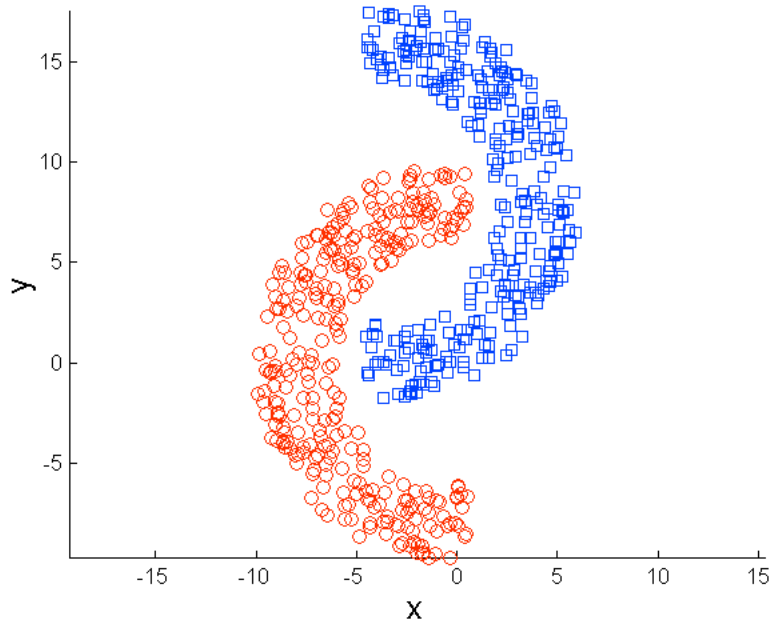


Original Points

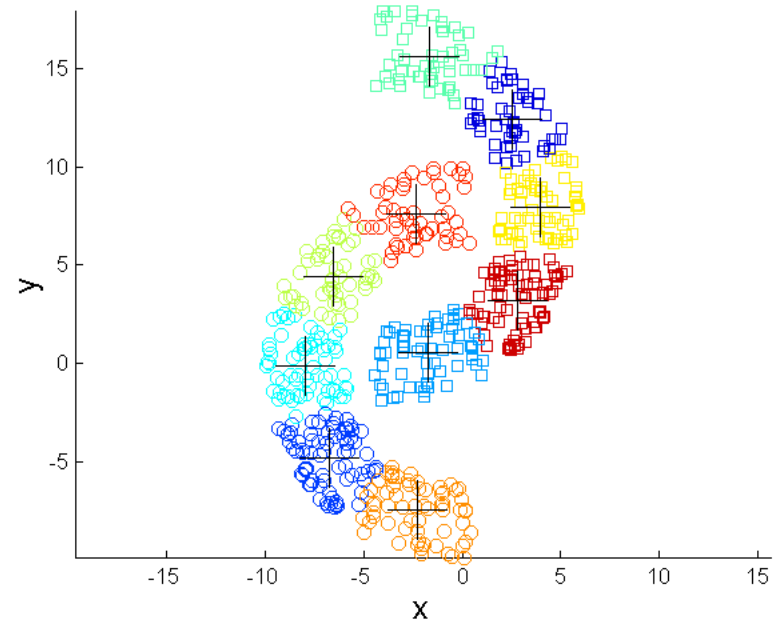


K-means Clusters

Overcoming K-means Limitations



Original Points



K-means Clusters

K-means++ [Arthur et al. '07]



Spreads out the centers

Choose first center, x_1 , uniformly at random from the data set

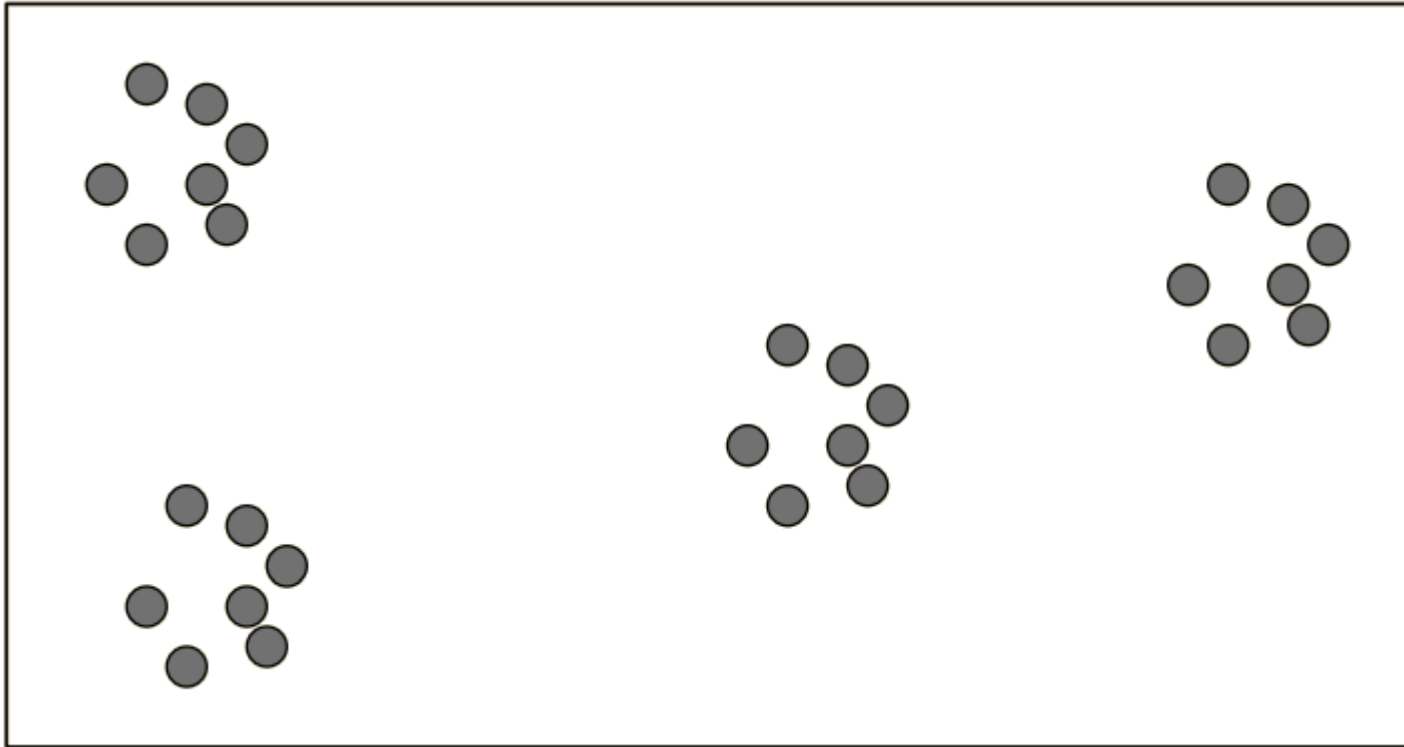
Repeat for $2 \leq i \leq k$:

- Choose x' to be equal to a data point **sampled from the distribution**:

$$\frac{D(x')^2}{\sum_{x \in \mathcal{X}} D(x)^2}$$

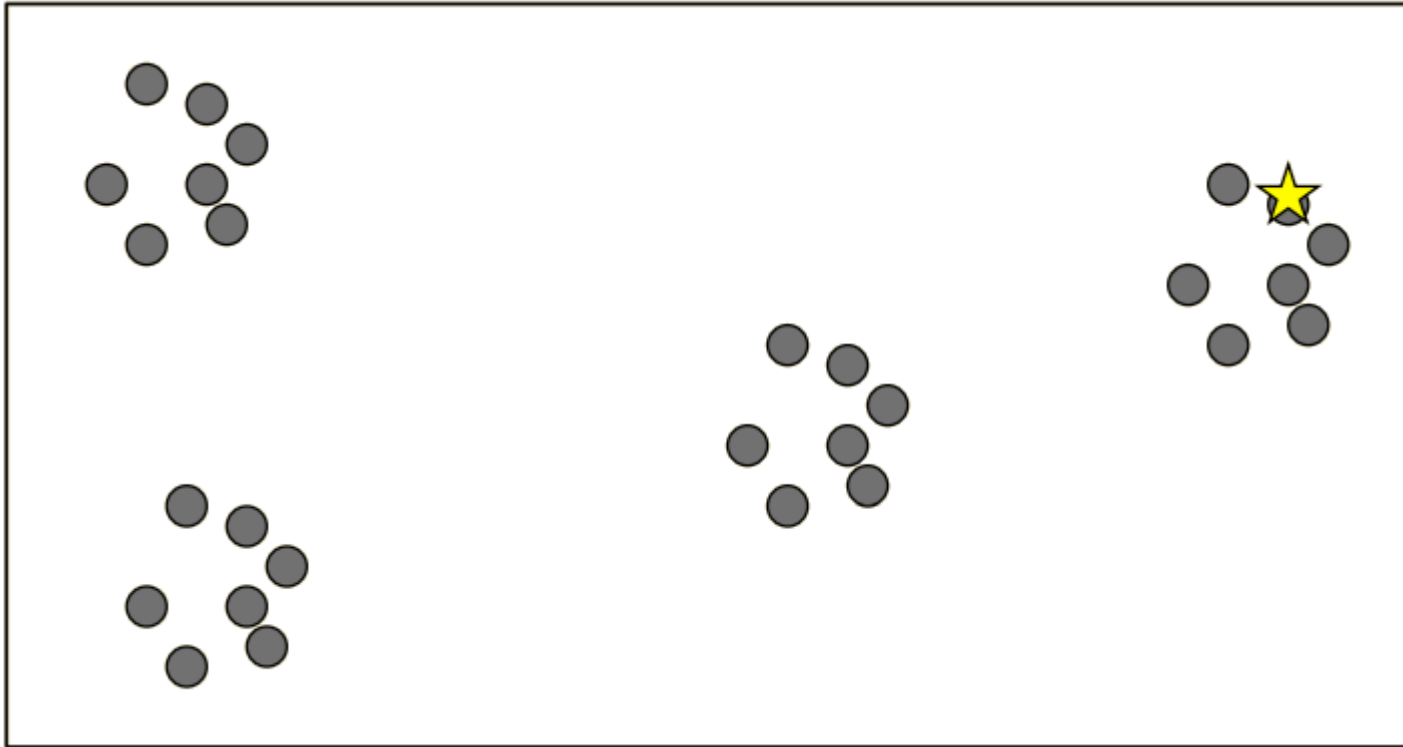
- $D(x)$: the shortest distance from a data point x to the closest center we have already chosen

K-means++ [Arthur et al. '07]



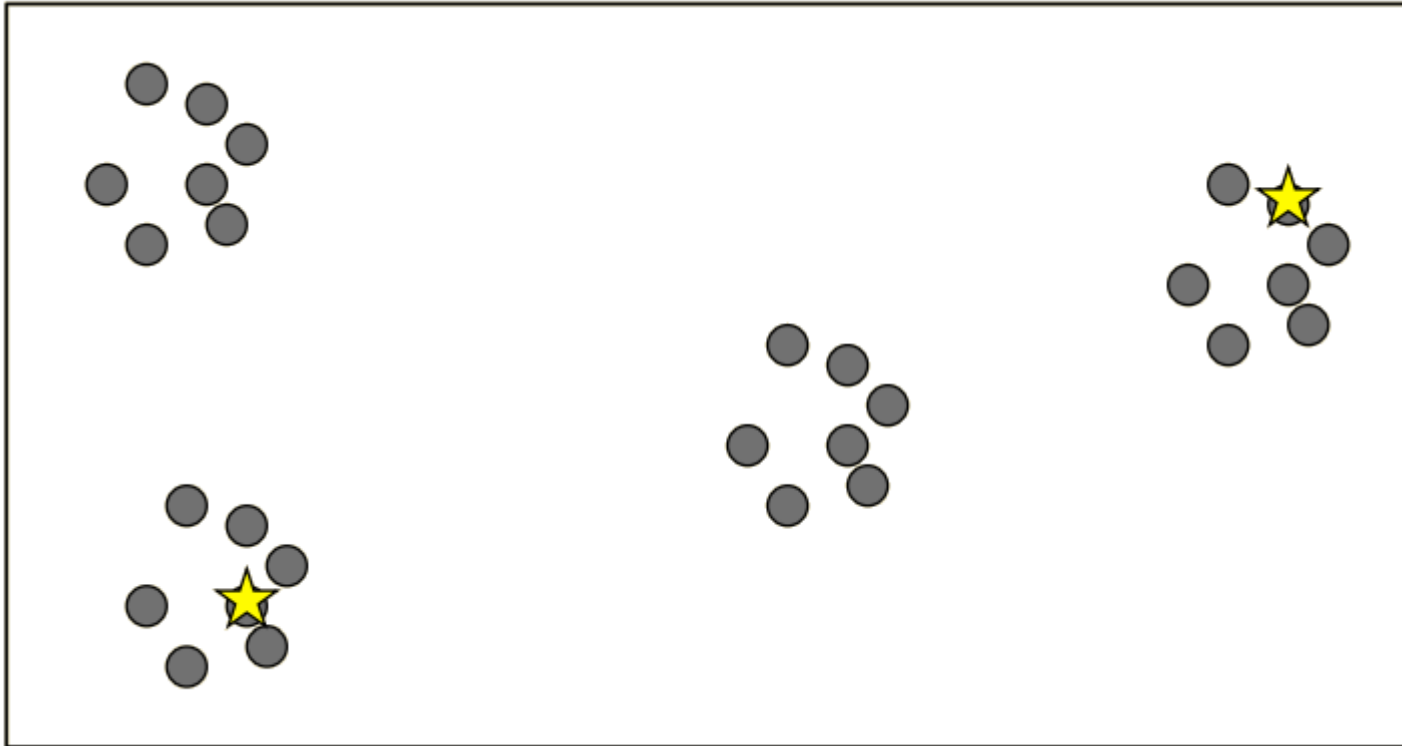
$$\frac{D(x')^2}{\sum_{x \in \mathcal{X}} D(x)^2}$$

