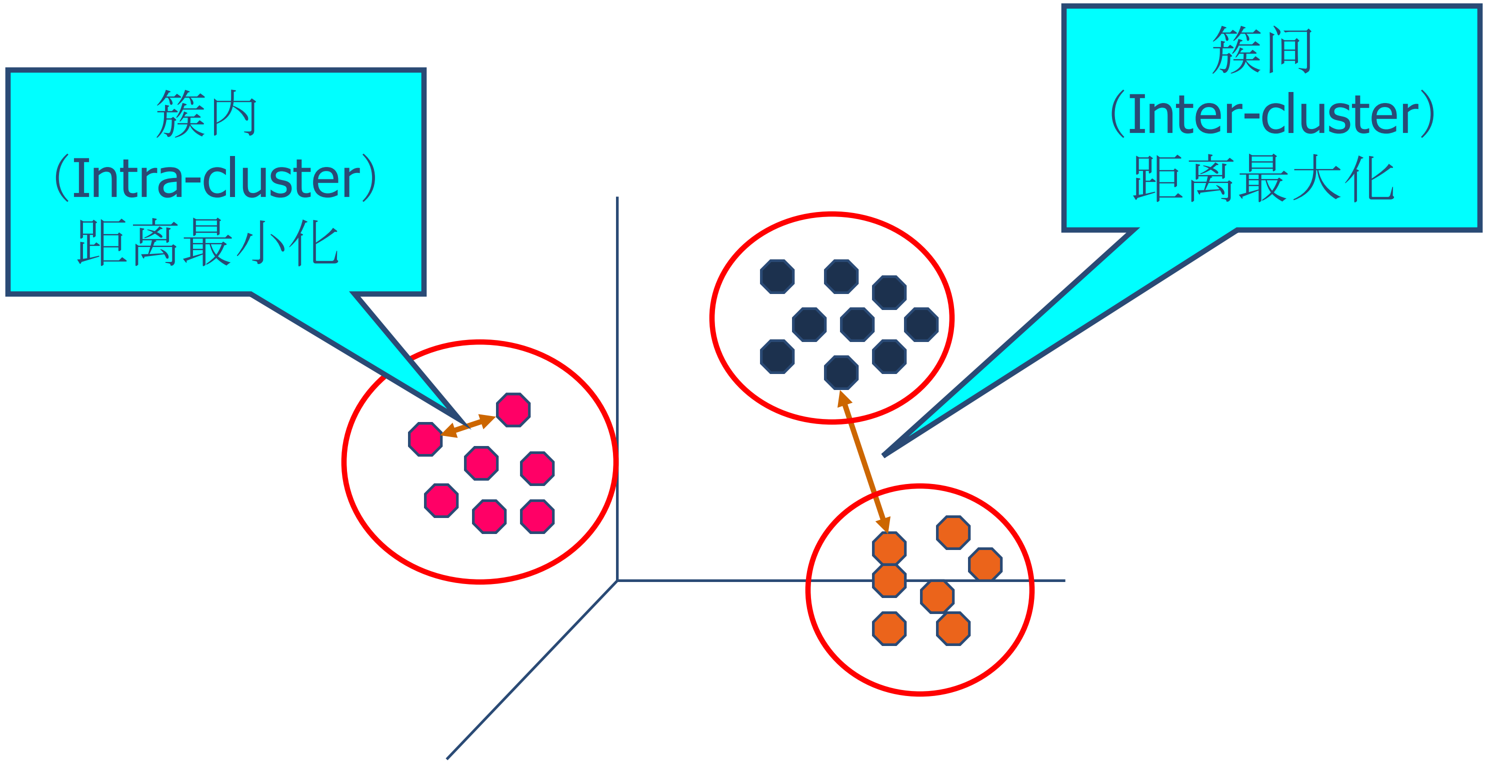
Python与人工智能

聚类-以K-means为例

什么是聚类What is Cluster Analysis?

- 查找对象组，以使一组（group）中的对象彼此相似（similar，或相关related），而与其他组中的对象不同（或不相关）

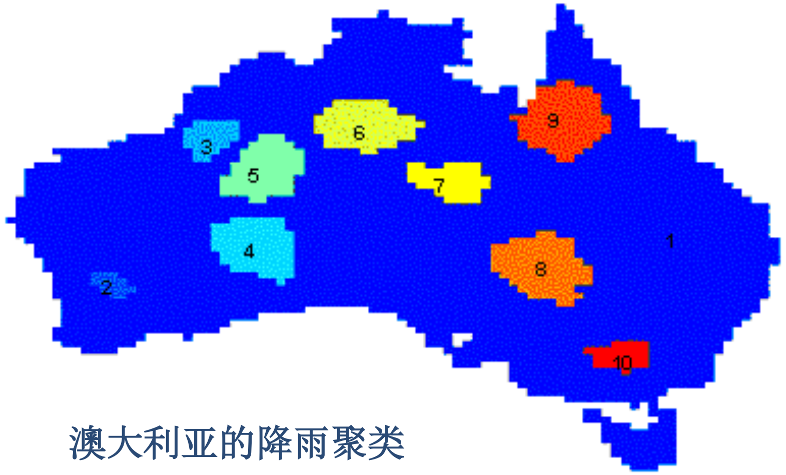


聚类分析应用 Applications of Cluster Analysis

- **理解 Understanding**
  - 将相关的文档分组便于浏览
  - 将具有相似功能的基因和蛋白质分组
  - 将具有相似价格波动的股票分组

	Discovered Clusters	***Industry Group***
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**

- **实用**
  - 汇总 Summarization：减少大型数据集的大小
  - 压缩，例如向量化
  - 有效发现最近邻



澳大利亚的降雨聚类

哪些不是聚类分析 What is not Cluster Analysis?

- 简单分割 Simple segmentation
  - 按姓氏的字母顺序将学生分为不同的注册组
- 查询结果 Results of a query
  - 是外部规范（external specification）的结果
  - 聚分组类是基于数据的对象分组
- 监督分类 Supervised classification
  - 有类别标签信息
- 关联分析
  - Local vs. global connections

单选题 1分

⚙ 设置

下述哪一项**属于**聚类分析？

A 按照学生的学号进行排序

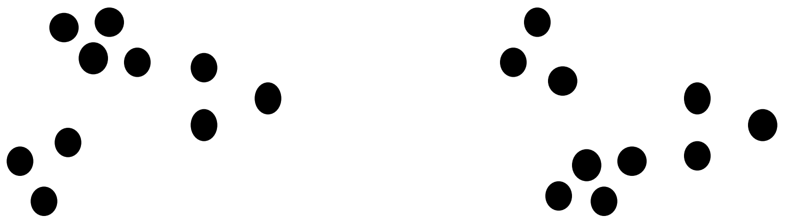
B 根据历史天气记录预测未来的天气状况

C 从数据库中查找指定日期的交易记录

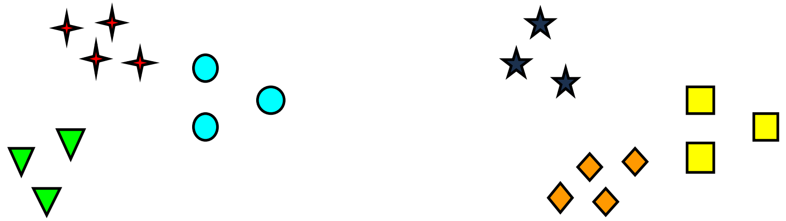
[✓ D] 给定用户的购物记录数据，将用户分组，便于后续的用户购物习惯分析和画像分析

提交

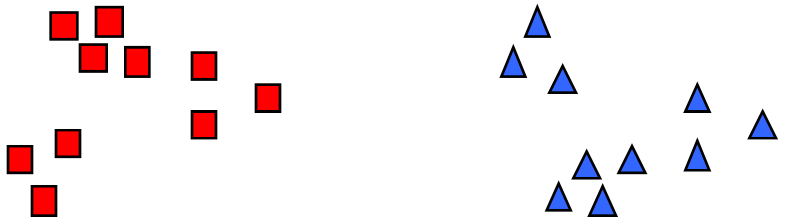
簇的概念可能不明确 Notion of a Cluster can be Ambiguous



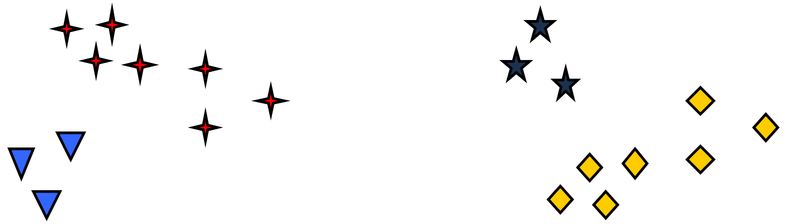
多少簇（cluster）？



Six Clusters



Two Clusters



Four Clusters

聚类的类型 Types of Clusterings

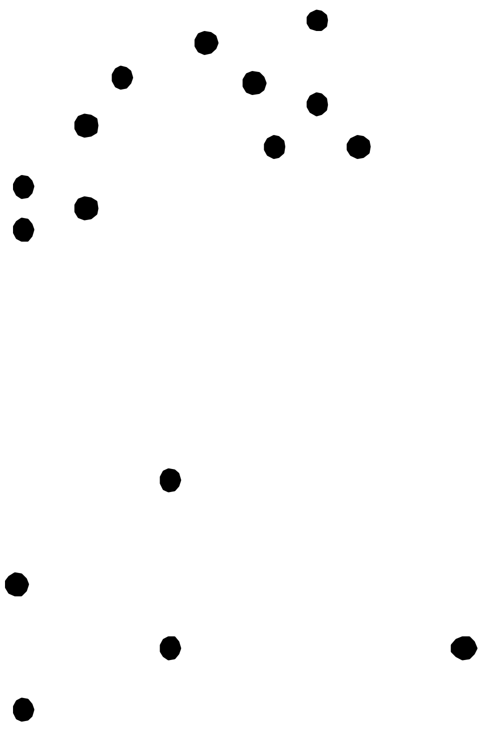
- 整个簇（clustering）的集合被称为聚类（clusters）

- 分层簇集和分区簇集区别很大

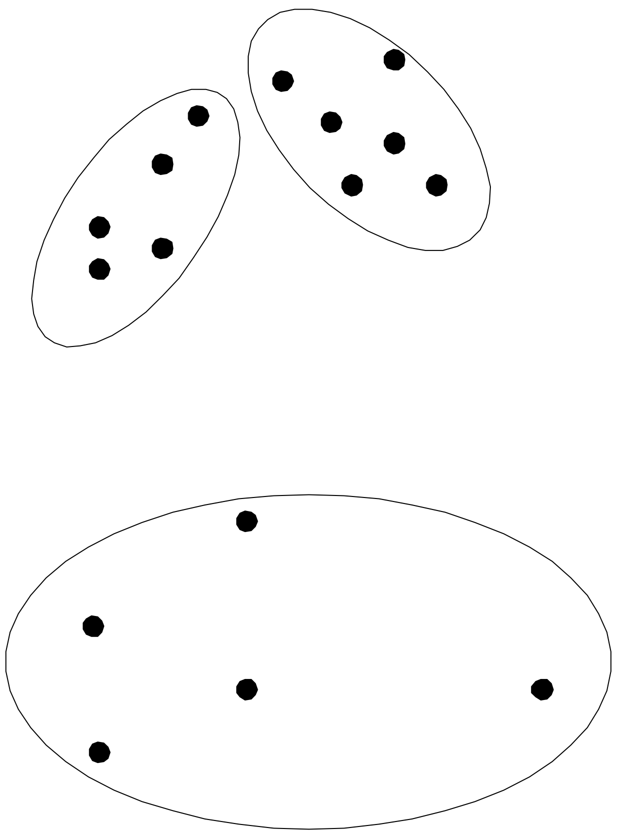
- 划分聚类 Partitional Clustering
  - 将数据对象划分为不重叠的子集（集群），以便每个数据对象恰好在一个子集中

- 层次聚类 Hierarchical clustering
  - 一组嵌套的聚类，组织成一个层次树

划分聚类 Partitional Clustering（非嵌套）

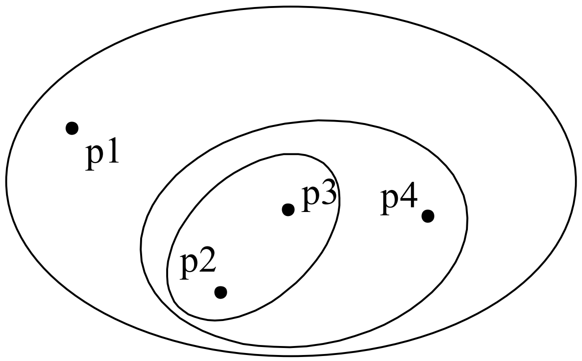


Original Points

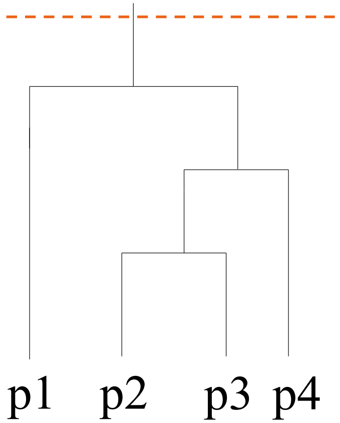


A Partitional Clustering

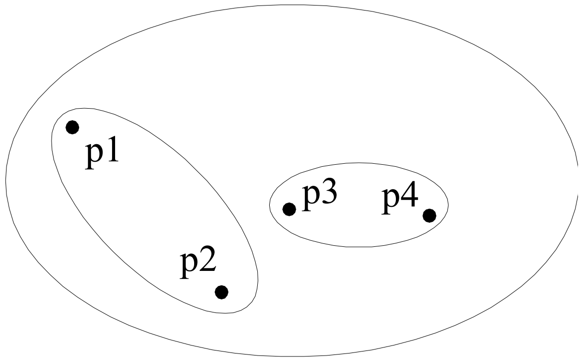
层次聚类 Hierarchical Clustering（嵌套）



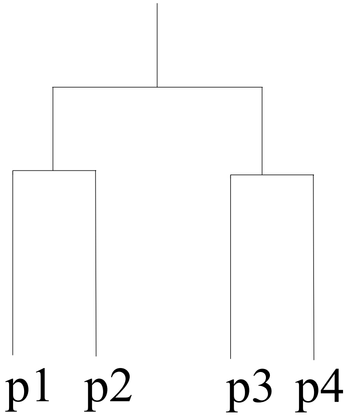
Traditional Hierarchical Clustering



Traditional Dendrogram（树状图）



Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

其它区分标准 Other Distinctions Between Sets of Clusters

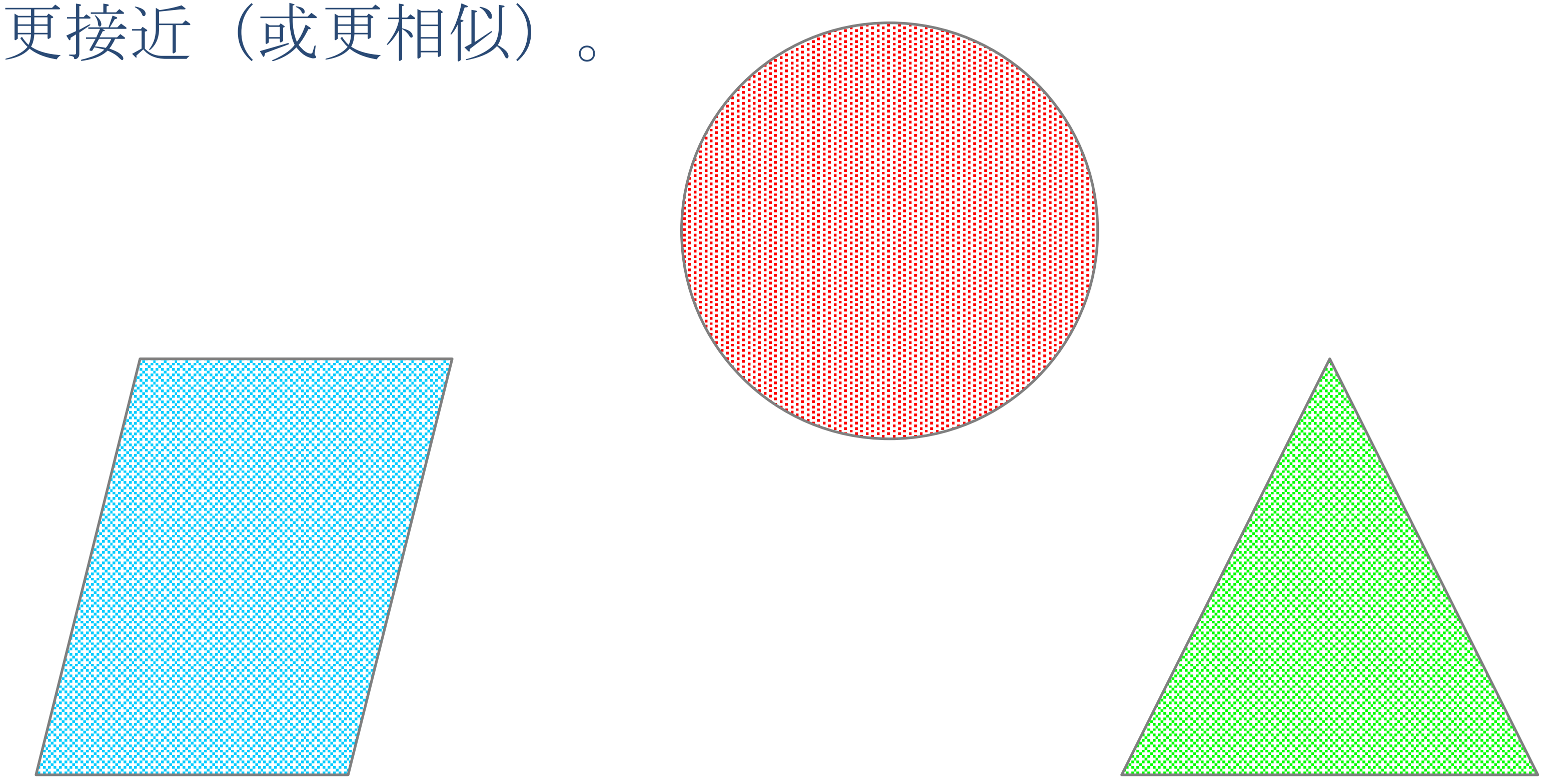
- 排他性与非排他性 Exclusive versus non-exclusive
  - 非排他性聚类中，点可以属于多个簇
  - 可以表示多类别或者边界点（‘border’ points）
- 模糊与非模糊 Fuzzy versus non-fuzzy
  - 在模糊聚类中，每个点以[0,1]的权重（weight）属于每个簇
  - 权重和为1
  - 类似于概率聚类（Probabilistic clustering）
- 部分与完整 Partial versus complete
  - In some cases, we only want to cluster some of the data
- 异构与同构 Heterogeneous versus homogeneous
  - Clusters of widely different sizes, shapes, and densities