K-means++ [Arthur et al. '07]



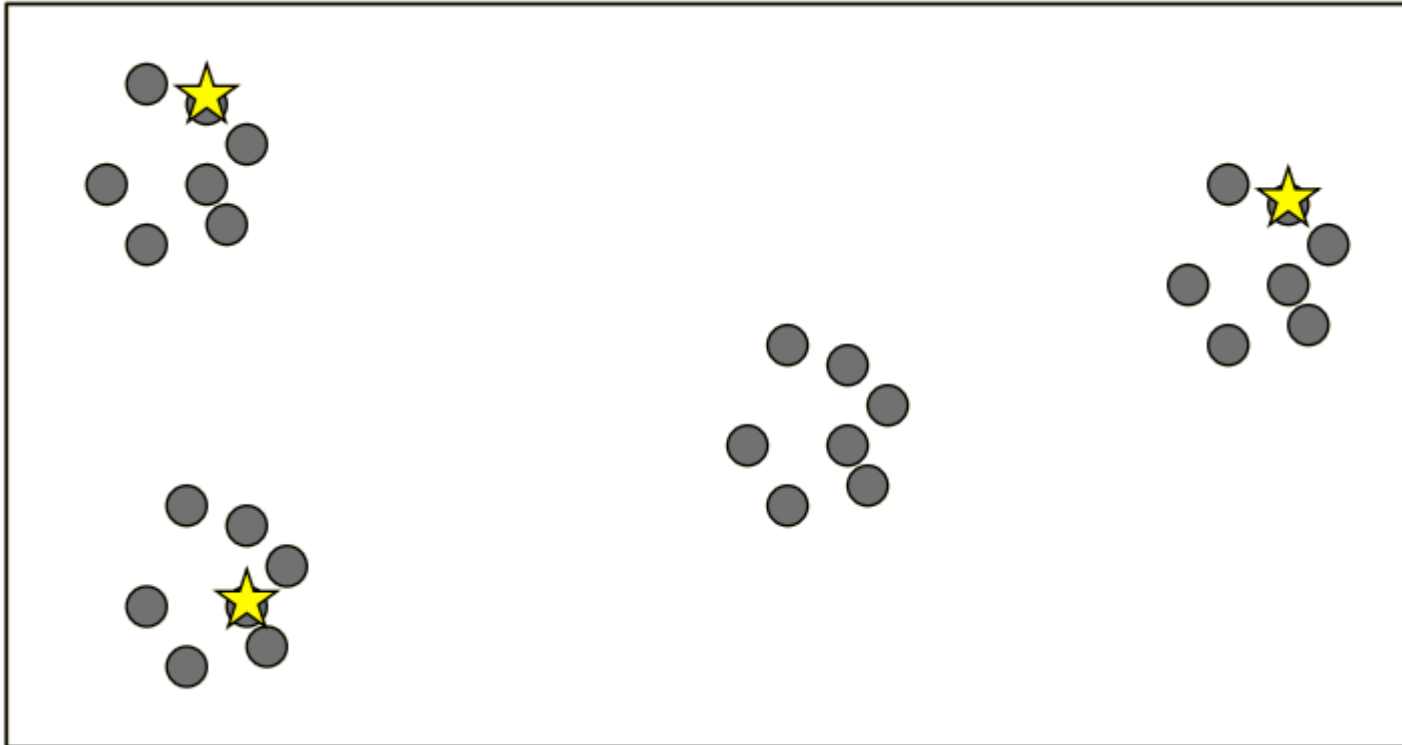
$$\frac{D(x')^2}{\sum_{x \in \mathcal{X}} D(x)^2}$$

K-means++ [Arthur et al. '07]



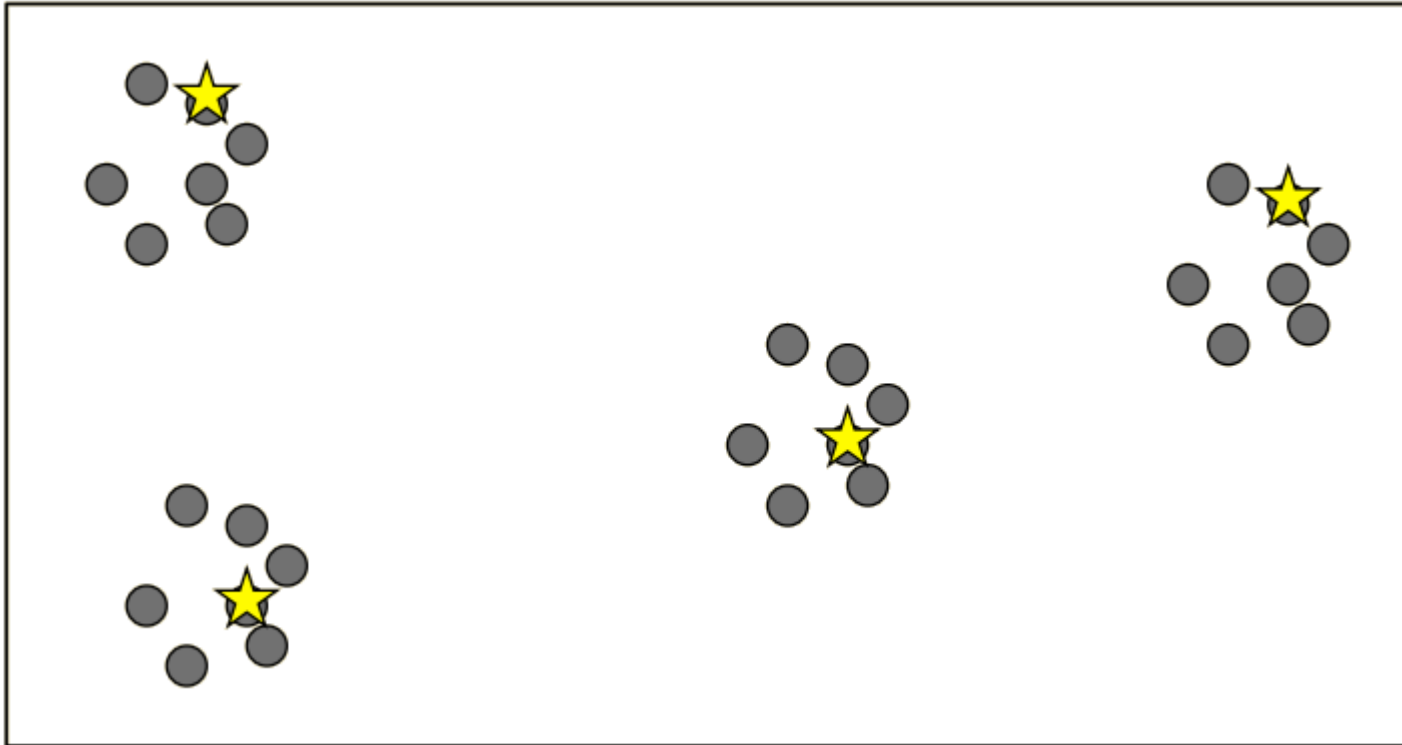
$$\frac{D(x')^2}{\sum_{x \in \mathcal{X}} D(x)^2}$$

K-means++ [Arthur et al. '07]



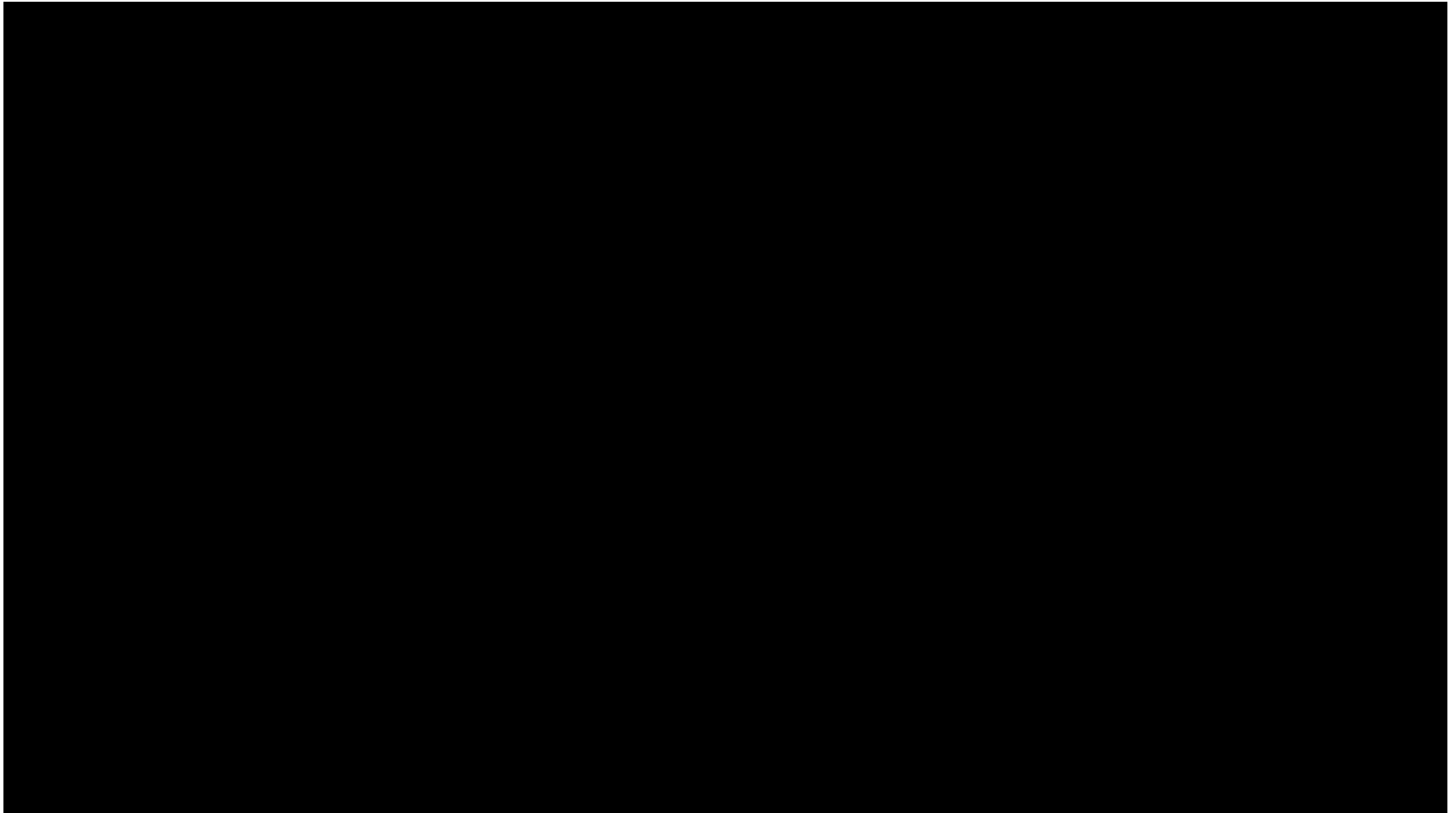
$$\frac{D(x')^2}{\sum_{x \in \mathcal{X}} D(x)^2}$$

K-means++ [Arthur et al. '07]



$$\frac{D(x')^2}{\sum_{x \in \mathcal{X}} D(x)^2}$$

K-means++



K-means++ [Arthur et al. '07]



What's **wrong** with K-means++?

- Needs K passes over the data
- In large data applications, not only the data is massive, but also K is typically large (e.g., easily 1000).
- Does not scale!



Intuition of

- K-means++ samples one point per iteration and updates its distribution
- What if we **oversample** by sampling each point independently with a larger probability?
- Intuitively equivalent to updating the distribution much less frequently
- Coarser sampling
- Turns out to be sufficient: K-means||

Scalable K-means



主要思路在于改变每次遍历时的取样策略，并非按照 k-means++ 那样每次遍历只取样一个样本，而是每次遍历取样 $O(k)$ 个样本，重复该取样过程大约 $O(\log n)$ 次，共得到 $O(k \log n)$ 个样本点组成的集合

然后再聚类这 $O(k \log n)$ 个点成 k 个点，最后将这 k 个点作为初始聚类中心

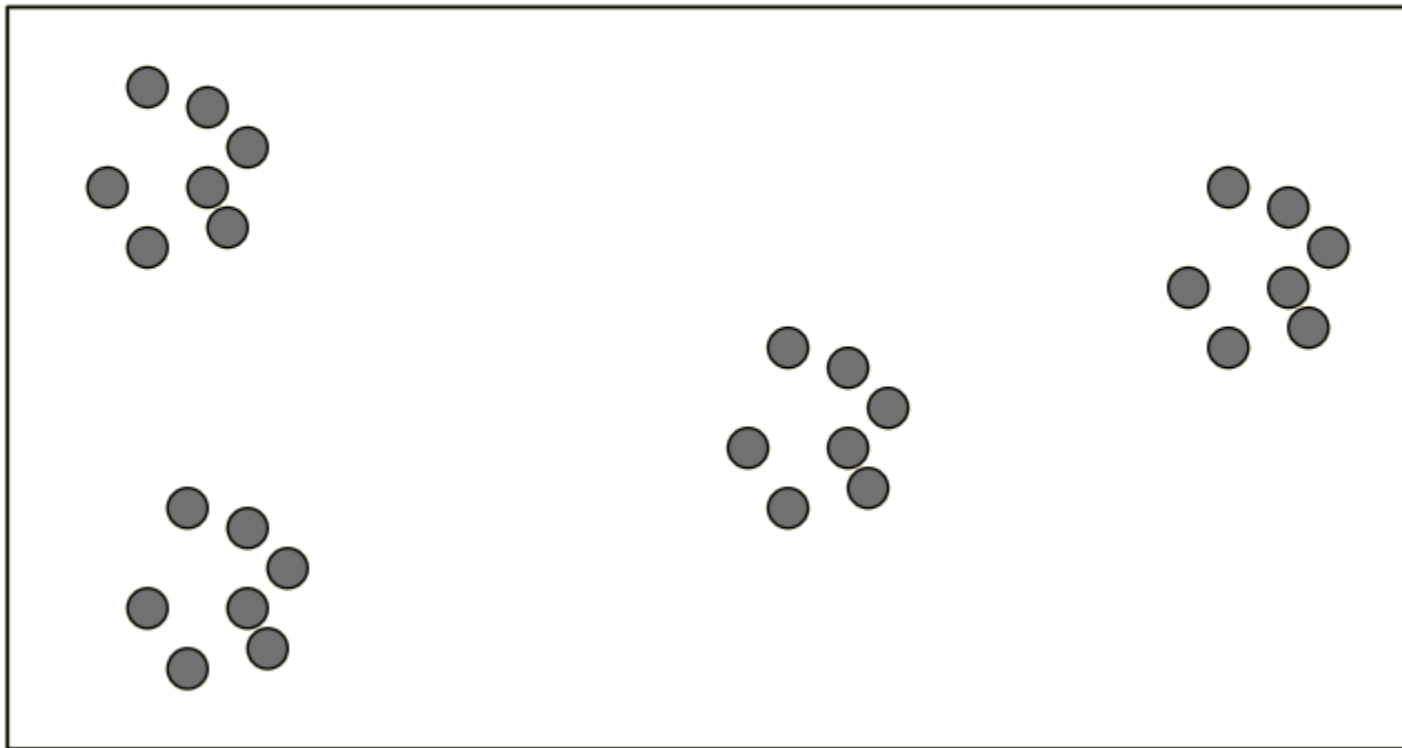
实际实验证明 $O(\log n)$ 次重复取样是不需要的，一般5次重复取样就可以得到一个较好的聚类初始中心