聚类类型 Types of Clusters

- 明显分离的簇 Well-separated clusters
  - 其中每个对象到同簇中每个对象的距离比到不同簇中任意对象的距离都近（或更加相似）；任意形状
- 基于原型的簇（基于中心的簇） Center-based clusters
  - 每个对象到定义该簇的原型（中心）的距离比到其他簇的原型的距离更近；球状
- 连续/邻近簇 Contiguous clusters
- 基于密度的簇Density-based clusters
  - 簇是对象的稠密区域，被低密度的区域环绕。
- 属性或概念 Property or Conceptual
  - 具有共同性质的（概念簇）
- 由目标函数描述 Described by an Objective Function

明显分离的簇 Types of Clusters: Well-Separated

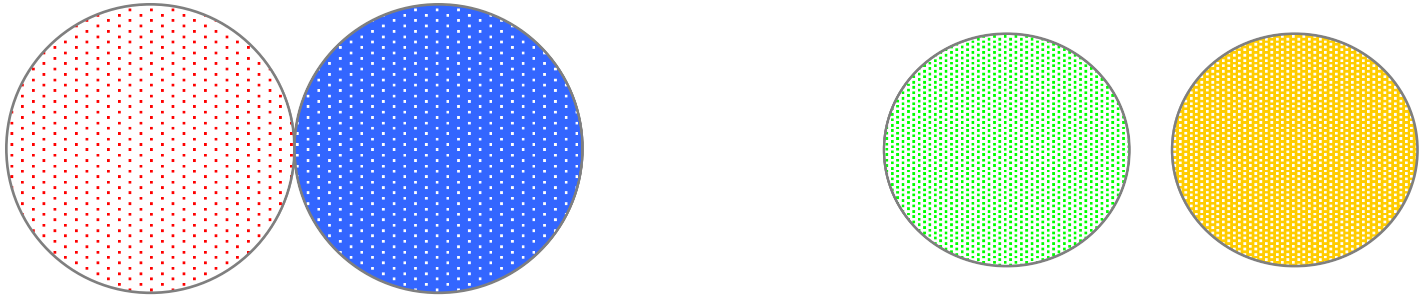
- Well-Separated Clusters:
  - 簇（cluster）是一组点（point），且此簇中的任何点比簇外的任何点更接近（或更相似）。



(a) 3 well-separated clusters

基于原型/中心 Types of Clusters: Center-Based

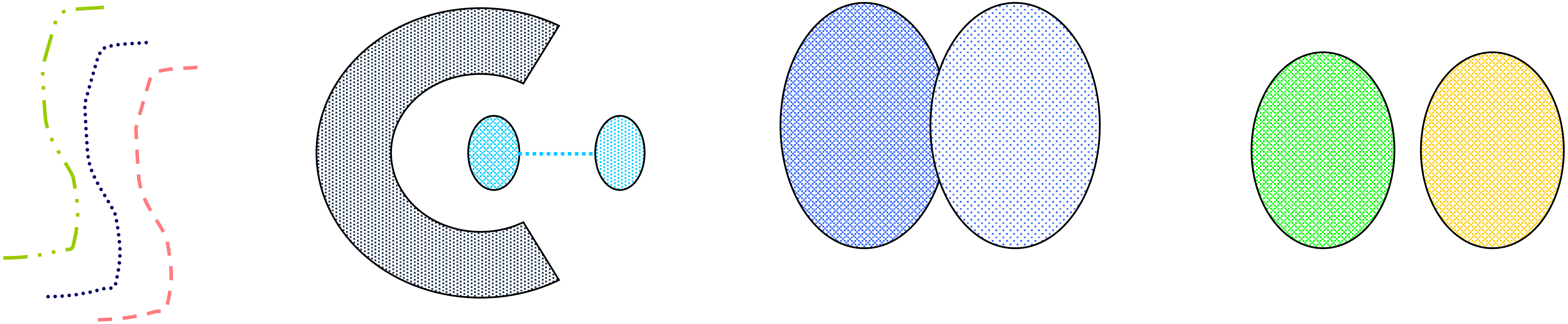
- Center-based
  - 簇是一组对象，相比于簇外的对象/点，簇中的对象都更接近（更类似于）该簇的“**中心**”
  - 群集的中心通常是质心/形心（centroid），即群集中所有点的平均值或质心，是群集中最具“代表性”的点



(b) 4 center-based clusters

连续簇 Types of Clusters: Contiguity-Based

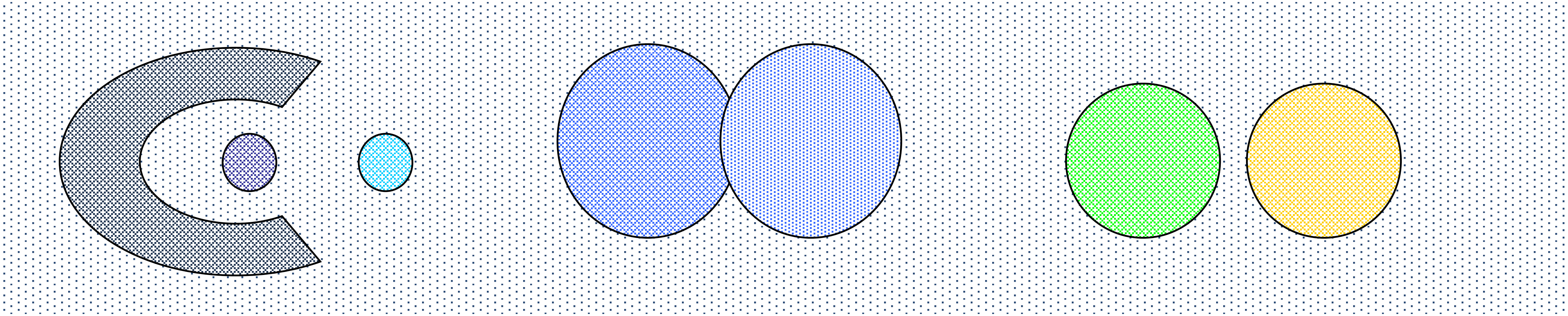
- 连续/邻近簇Contiguous Cluster (Nearest neighbor or Transitive)
  - 基于邻近的簇。每个点到该簇中**至少一个点**的距离比到不同簇中任意点的距离更近
  - A cluster is a set of points such that a point in a cluster is closer (or more similar) to **one or more other points** in the cluster than to any point not in the cluster.



(c) 8 contiguous clusters

基于密度的簇 Types of Clusters: Density-Based

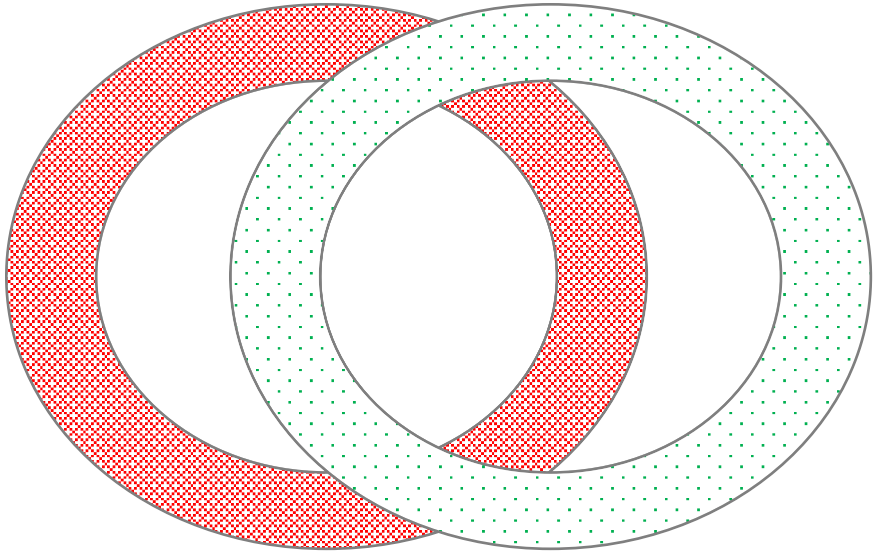
- 基于密度的 Density-based
  - 基于密度的簇。簇是被低密度区域分开的高密度区域。A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
  - 当簇不规则或互相盘绕，并且有噪声和离群点时，常常使用基于密度的簇定义。Used when the clusters are irregular or intertwined, and when noise and outliers are present.



(d) 6 density-based clusters

概念簇 Types of Clusters: Conceptual Clusters

- 共同性质的（概念簇）Shared Property (Conceptual Clusters)
  - 查找具有某些共有属性或表示特定概念的簇。Finds clusters that share some common property or represent a particular concept.