Scalable K-means



K-means|| Initialization

$K=4$,
Oversampling factor = 3

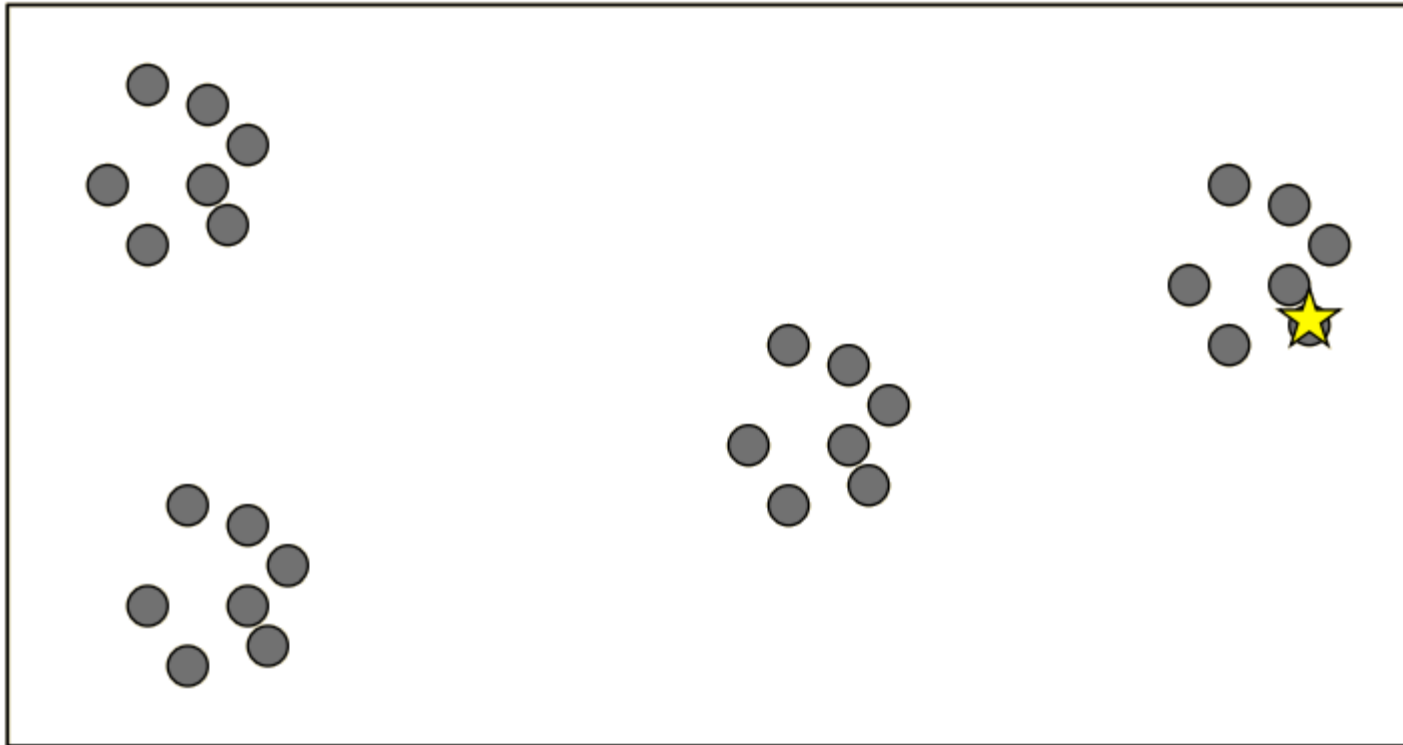


Scalable K-means



K-means|| Initialization

$K=4$,
Oversampling factor = 3

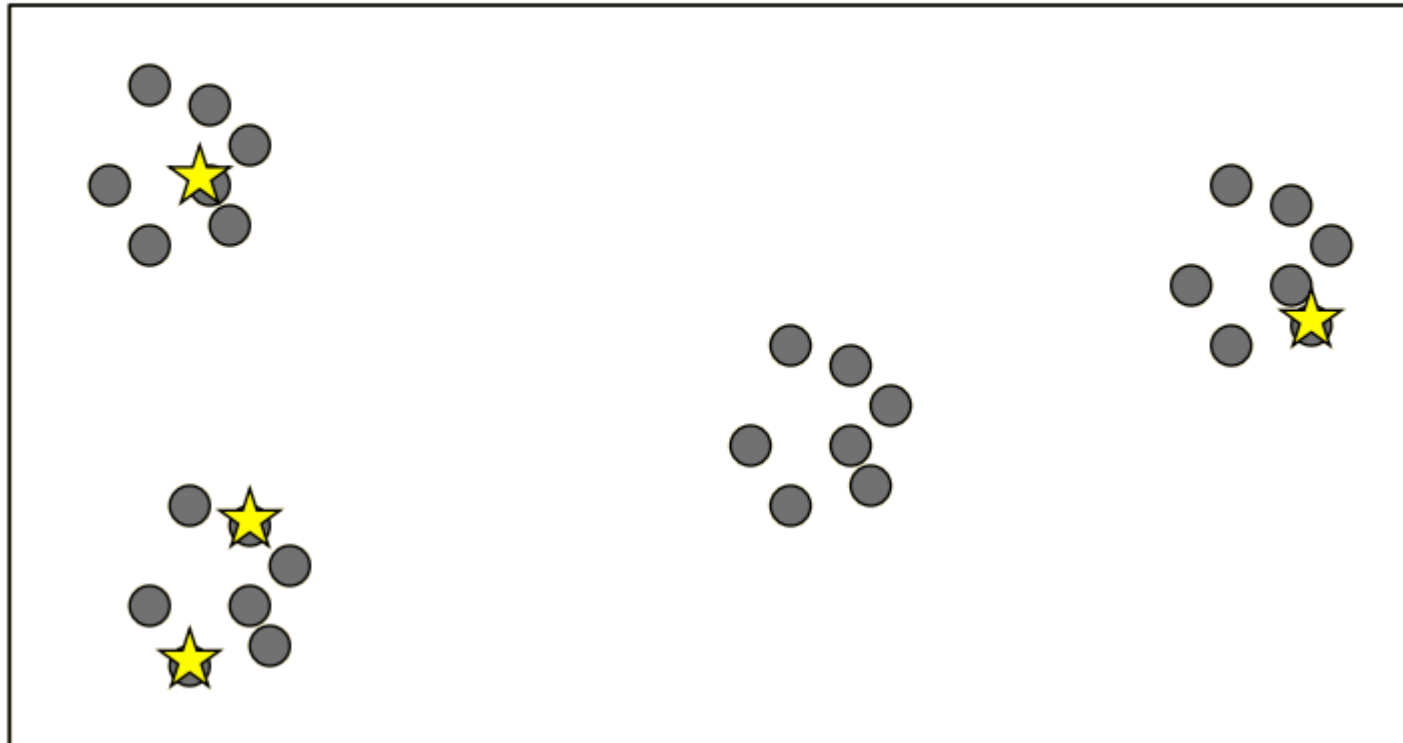


Scalable K-means



K-means|| Initialization

$K=4$,
Oversampling factor = 3



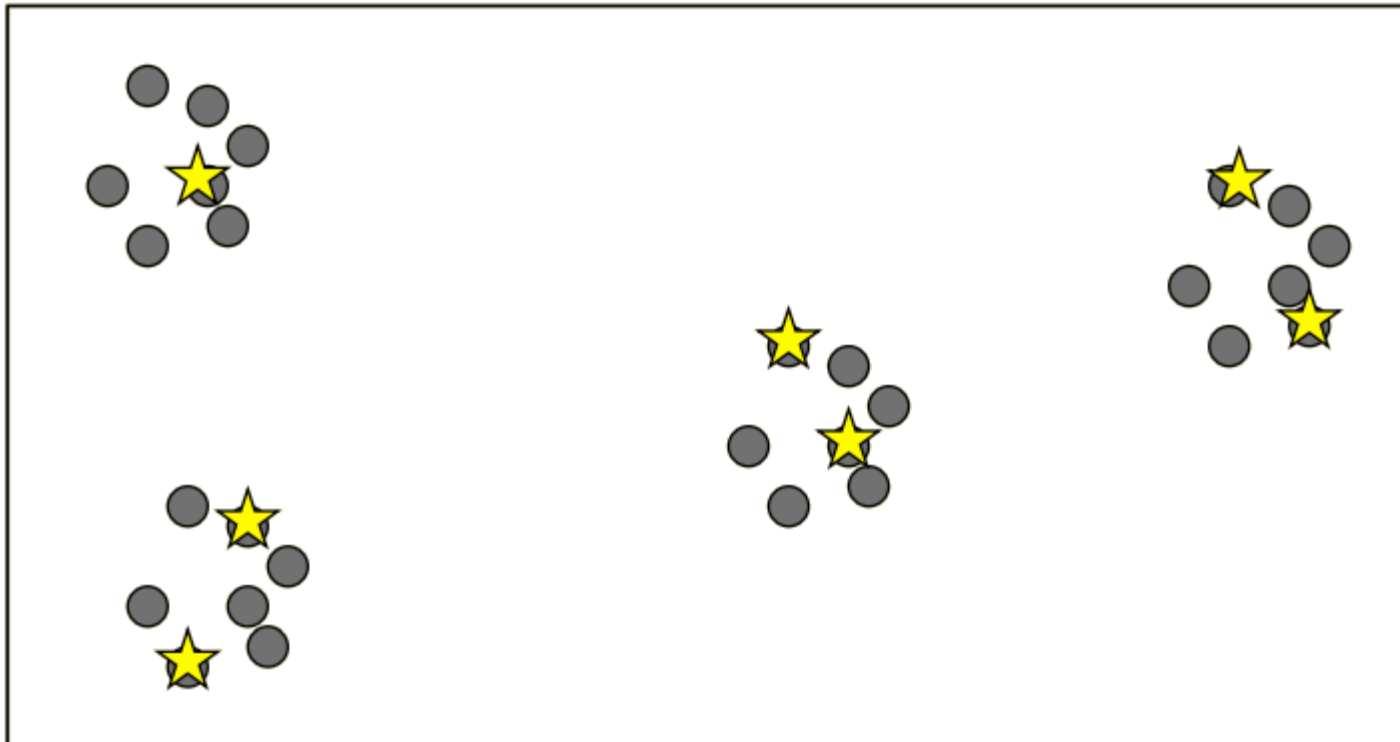
Scalable K-means



K-means|| Initialization

$K=4,$

Oversampling factor = 3



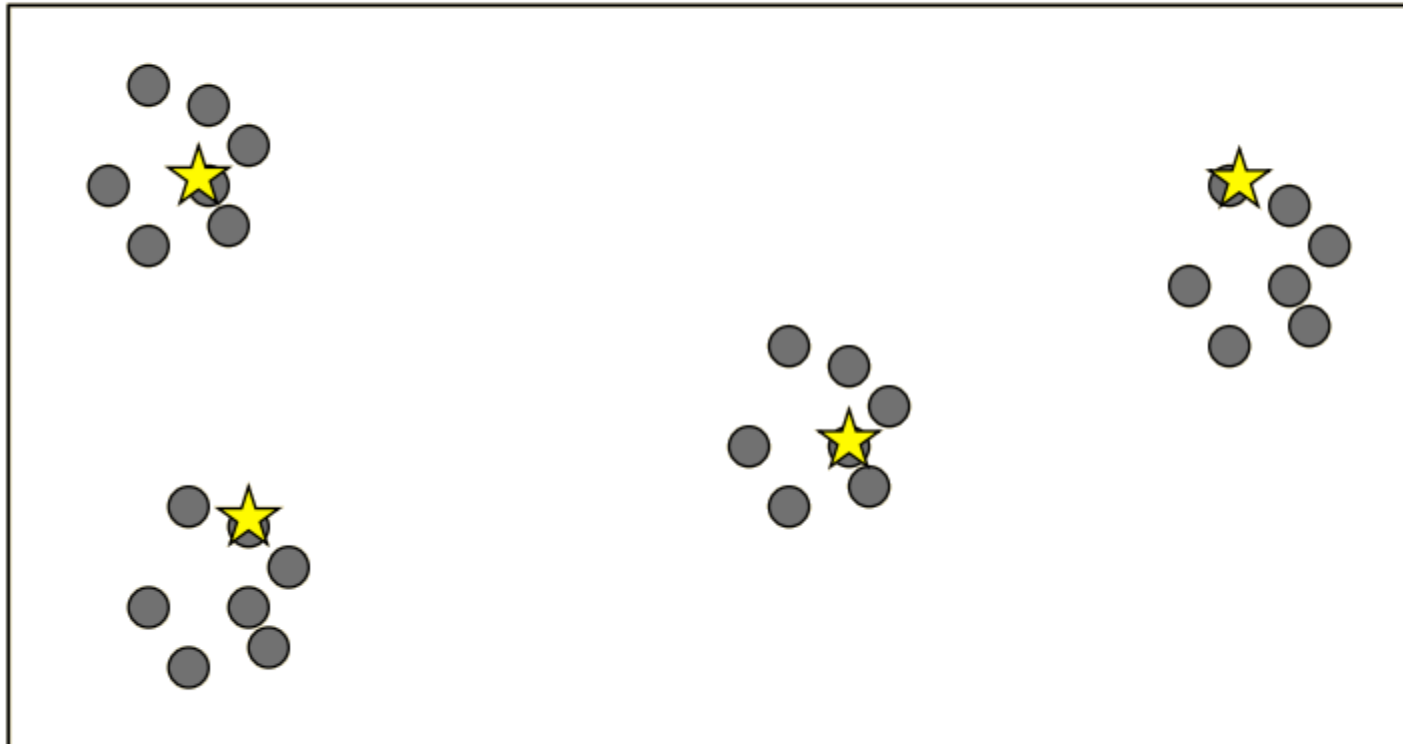
Scalable K-means



K-means|| Initialization

$K=4$,

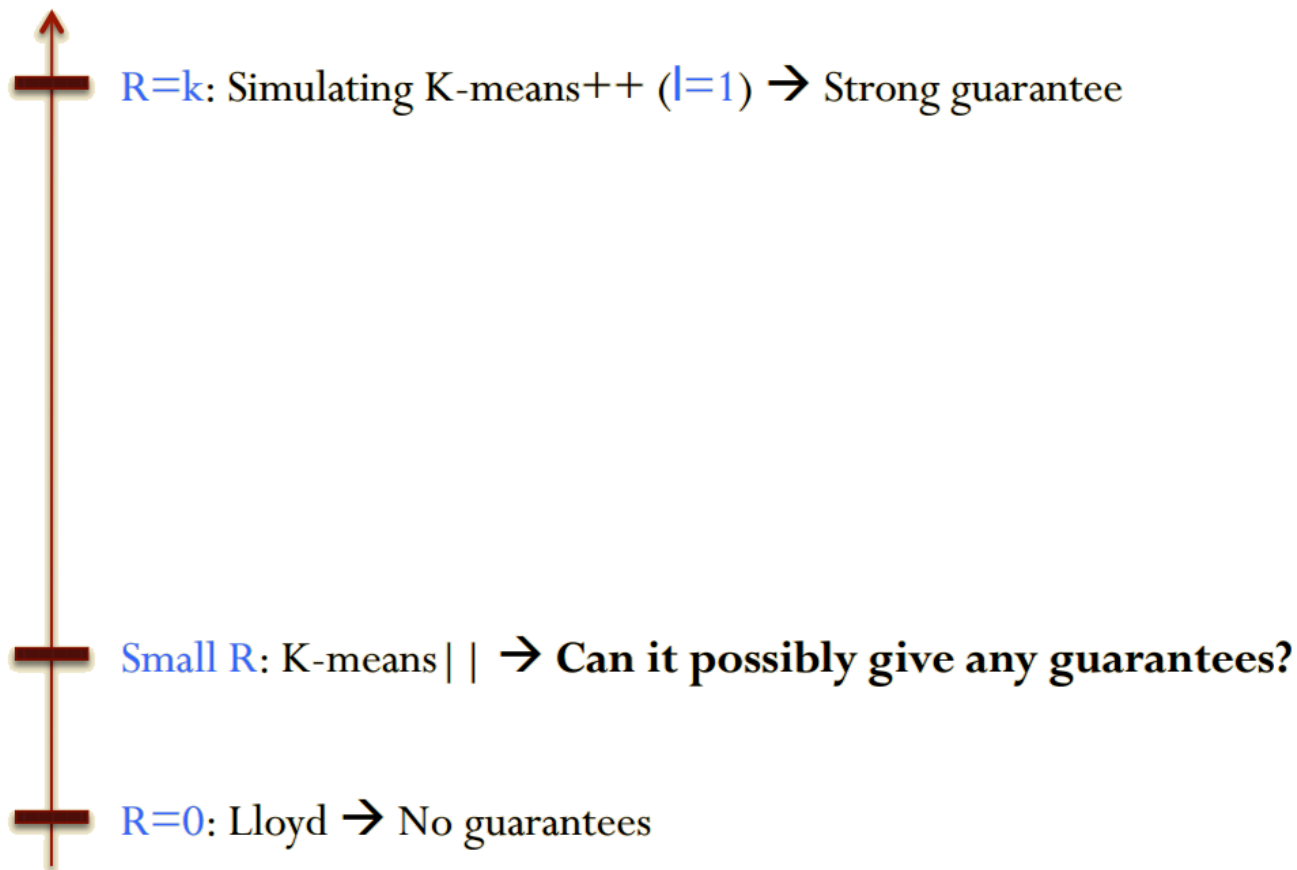
Oversampling factor = 3



Cluster the intermediate centers

K-means, K-means++, and K-means||

Number of
iterations (R)



Theorem

- Optimal clustering: the sum of the squared distances between each point and its closest center is minimal
 - ◆ Solving this problem exactly is NP-hard
- Let ψ = cost of initial clustering, ψ' = cost of clustering after convergence, OPT = cost of the optimal clustering
- Using K-means++ for clustering the intermediate centers, the overall approximation factor = $O(\log k)$
 - ◆ It means $\psi' = O(\log k) * OPT$
 - ◆ 也就是说最终确定的质心能够使cost达到最优聚类的cost的 $O(\log k)$ 倍
- K-means|| produces a constant-factor approximation to OPT , using only $O(\log(\psi/OPT))$ iterations

Experimental Results: Quality

Let $\mathcal{C} = \{c_1, \dots, c_k\}$ be a set of points and let $Y \subseteq X$.
We define the *cost* of Y with respect to \mathcal{C} as

$$\phi_Y(\mathcal{C}) = \sum_{y \in Y} d^2(y, \mathcal{C}) = \sum_{y \in Y} \min_{i=1, \dots, k} \|y - c_i\|^2.$$

	Clustering Cost Right After Initialization	Clustering Cost After Convergence
Random	NA	22,000
K-means++	430	65
K-means	16	14

**GAUSSMIXTURE: 10,000 points in 15 dimensions
K=50**

Costs scaled down by 10^4

Experimental Results: Convergence

- K-means|| reduces number of iterations even more than K-means++

	Iterations After Initialization
Random	167
K-means++	42
K-means	28

SPAM: 4,601 points in 58 dimensions
K=50

Kmeans练习



假设有如下八个点：(1, 2), (2, 4), (1, 9), (6, 5), (4, 2), (7, 2), (8, 2), (4, 3), 使用k-Means算法对其进行聚类。设K=2, 初始聚类两个中心点 坐标分别为(1, 2), (8, 2), 算法使用曼哈顿距离 (L1 距离) 作为距离度量。请计算一次迭代后的聚类中心坐标, 并写出上述八个点分别属于哪个簇 (cluster)。

对于点(1, 2):

$$d_{11} = |1 - 1| + |2 - 2| = 0, \quad d_{12} = |1 - 8| + |2 - 2| = 7。$$

因此, (1, 2)应当被聚为第 1 类;

对于点(2, 4):

$$d_{21} = |2 - 1| + |4 - 2| = 3, \quad d_{22} = |2 - 8| + |4 - 2| = 8。$$

因此, (2, 4)应当被聚为第 1 类;

对于点(1, 9):

$$d_{31} = |1 - 1| + |9 - 2| = 7, \quad d_{32} = |1 - 8| + |9 - 2| = 14。$$

因此, (1, 9)应当被聚类为第 1 类;

对于点(6, 5):

$$d_{41} = |6 - 1| + |5 - 2| = 8, \quad d_{42} = |6 - 8| + |5 - 2| = 5。$$

因此, (6, 5)应当被聚类为第 2 类;

Kmeans练习



假设有如下八个点：(1, 2), (2, 4), (1, 9), (6, 5), (4, 2), (7, 2), (8, 2), (4, 3), 使用k-Means算法对其进行聚类。设K=2, 初始聚类两个中心点 坐标分别为(1, 2), (8, 2), 算法使用曼哈顿距离 (L1 距离) 作为距离度量。请计算一次迭代后的聚类中心坐标, 并写出上述八个点分别属于哪个簇 (cluster)。

对于点(4, 2):

$$d_{51} = |4 - 1| + |2 - 2| = 3, \quad d_{52} = |4 - 8| + |2 - 2| = 4。$$

因此, 点(4, 2)应当被聚类为第 1 类;

对于点(7, 2):

$$d_{61} = |7 - 1| + |2 - 2| = 6, \quad d_{62} = |7 - 8| + |2 - 2| = 1。$$

因此, 点(7, 2)应当被聚类为第 2 类;

对于点(8, 2):

$$d_{71} = |8 - 1| + |2 - 2| = 7, \quad d_{72} = |8 - 8| + |2 - 2| = 0。$$

因此, 点(8, 2)应当被聚类为第 2 类;

对于点(4, 3):

$$d_{81} = |4 - 1| + |3 - 2| = 4, \quad d_{82} = |4 - 8| + |3 - 2| = 5。$$

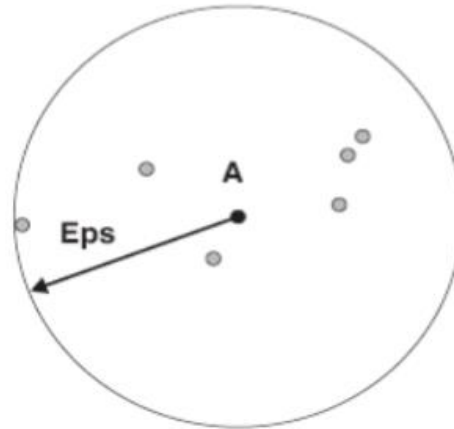
因此, 点(4, 3)应当被聚类为第 1 类。

综上, 类 1 的点有(1,2) (2,4) (1,9) (4,2) (4,3), 则对应的聚类中心为: (2.4,4)。

类 2 的点有(6,5) (7,2) (8,2), 则对应的聚类中心为: (7,3)。

DBSCAN is a density-based algorithm.

- Density = number of points within a specified radius (Eps)
- A point is a **core point** if it has more than a specified number of points (MinPts) within Eps (半径范围内有一定数量的点)
 - ◆ These are points that 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 (半径范围内未达到一定数量的点, 但是在**core point** 范围内)
- A **noise point** is any point that is not a core point or a border point. (啥都不是)



Density = 7 points

Figure 8.20. Center-based density.

- **Density** = number of points within a specified radius (Eps)

Core, border, noise points

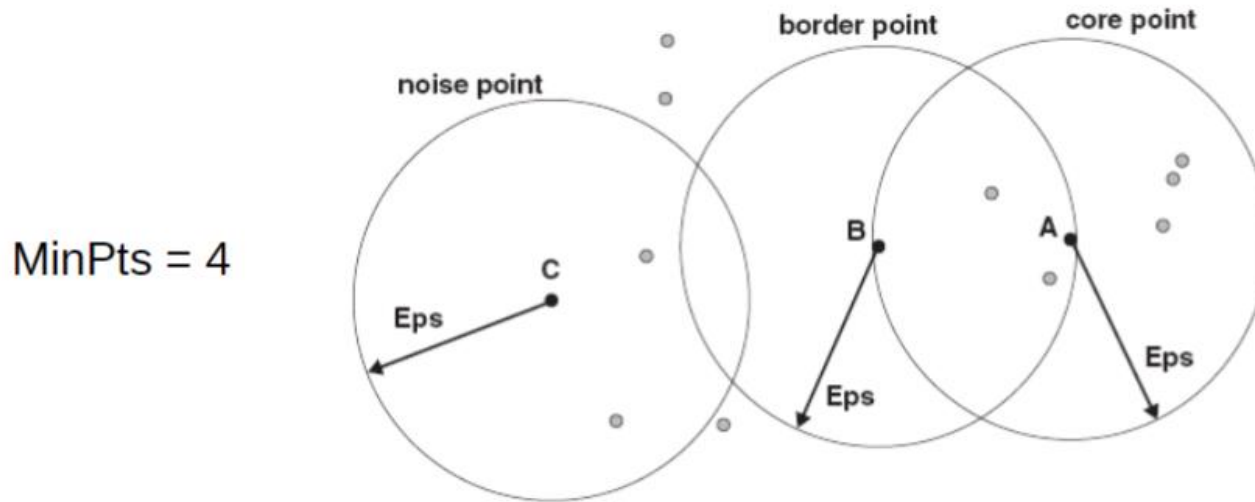
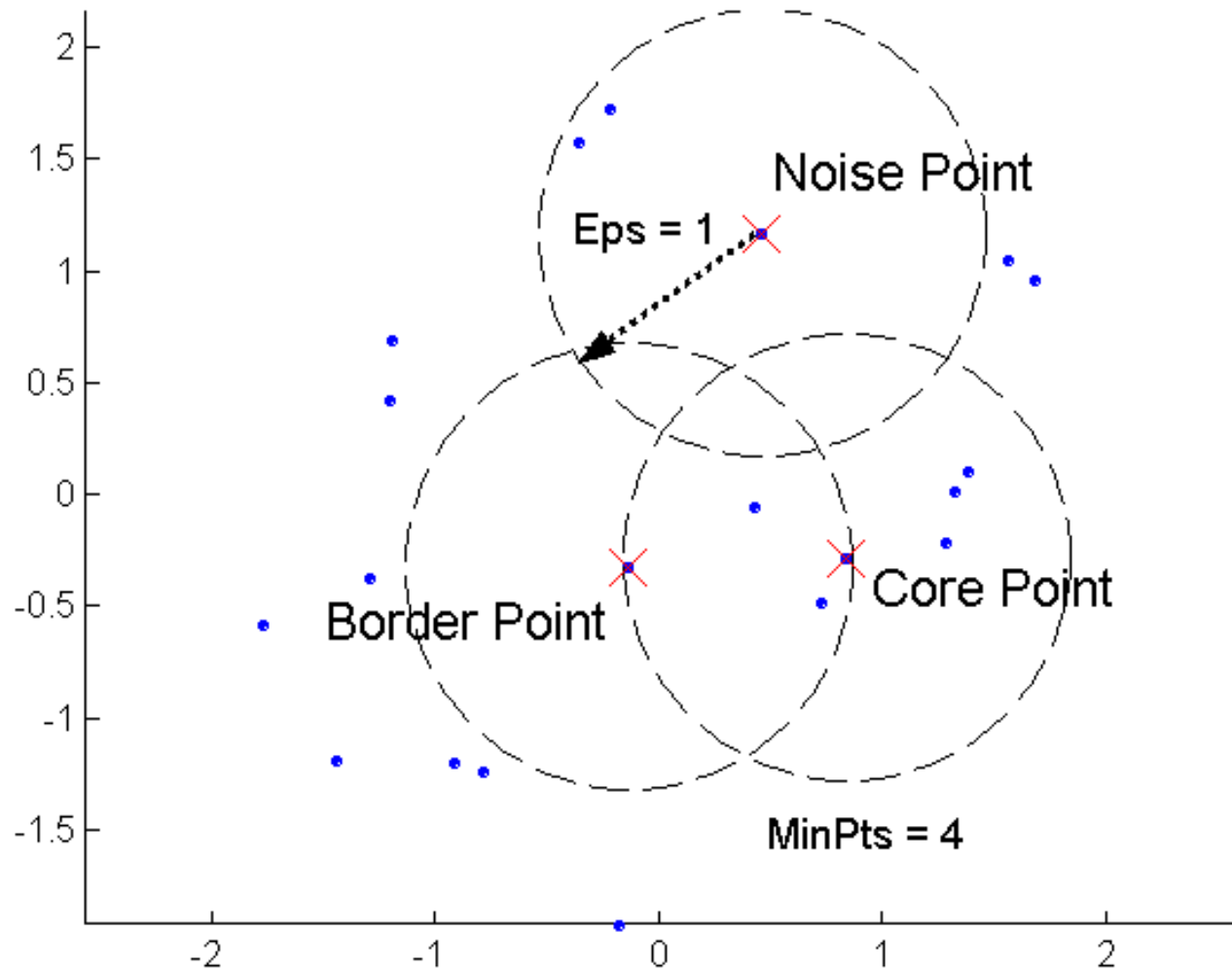


Figure 8.21. Core, border, and noise points.

- A point is a **core point** if it has more than a specified number of points (MinPts) within Eps. These are points that 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



Center-based approach

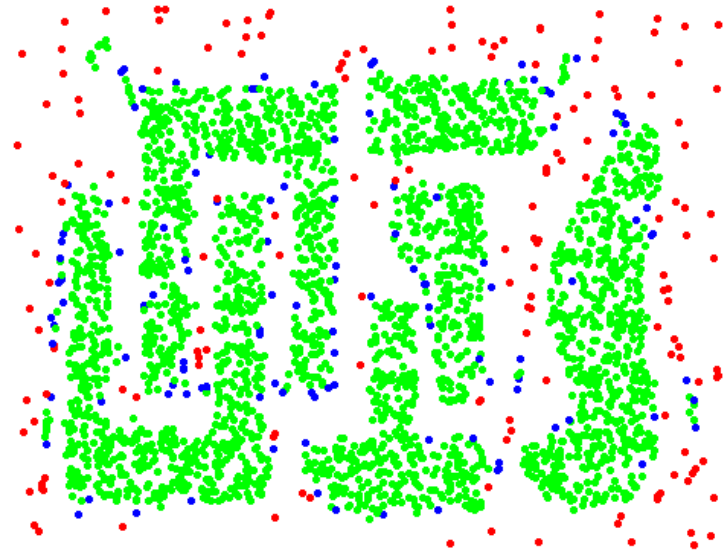
Algorithm 8.4 DBSCAN algorithm.

- 1: Label all points as core, border, or noise points.
 - 2: Eliminate noise points.
 - 3: Put an edge between all core points that are within Eps of each other.
 - 4: Make each group of connected core points into a separate cluster.
 - 5: Assign each border point to one of the clusters of its associated core points.
-

DBSCAN: Core, Border and Noise Points



Original Points



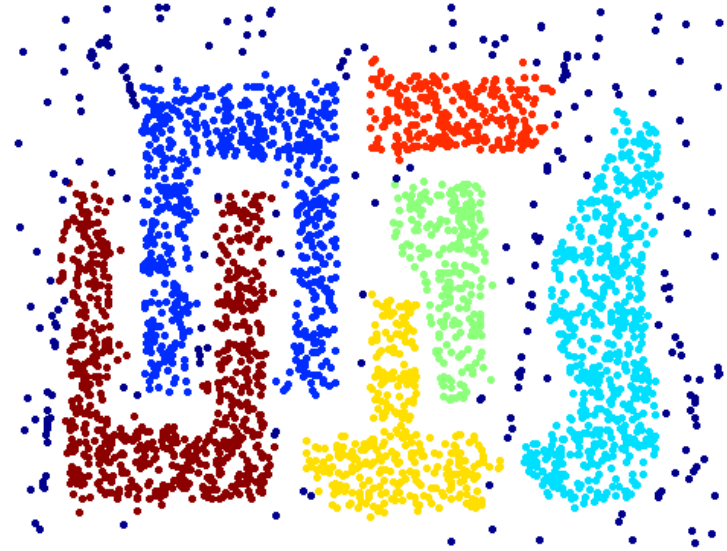
Point types: **core**,
border and **noise**