(e) 2 Overlapping Circles

聚类算法 Clustering Algorithms

- **K均值及其变体 K-means and its variants（课程讲解）**

- **层次聚类 Hierarchical clustering（自行了解）**

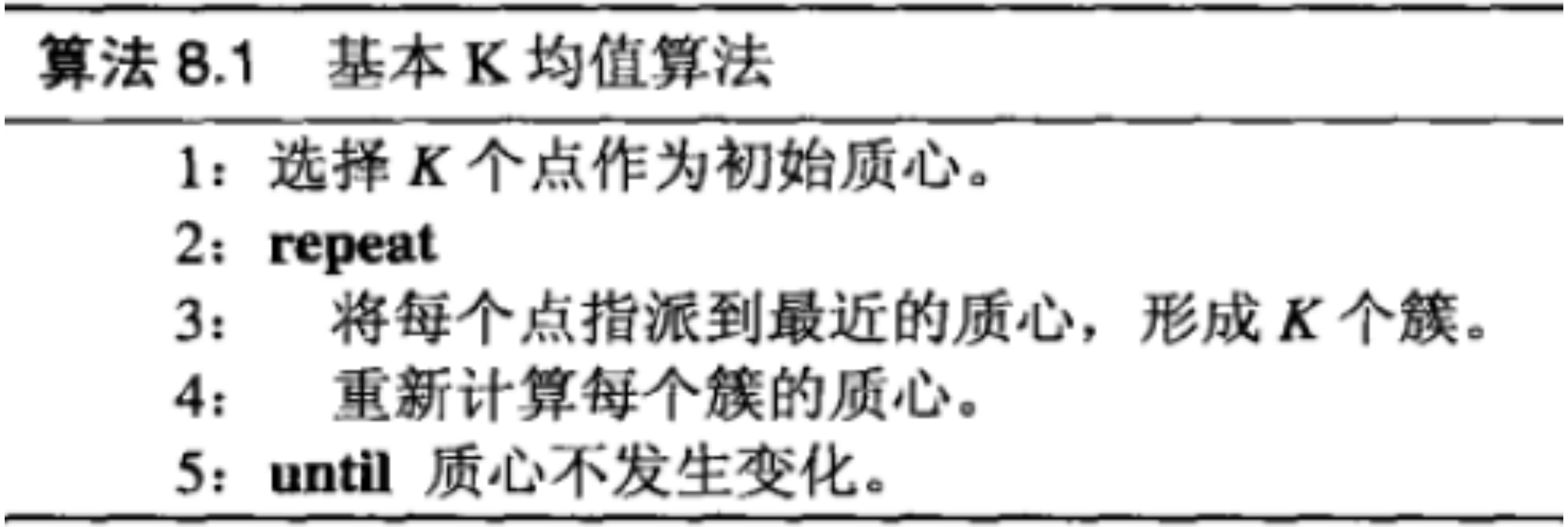
- **基于密度的聚类 Density-based clustering（自行了解）**

K均值聚类 K-means Clustering

- 基于原型的聚类技术
  - K均值
  - K中心点

K-means Clustering

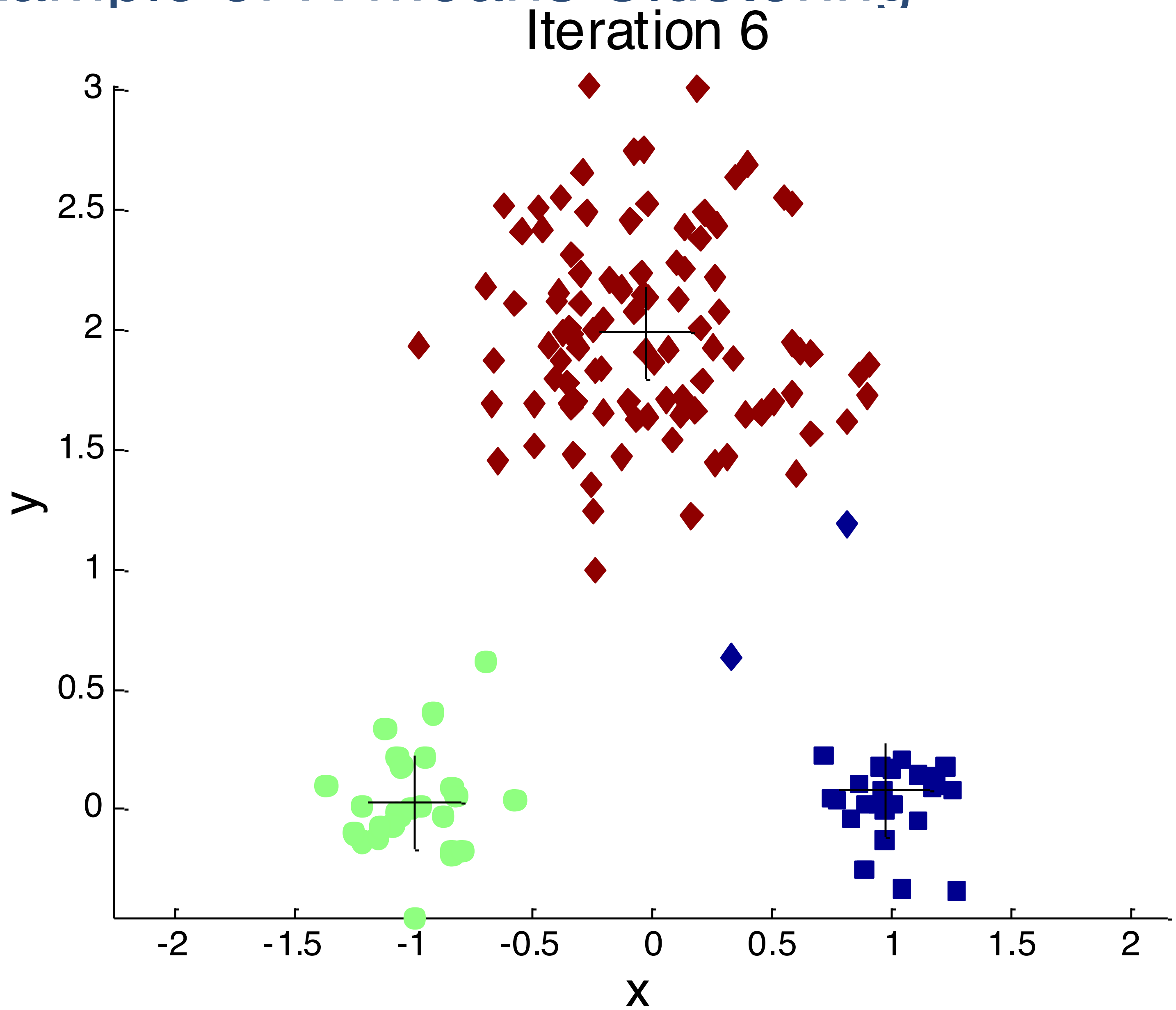
- 划分聚类（Partitional clustering）方法
- 必须指定簇的数目K
- 每个簇与一个质心/中心点（centroid /center point ）相关联
- 每个点都指派给与其最接近的质心对应的簇
- The basic algorithm is very simple



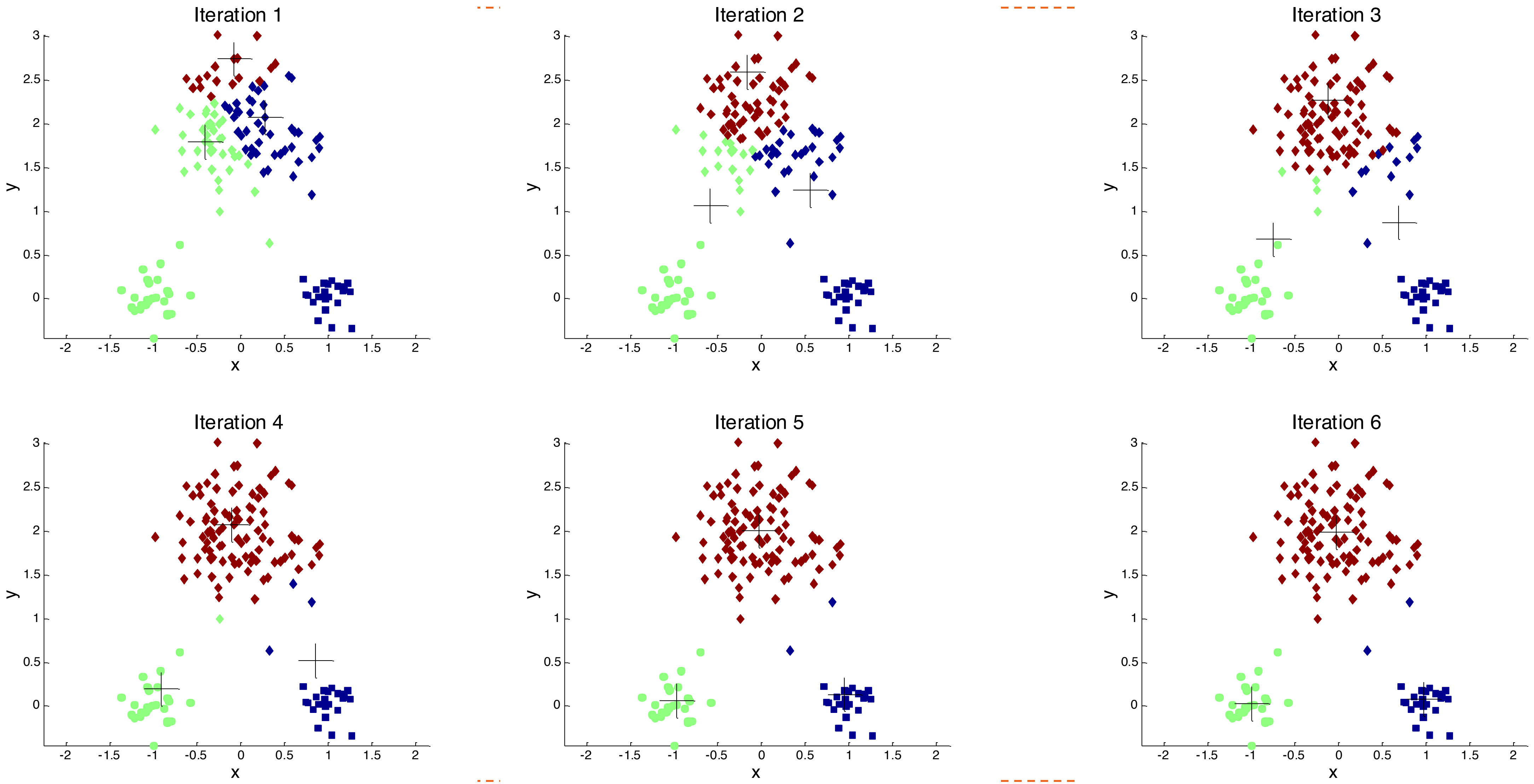
算法 8.1　基本 K 均值算法

```
选择 K 个点作为初始质心。
repeat
    将每个点指派到最近的质心，形成 K 个簇。
    重新计算每个簇的质心。
until 质心不发生变化。
```