Eps = 10, MinPts = 4

When DBSCAN Works Well



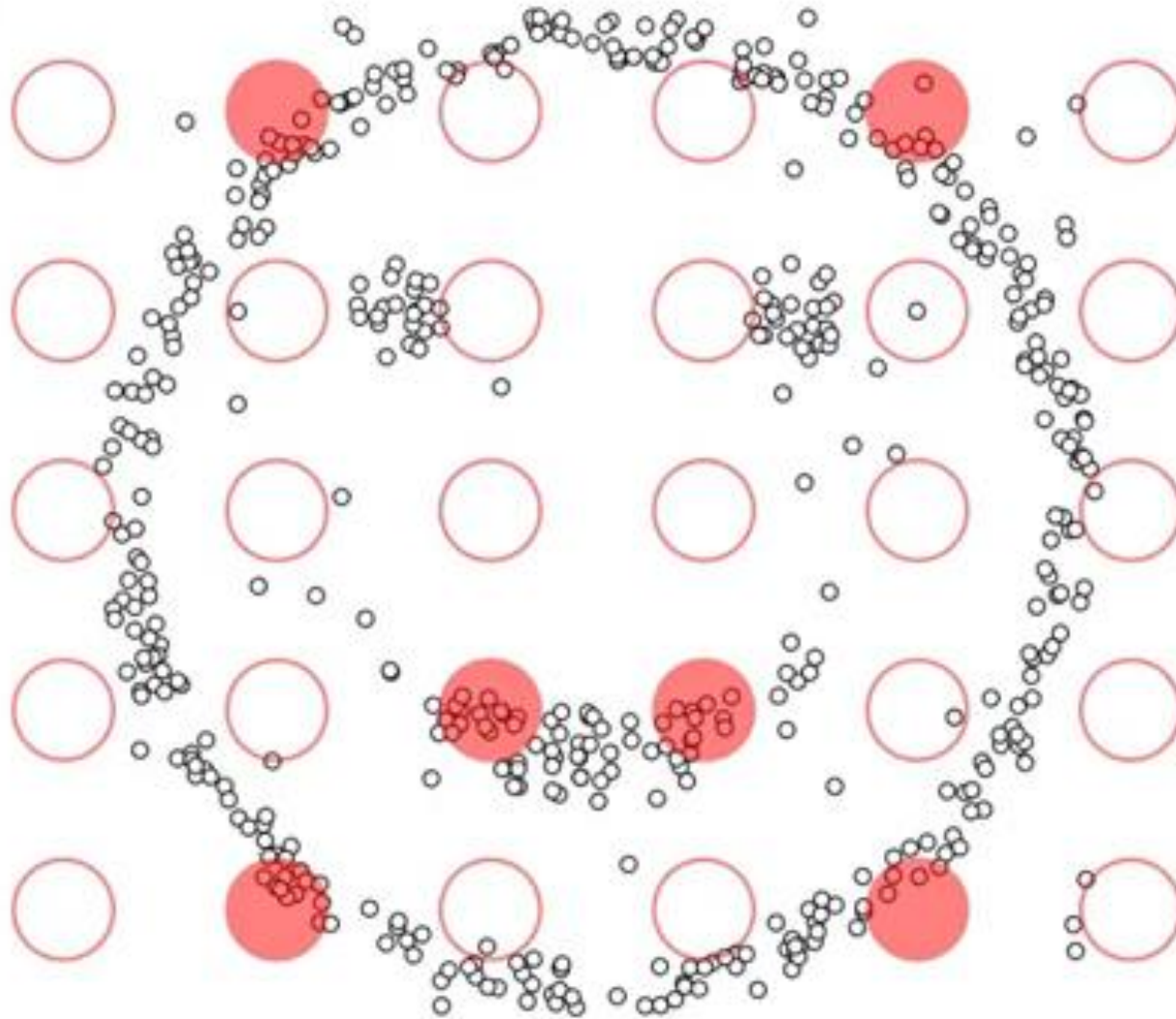
Original Points



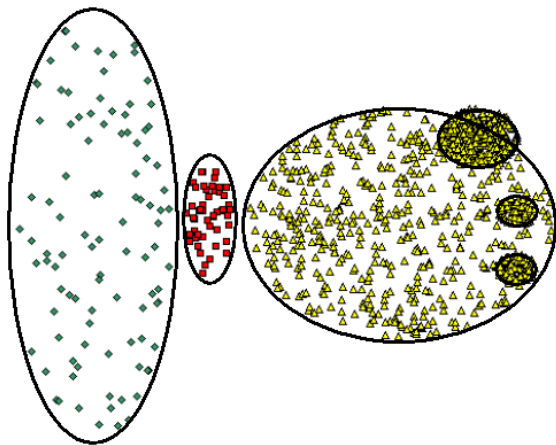
Clusters

- Resistant to Noise
- Can handle clusters of different shapes and sizes

DBSCAN



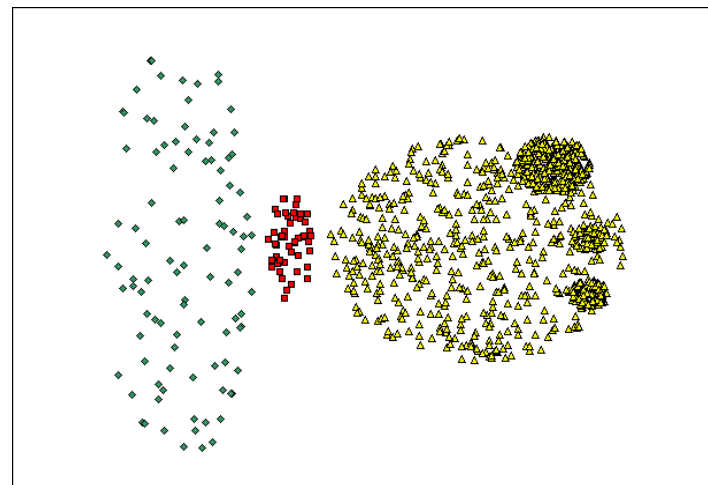
When DBSCAN Does NOT Work Well



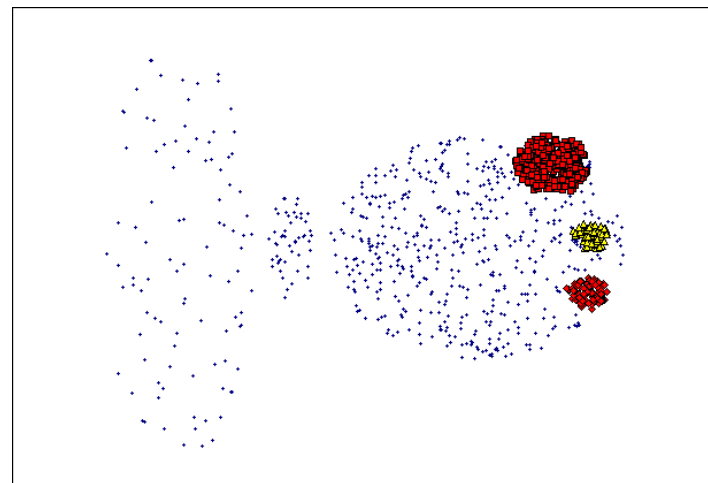
Original Points

- Varying densities
- High-dimensional data

维度灾难：距离失效



(MinPts=4, Eps=9.75).



(MinPts=4, Eps=9.92)

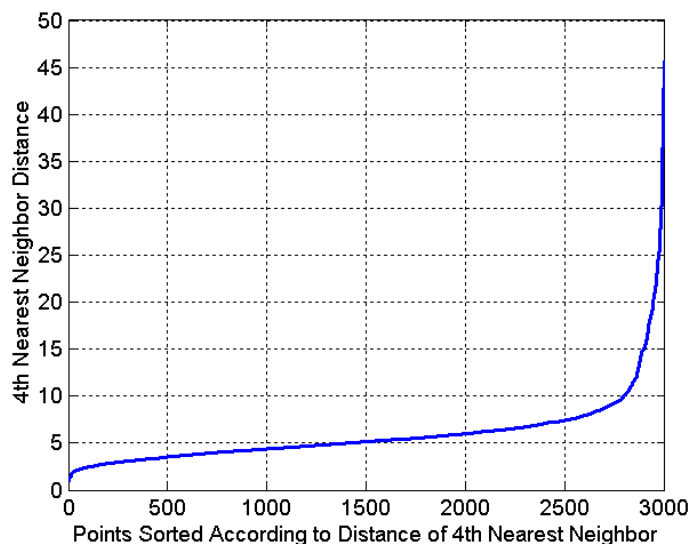
DBSCAN: Determining EPS and MinPts



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



Time and Space Complexity

The basic time complexity of the DBSCAN algorithm is $O(m \times \text{time to find points in the } Eps\text{-neighborhood})$, where m is the number of points. In the worst case, this complexity is $O(m^2)$. However, in low-dimensional spaces, there are data structures, such as kd-trees, that allow efficient retrieval of all points within a given distance of a specified point, and the time complexity can be as low as $O(m \log m)$. The space requirement of DBSCAN, even for high-dimensional data, is $O(m)$ because it is only necessary to keep a small amount of data for each point, i.e., the cluster label and the identification of each point as a core, border, or noise point.

2014 KDD TEST OF TIME AWARD



A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise [KDD 1996]



HOME

CONFERENCES

AWARDS

PUBLICATIONS

2014 SIGKDD TEST OF TIME AWARD

Aug 18 2014 /

2014 SIGKDD Test of Time Award:

The SIGKDD Test of Time award recognizes outstanding papers from past KDD Conferences beyond the last decade that have had an important impact on the data mining research community.

The 2014 Test of Time award recognizes the following influential contributions to SIGKDD that have withstood the test of time:

A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise [KDD 1996]





A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise [KDD 1996]

- Proposed the now well-known clustering algorithm DBSCAN
- 10000+ citations

[\[PDF\] A density-based algorithm for discovering clusters in large spatial databases with noise.](#)

[M Ester](#), [HP Kriegel](#), [J Sander](#), [X Xu](#) - Kdd, 1996 - [aaai.org](#)

Abstract Clustering algorithms are attractive for the task of class identification in spatial databases. However, the application to large spatial databases rises the following requirements for clustering algorithms: minimal requirements of domain knowledge to determine the input parameters, discovery of clusters with arbitrary shape and good efficiency on large databases. The well-known clustering algorithms offer no solution to ...

被引用次数 : 10010 相关文章 所有 75 个版本 引用 保存 更多



A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise [KDD 1996]

- Claimed that the average run time complexity of DBSCAN is $O(n \cdot \log n)$
- Follow papers mis-claimed that the complexity of DBSCAN is $O(n \cdot \log n)$

2015 SIGMOD Best Paper Award



DBSCAN Revisited: Mis-Claim, Un-Fixability, and Approximation.

SIGMOD Best Paper Award

Recipients of the award are the following:

- 2015 [DBSCAN Revisited: Mis-Claim, Un-Fixability, and Approximation.](#) Yufei Tao, Junhao Gan
- 2014 [Materialization Optimizations for Feature Selection Workloads.](#) Ce Zhang, Arun Kumar, Christopher Ré
- 2013 [Massive Graph Triangulation.](#) Xiaocheng Hu, Yufei Tao, Chin-Wan Chung
- 2012 [High-Pe](#) 2013 SIGMOD Best Paper Award [Processing over XML Streams.](#) Barzan Mozafari, Kai Zeng, Carlo Zaniolo
- 2011 [Entangled Queries: Enabling Declarative Data-Driven Coordination.](#) Nitin Gupta, Lucja Kot, Sudip Roy, Gabriel Bender, Johannes Gehrke, Christoph Koch
- 2010 FAST: Fast Architecture Sensitive Tree Search on Modern CPUs and GPUs. Changkyu Kim, Jatin Chhugani, Nadathur Satish, Eric Sedlar, Anthony Nguyen, Tim Kaldewey, Victor Lee, Scott Brandt, and Pradeep Dubey [[award citation](#)]

SIGMOD/PODS 2015

[Website](#)

[Facebook](#)

RSS

[SIGMOD News](#)

SIGMOD Record

**SIGMOD
RECORD**
Web

2015 SIGMOD Best Paper Award



- DBSCAN Revisited: Mis-Claim, Un-Fixability, and Approximation.
 - For $d \geq 3$, DBSCAN can't have $O(n \cdot \log n)$ time complexity.
 - Prove that the DBSCAN problem requires $\Omega(n^{4/3})$ time to solve in $d \geq 3$.
 - Introduce a new concept called *p-approximate DBSCAN* as an alternative to DBSCAN on large datasets of $d \geq 3$.

2015 SIGMOD Best Paper Award



Yufei Tao



Professor

Room 633, Building 78
[School of Information Technology and Electrical Engineering](#)
[University of Queensland](#)
Brisbane, Queensland 4072
Australia

Tel: +61-7-33658319

Email: taoyf@itee.uq.edu.au

[Publications](#) and [Google Scholar](#)

General

Short Bio

Yufei Tao joined as a Professor the School of Information Technology and Electrical Engineering, the University of Queensland (UQ) in Jan 2016. He obtained his PhD degree from Hong Kong University of Science and Technology (HKUST) in 2002, proudly under the supervision of Prof. Dimitris Papadias. He was a Visiting Scientist at the Carnegie Mellon University during 2002-2003, and then an Assistant Professor at the City University of Hong Kong during 2003-2006, after which he worked till Jan 2016 at the Chinese University of Hong Kong (CUHK), where he was promoted to Professor in 2009. From 2011 to 2013, he was simultaneously a Visiting Professor, under the World Class University program of the Korean government, in the Korea Advanced Institute of Science and Technology (KAIST), Korea. Currently, he also holds an honorary position of Adjunct Professor in the Department of Computer Science and Engineering, CUHK.

He served as an associate editor of *ACM Transactions on Database Systems (TODS)* from 2008 to 2015, and of *IEEE Transactions on Knowledge and Data Engineering (TKDE)* from 2012 to 2014. He served as a PC co-chair of *International Conference on Data Engineering (ICDE)* 2014, and of *International Symposium on Spatial and Temporal Databases (SSTD)* 2011. He gave a keynote speech at *International Conference on Database Theory (ICDT)* 2016.

Major Awards

SIGMOD Best Paper Award 2015
SIGMOD Best Paper Award 2013
Hong Kong Young Scientist Award 2002

怀疑的精神

善于思考，善于发现问题

科学研究的发展是在曲折中不断前进的！



Reference



Ester, Martin, et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." *Kdd*. Vol. 96. No. 34. 1996.

Gan, Junhao, and Yufei Tao. "DBSCAN Revisited: Mis-Claim, Un-Fixability, and Approximation." *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 2015.

如何评价 SIGMOD 2015 最佳论文《DBSCAN Revisited》？

<http://www.zhihu.com/question/31845624>

Comparing K-means and DBSCAN



Both are partitional clustering that assign each object to a single cluster

- K-means: all object
- DBSCAN: discard noise

Type

- K-means: prototype-based notion
- DBSCAN: density-based notion

Both perform poorly with widely differing densities

- K-means: has difficult with non-globular cluster and clusters with different size
- DBSCAN: can handle clusters of different size and shapes, not affected by noise

Comparing K-means and DBSCAN



Definition requirement

- K-means: need well-defined centroid
- DBSCAN: need definition of density

Sparse, high-dimensional data

- K-means: work well, such as document data
- DBSCAN: performs poorly for Euclidean definition of density

Distribution of data

- K-means: spherical Gaussian distributions
- DBSCAN: no assumption

Comparing K-means and DBSCAN



Both using all attributes

Number of clusters

- K-means: user define
- DBSCAN: automatically,

Two parameters: Eps and Minpts

Determination

- K-means: no
- DBSCAN: yes

For cluster analysis, the analogous question is how to evaluate the “goodness” of the resulting clusters?

But “clusters are in the eye of the beholder”! 主观性太强!

Then why do we want to evaluate them?

- To compare clustering algorithms
- To compare two sets of clusters
- To compare two clusters

Different Aspects of Cluster Validation 簇评估



1. Determining the **clustering tendency** 聚类趋势 of a set of data, i.e., distinguishing whether non-random structure actually exists in the data **先看看是不是真的有簇结构存在**
2. Comparing the results of a cluster analysis to externally known information, e.g., to externally given class labels.
3. Evaluating how well the results of a cluster analysis fit the data *without* reference to external information.
 - Use only the data
4. Comparing the results of two different sets of cluster analyses to determine which is better.
5. Determining the 'correct' number of clusters.

For 2, 3, and 4, we can further distinguish whether we want to evaluate the entire clustering or just individual clusters.

Measures of Cluster Validity



Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.

- **External Index:** Used to measure the extent to which cluster labels match externally supplied class labels.
 - ◆ Entropy 熵
- **Internal Index:** Used to measure the goodness of a clustering structure *without* respect to external information.
 - ◆ Sum of Squared Error (SSE)
- **Relative Index:** Used to compare two different clusterings or clusters. 实际上不是一个单独的簇评估类型，而是度量的具体使用
 - ◆ Often an external or internal index is used for this function, e.g., SSE or entropy

Sometimes these are referred to as **criteria** instead of **indices**

- However, sometimes criterion is the general strategy and index is the numerical measure that implements the criterion.

Measuring Cluster Validity Via Correlation



Two matrices

- Proximity Matrix (Similarity/Dissimilarity Matrix)
- “Incidence” Matrix
 - ◆ One row and one column for each data point
 - ◆ An entry is 1 if the associated pair of points belong to the same cluster
 - ◆ An entry is 0 if the associated pair of points belongs to different clusters

Compute the correlation between the two matrices

- Since the matrices are symmetric, only the correlation between $n(n-1) / 2$ entries needs to be calculated.

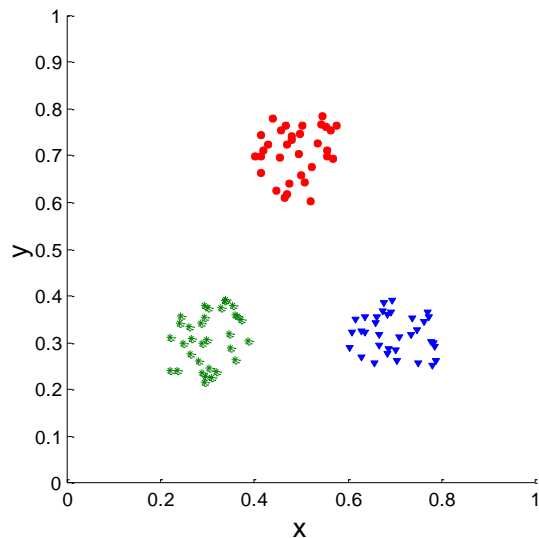
High correlation indicates that points that belong to the same cluster are close to each other.

Not a good measure for some density or contiguity based clusters.

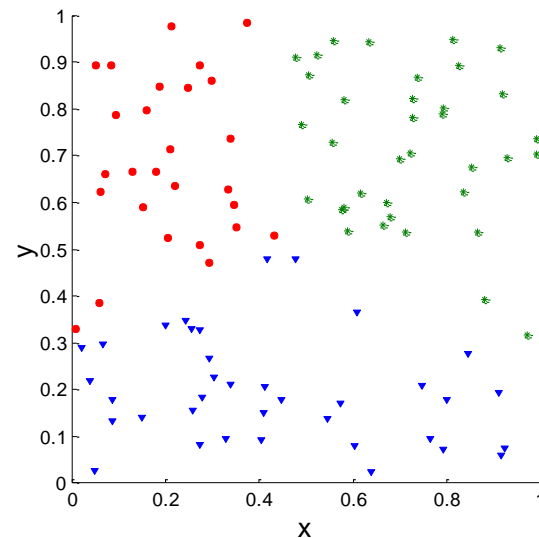
Measuring Cluster Validity Via Correlation



Correlation of incidence and proximity matrices for the K-means clusterings of the following two data sets.



Corr = 0.9235

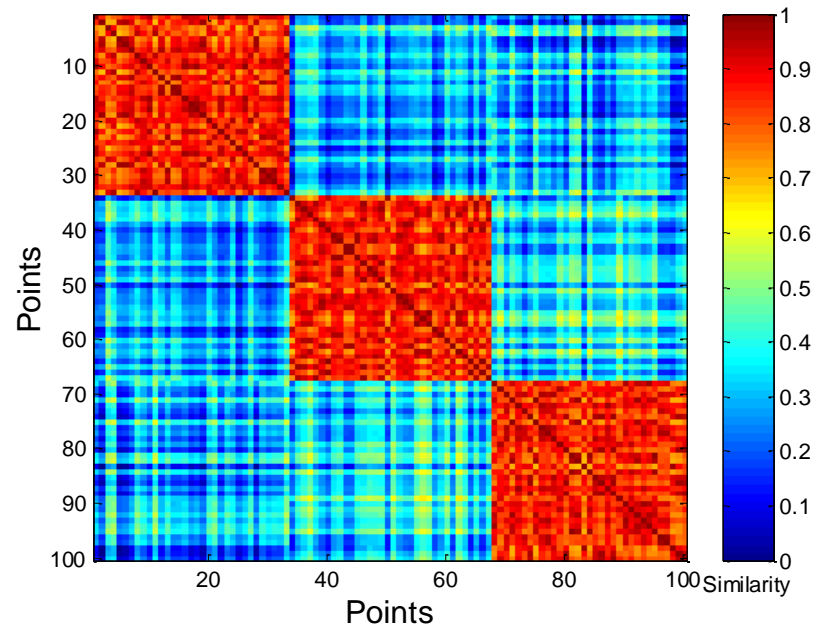
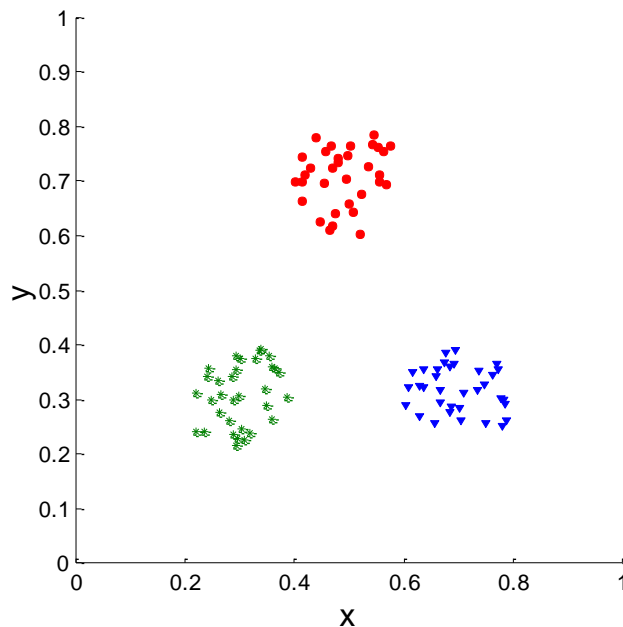


Corr = 0.5810

Using Similarity Matrix for Cluster Validation



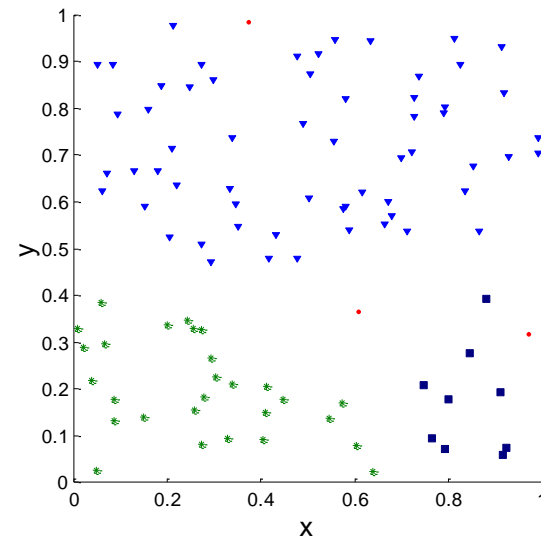
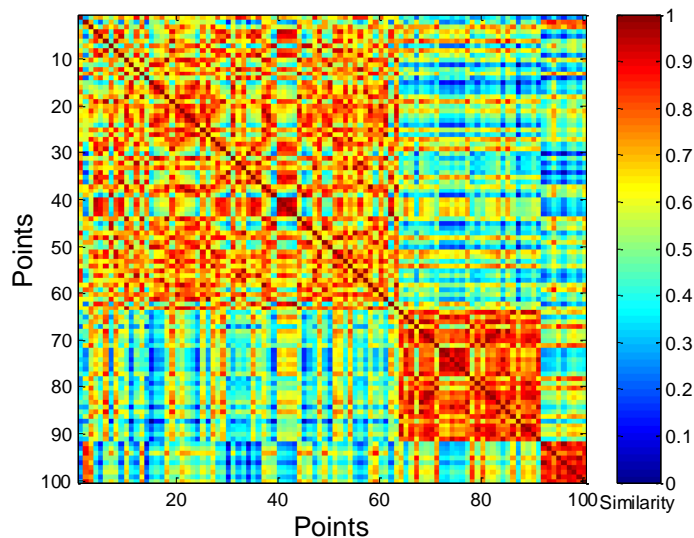
Order the similarity matrix with respect to cluster labels and inspect visually.



Using Similarity Matrix for Cluster Validation



Clusters in random data are not so crisp

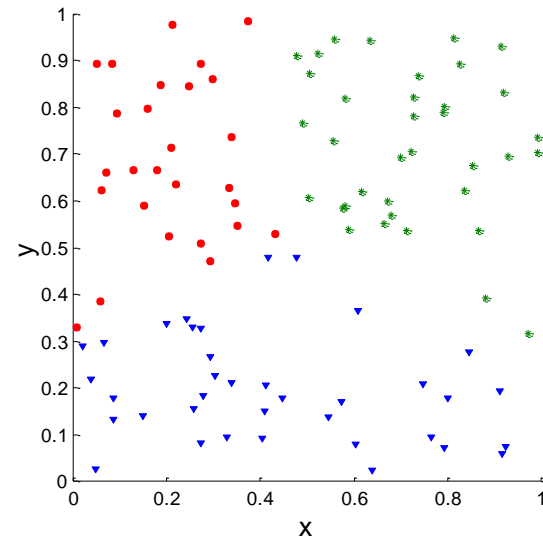
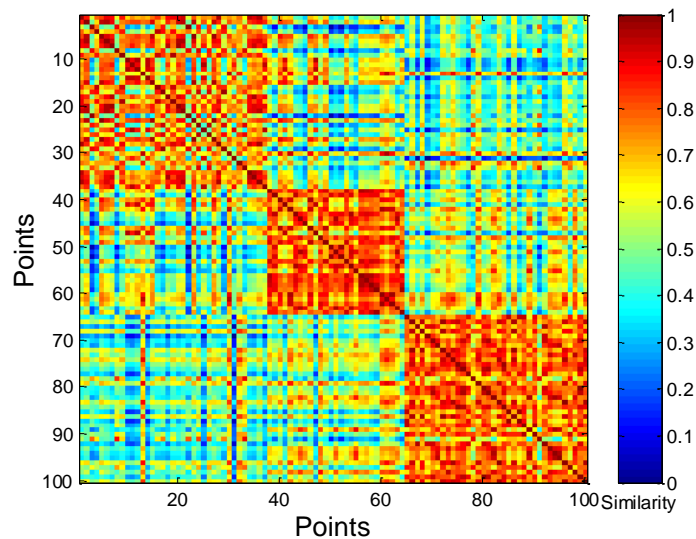


DBSCAN

Using Similarity Matrix for Cluster Validation



Clusters in random data are not so crisp

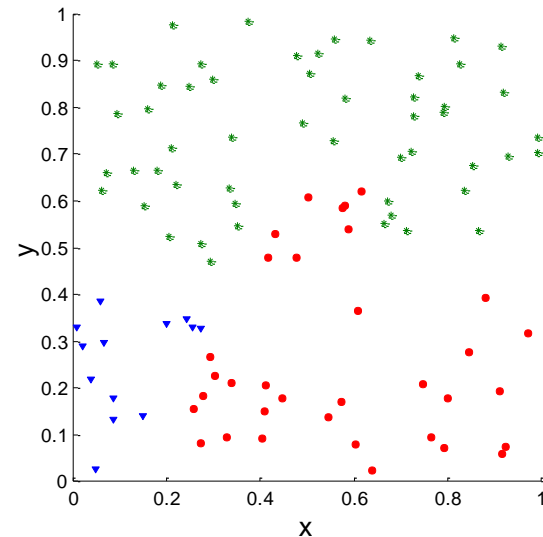
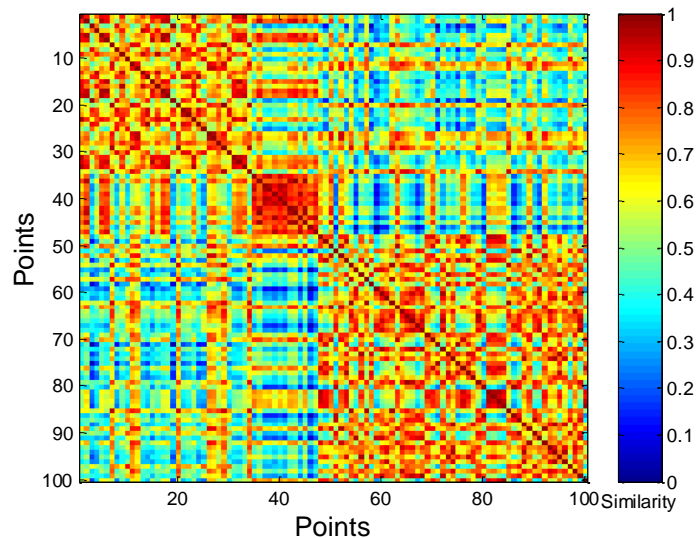