示例：Example of K-means Clustering

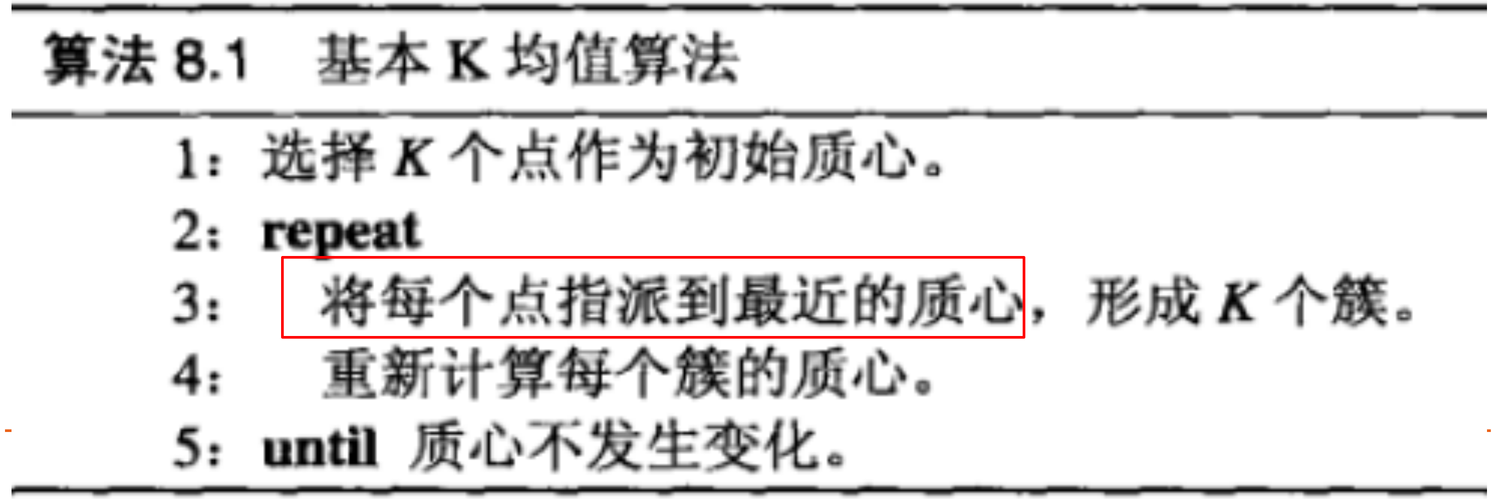


示例：Example of K-means Clustering



K均值聚类：1. 指派点到最近的质心

- 最近？
  - 邻近度度量策略，例如欧氏距离、余弦相似度
- 效率
  - 相似度遍历计算
  - 二分K均值



算法 8.1　基本 K 均值算法

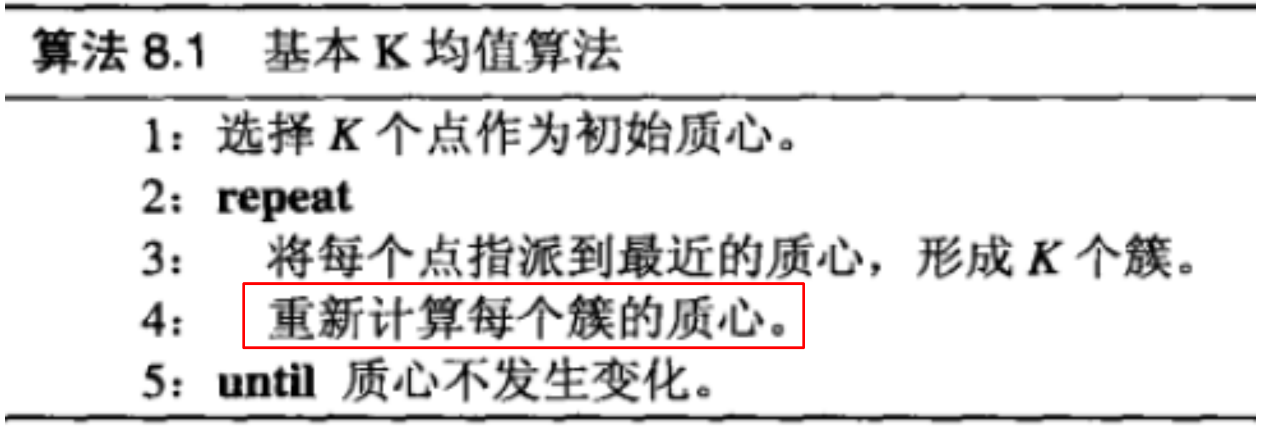
```
选择 $K$ 个点作为初始质心。
repeat
    将每个点指派到最近的质心，形成 $K$ 个簇。
    重新计算每个簇的质心。
until 质心不发生变化。
```

K均值聚类：2. 质心和目标函数

- 如何得到质心？
  - 聚类的目标函数（重新计算质心的标准）
- 欧式空间中的数据
  - 误差的平方和（Sum of Squared Error, **SSE**）
  - 均值！

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

- 增加K可以减小SSE，可取吗？
  - A good clustering with smaller K can have a lower SSE than a poor clustering with higher K

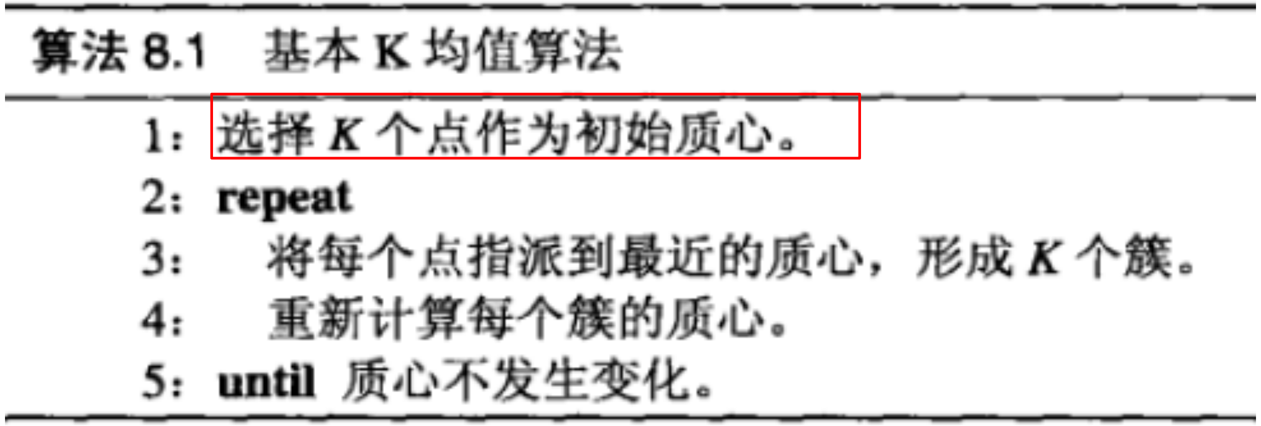


算法 8.1　基本 K 均值算法

```
选择 $K$ 个点作为初始质心。
repeat
    将每个点指派到最近的质心，形成 $K$ 个簇。
    重新计算每个簇的质心。
until 质心不发生变化。
```