K-means

Using Similarity Matrix for Cluster Validation

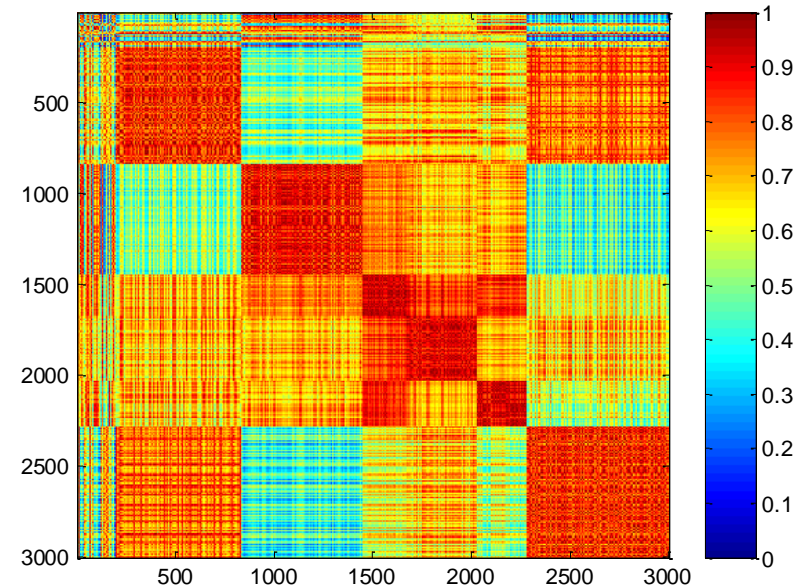
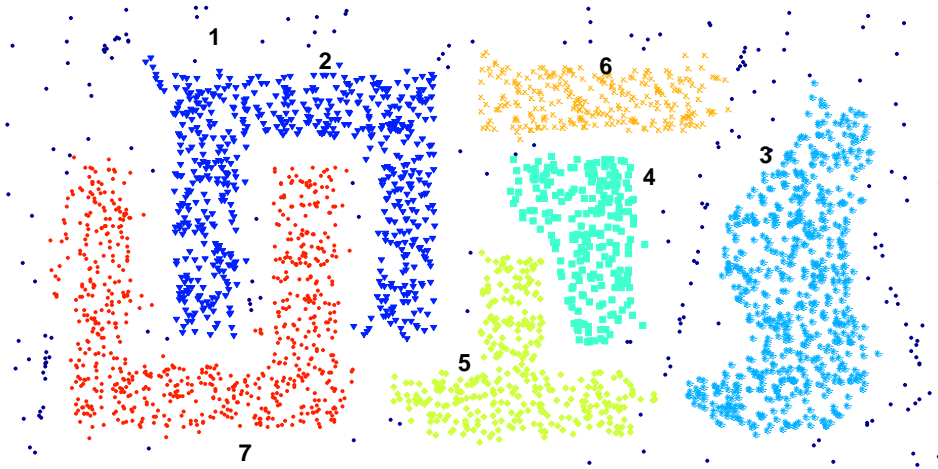


Clusters in random data are not so crisp



Complete linkage hierarchical clustering

Using Similarity Matrix for Cluster Validation



DBSCAN

Internal Measures: SSE



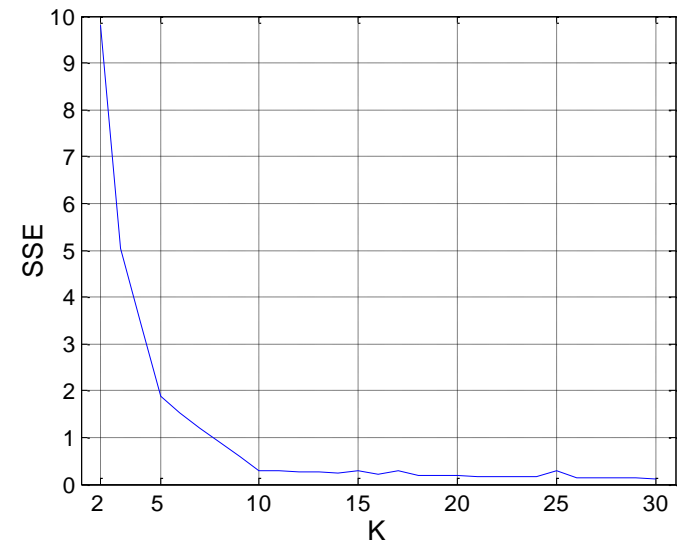
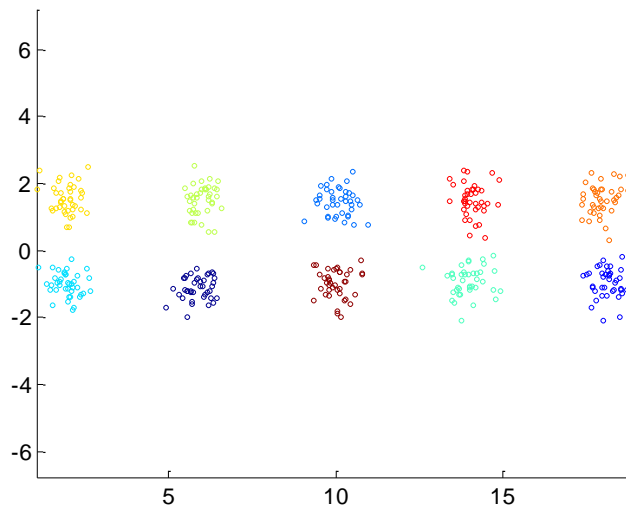
Clusters in more complicated figures aren't well separated

Internal Index: Used to measure the goodness of a clustering structure without respect to external information

- SSE

SSE is good for comparing two clusterings or two clusters (average SSE).

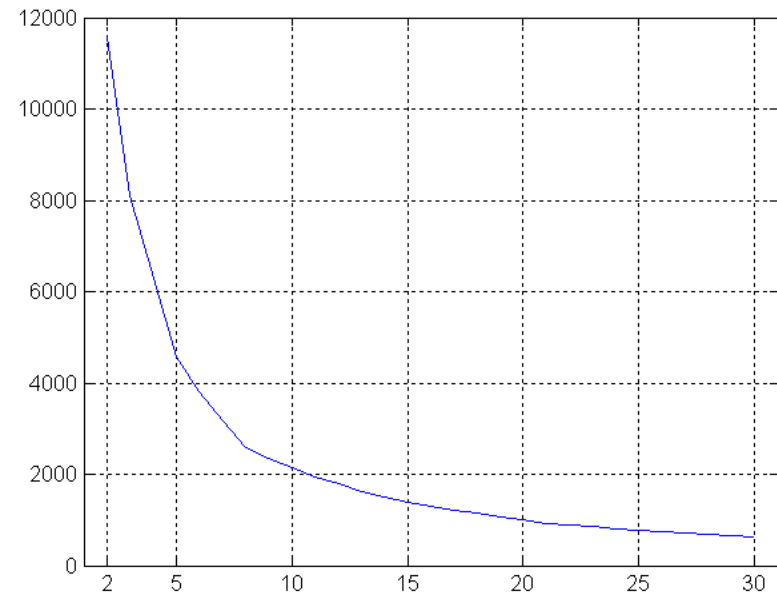
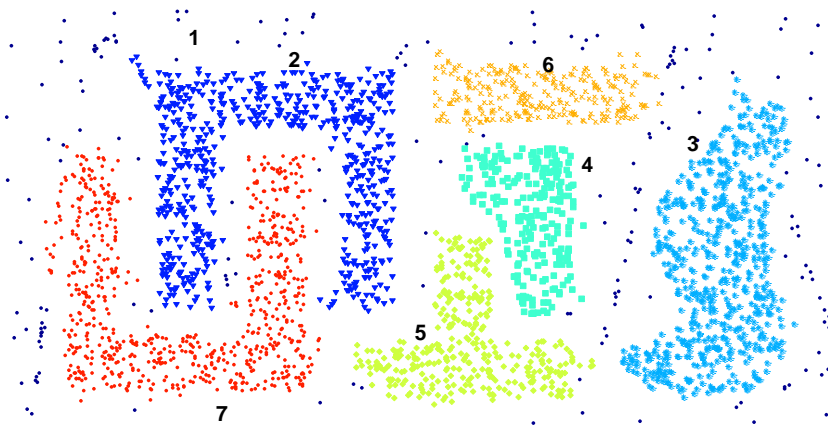
Can also be used to estimate the number of clusters



Internal Measures: SSE



SSE curve for a more complicated data set



SSE of clusters found using K-means

Internal Measures: Cohesion and Separation



Cluster Cohesion 凝聚度: Measures how closely related are objects in a cluster

- Example: SSE

Cluster Separation 分离度: Measure how distinct or well-separated a cluster is from other clusters

Example: Squared Error

- Cohesion is measured by the within cluster sum of squares (SSE)

$$SSE = \sum_i \sum_{x \in C_i} (x - m_i)^2$$

- Separation is measured by the between cluster sum of squares

$$SSB = \sum_i |C_i| (m - m_i)^2$$

- Where $|C_i|$ is the size of cluster i

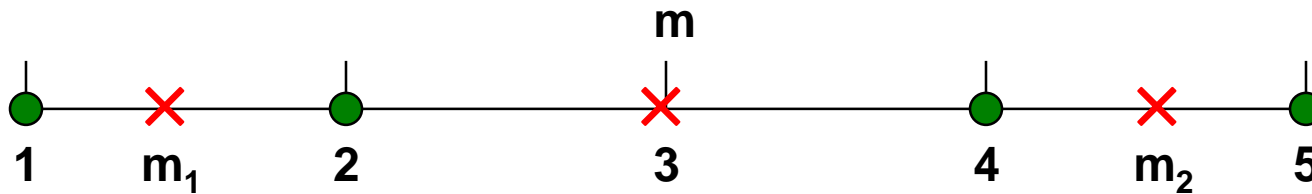
Internal Measures: Cohesion and Separation



Example:

$$TSS = \sum_x (x - m)^2$$

SSE+SSB=TSS **constant**



K=1 cluster:

$$SSE = (1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2 = 10$$

$$SSB = 4 \times (3-3)^2 = 0$$

$$Total = 10 + 0 = 10$$

K=2 clusters:

$$SSE = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$

$$SSB = 2 \times (3-1.5)^2 + 2 \times (4.5-3)^2 = 9$$

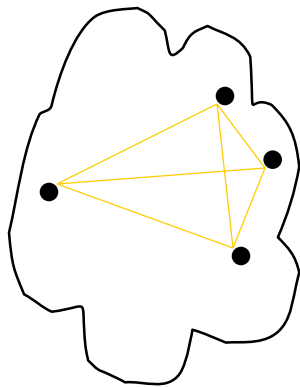
$$Total = 1 + 9 = 10$$

Internal Measures: Cohesion and Separation

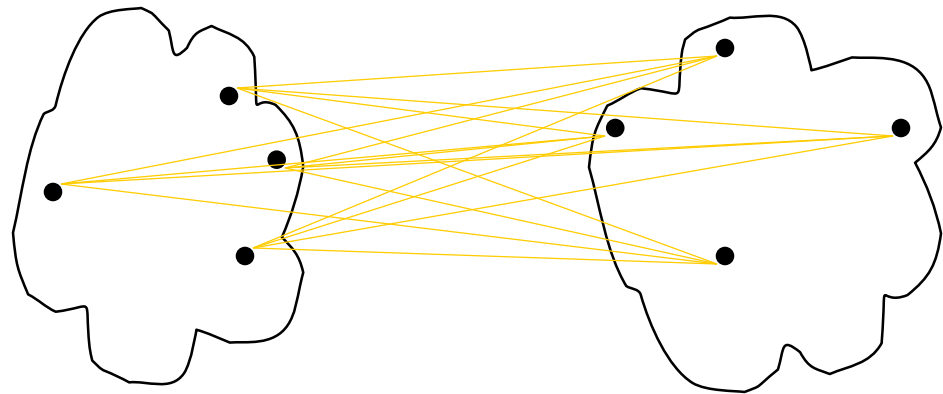


A proximity graph based approach can also be used for cohesion and separation.

- Cluster cohesion is the sum of the weight of all links within a cluster.
- Cluster separation is the sum of the weights between nodes in the cluster and nodes outside the cluster.



cohesion



separation

Final Comment on Cluster Validity



“The validation of clustering structures is the most difficult and frustrating part of cluster analysis. Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage.” 聚类结构的验证是聚类分析中最困难和最令人沮丧的部分。如果不在这方面付出巨大努力，聚类分析将仍然是一门只有那些有经验和极大勇气的真正信徒才能掌握的黑暗魔法。

Algorithms for Clustering Data, Jain and Dubes



Discussion



The entire time I knew Thanos, he only ever had one goal.
To bring balance to the universe by wiping out half of all life.

Q: 灭霸一个响指就能灭掉一半生命，那怎么保证物种均衡平衡嘞~？

A: 聚类分析帮助你！

Method : K-means Clustering

宇宙包含“九大国度”，为防止某个国度灭绝，在每个国度选取一个聚类中心，通过K-means得到聚类簇，再在每个聚类中随机选取一半生物进行消灭，宇宙生物多样性得到进一步保证。



Method : K-means Clustering