K均值聚类：3. 选择初始质心

- 随机选取？

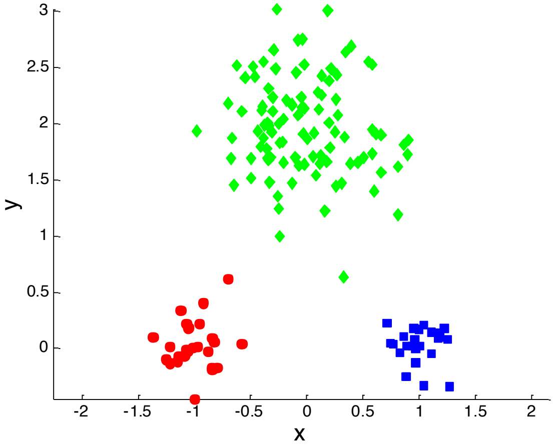


算法 8.1　基本 K 均值算法

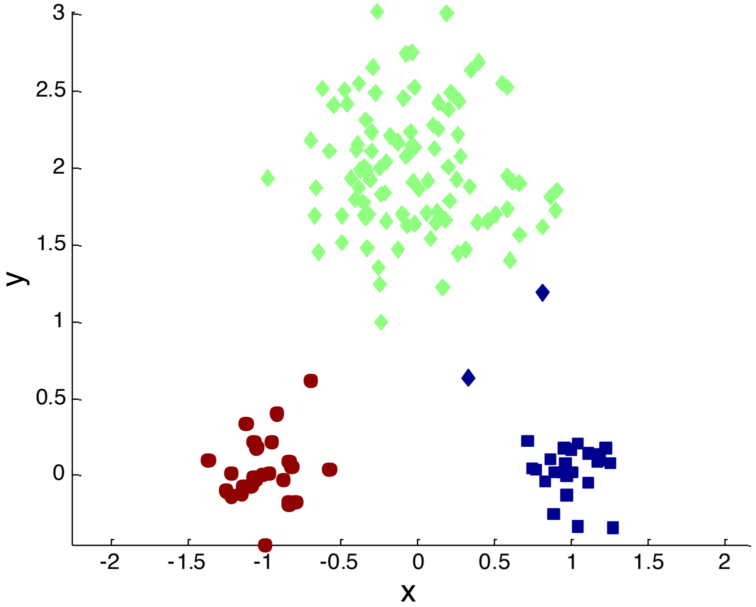
```
选择 K 个点作为初始质心。
repeat
    将每个点指派到最近的质心，形成 K 个簇。
    重新计算每个簇的质心。
until 质心不发生变化。
```

K均值聚类：3. 选择初始质心

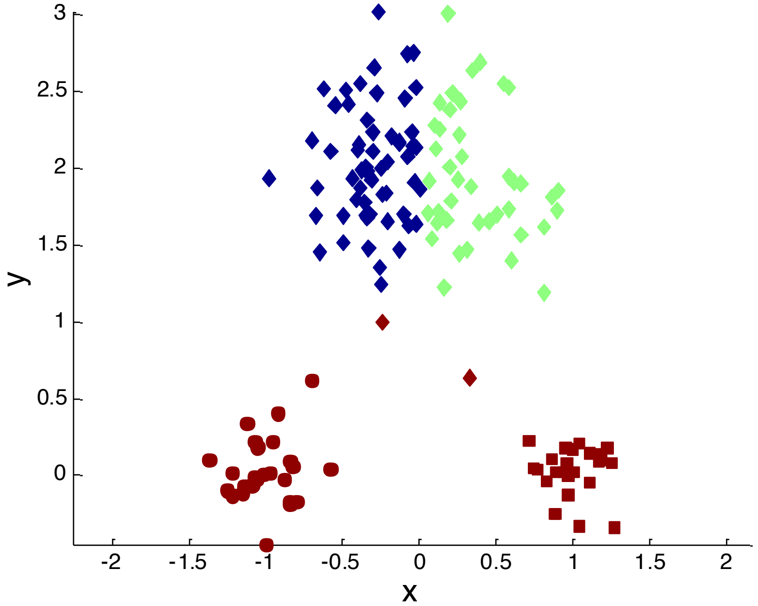
Original Points



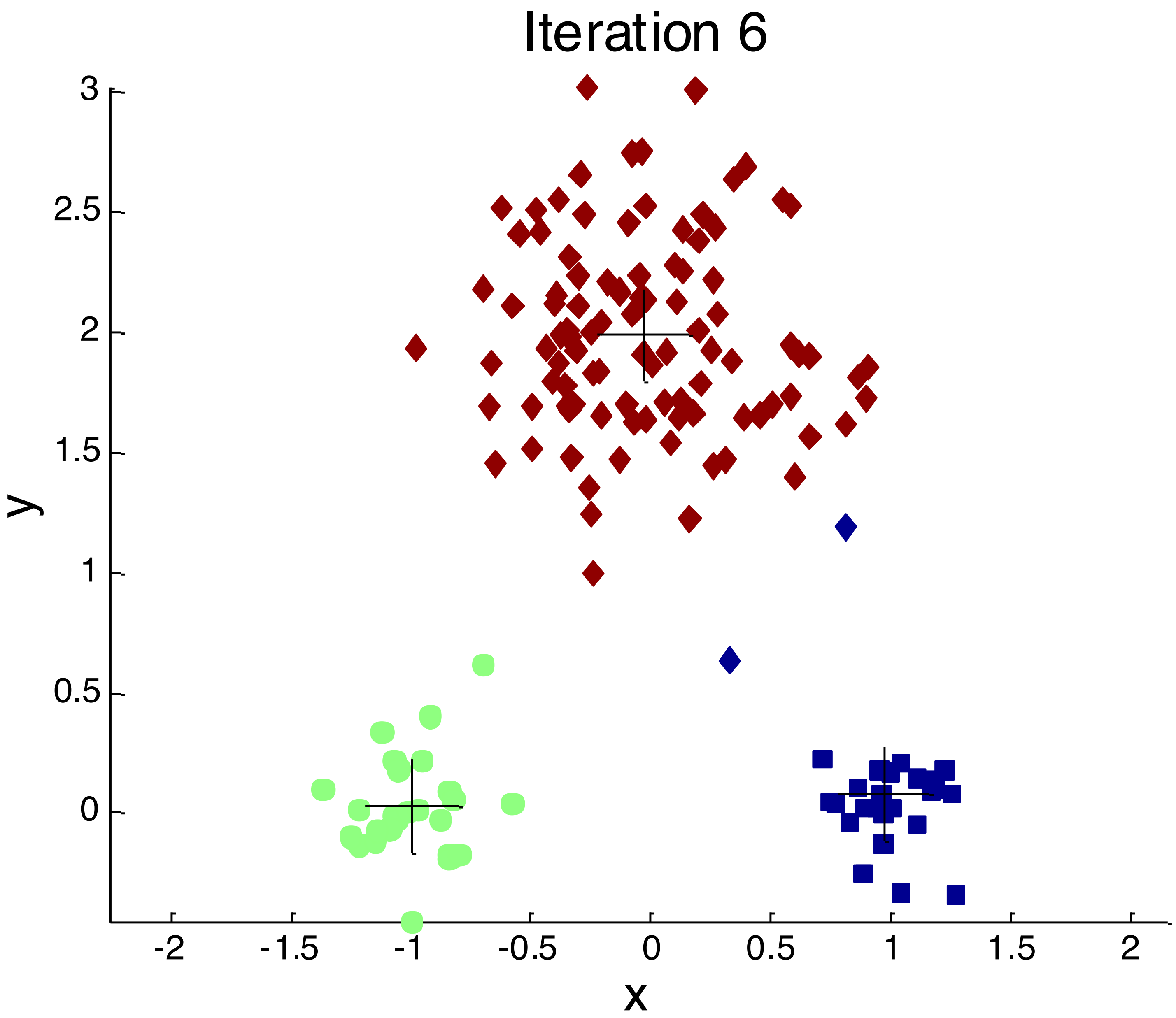
Optimal Clustering



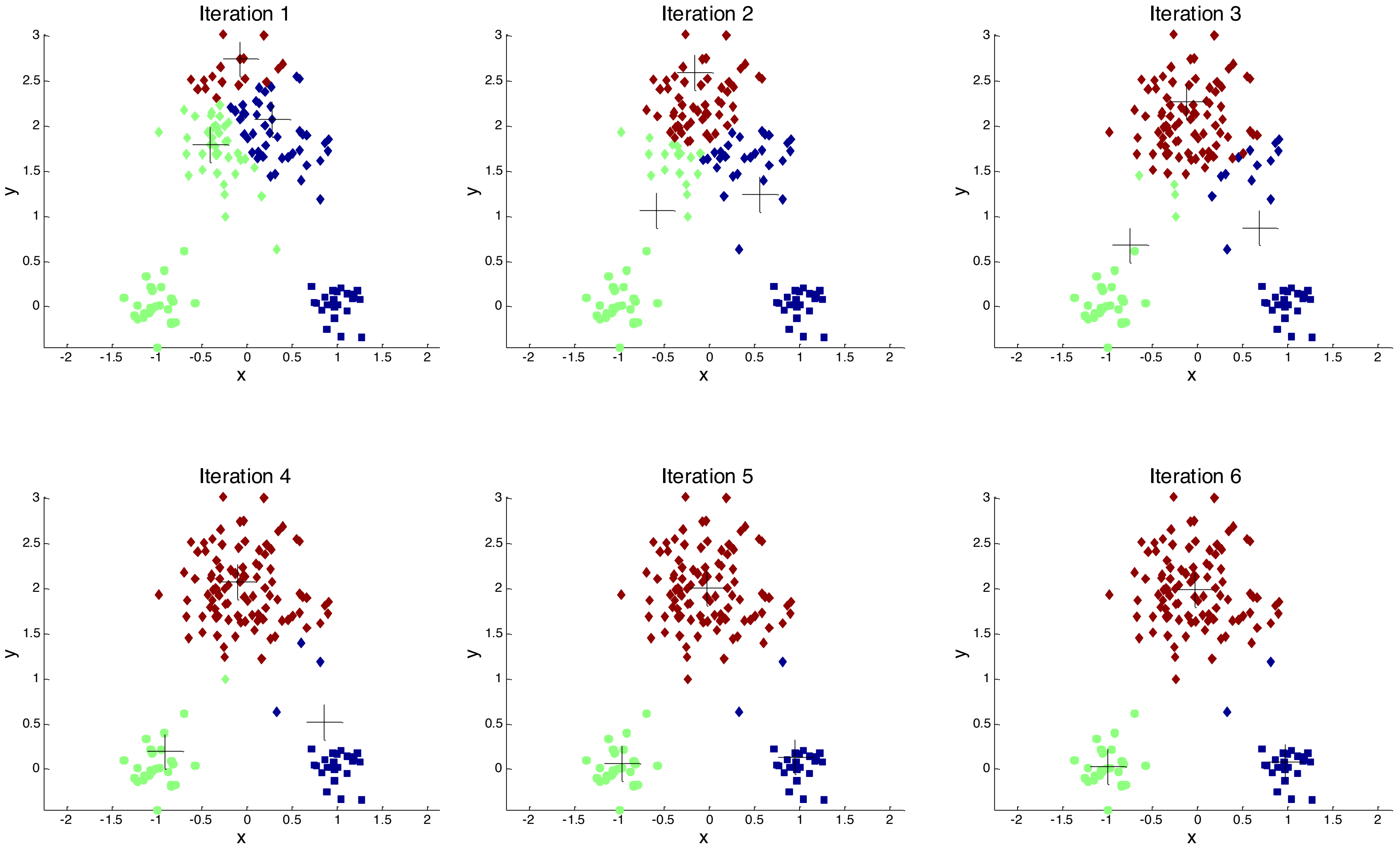
Sub-optimal Clustering



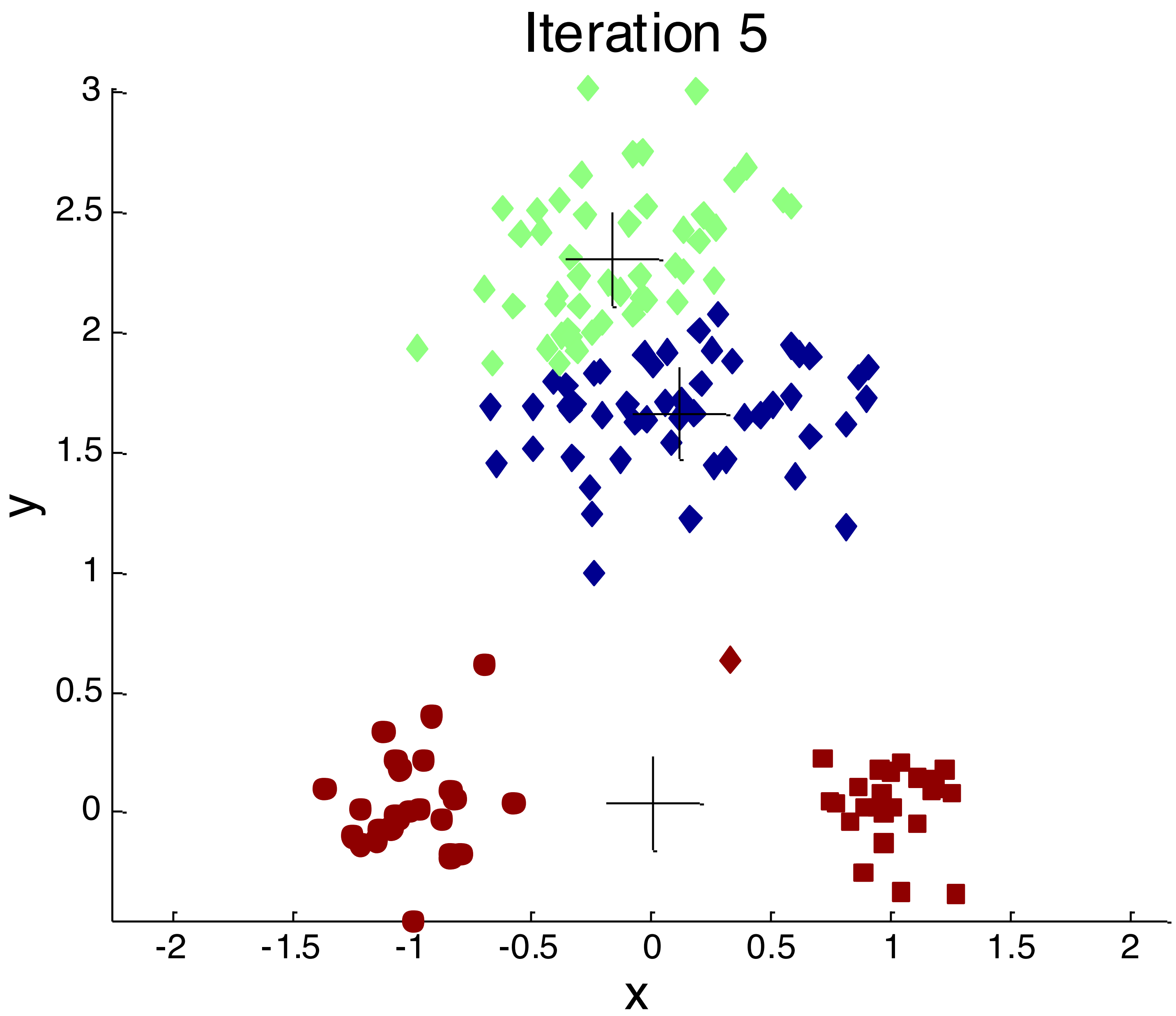
初始质心选择 Importance of Choosing Initial Centroids



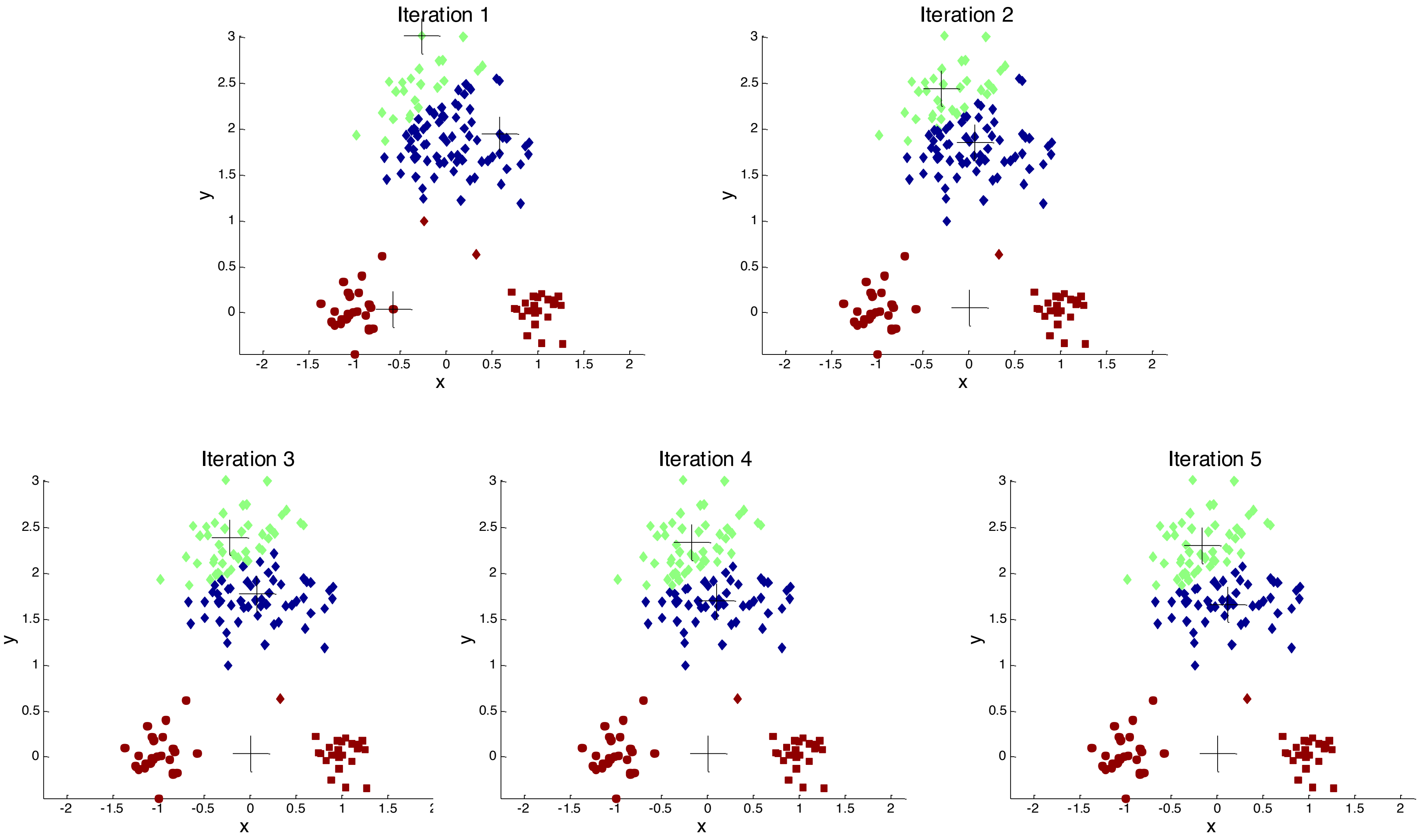
初始质心选择 Importance of Choosing Initial Centroids



初始质心选择 Importance of Choosing Initial Centroids



初始质心选择 Importance of Choosing Initial Centroids



Problems with Selecting Initial Points

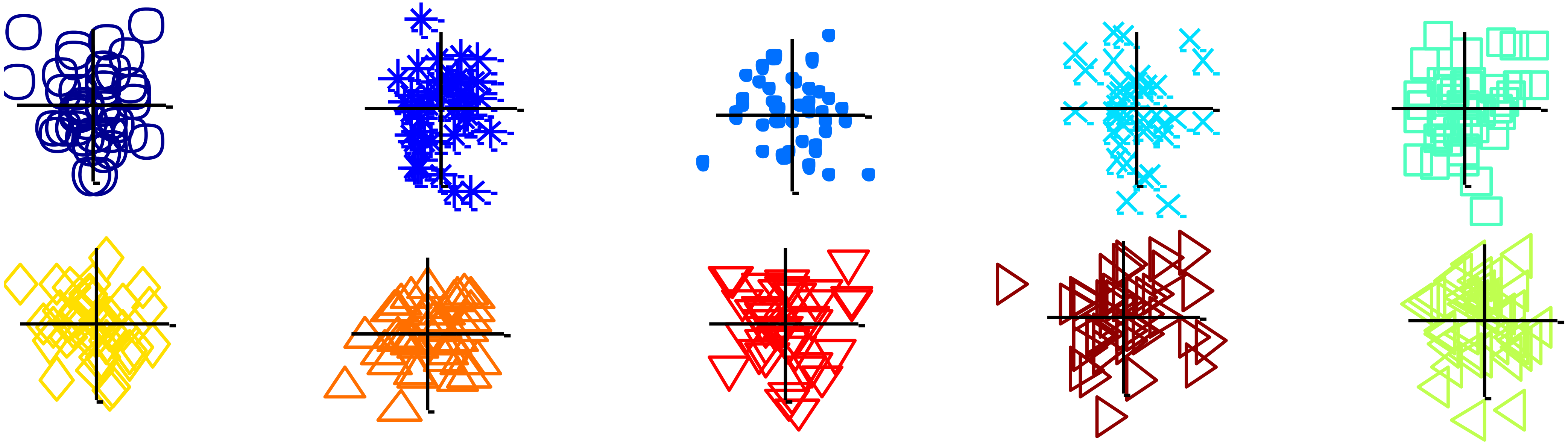
- 如果有K个“真实”簇，则从每个簇中选择一个质心的概率很小
  - 这个概率会随着K的增加而降低
  - 如果簇的大小都为n，则

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

  - 例如, 如果 K = 10，那么概率为 = $10!/10^{10} = 0.00036$
  - 有时最初的质心会以“正确”的方式重新调整自身，有时却不会

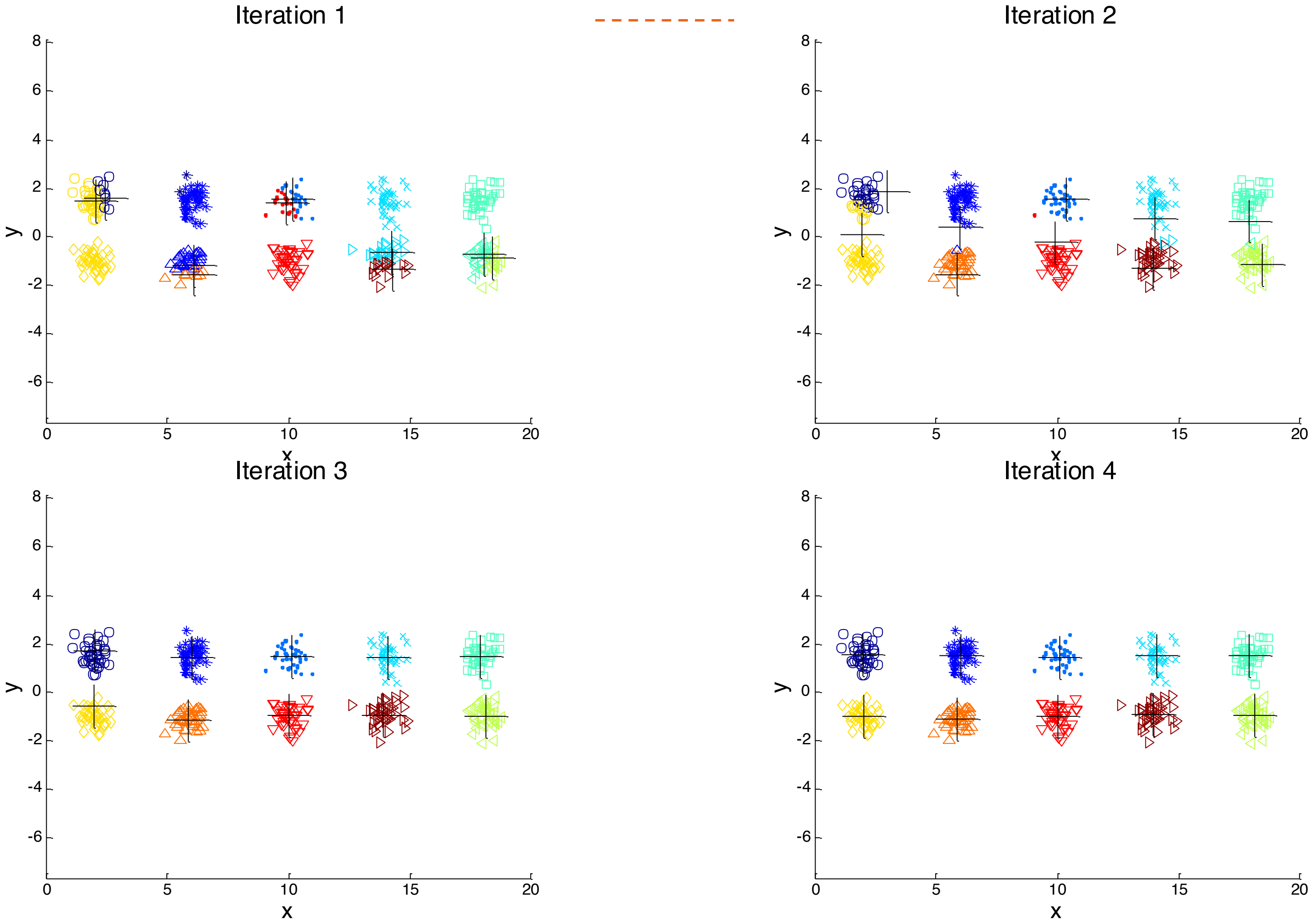
  - 考虑五对簇（five pairs of clusters）的例子

10 Clusters Example



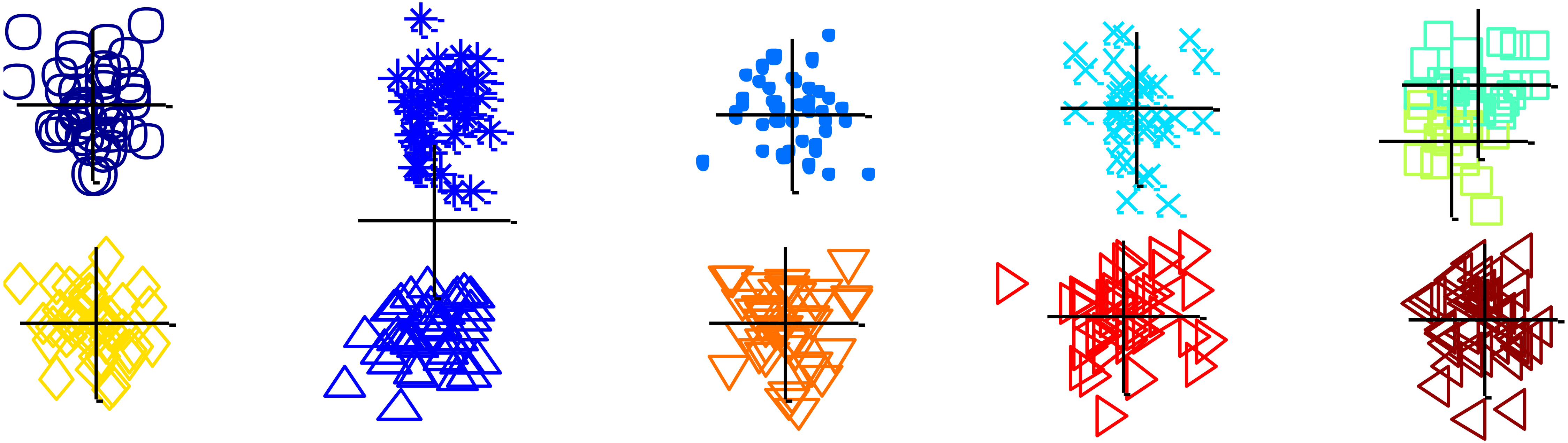
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example



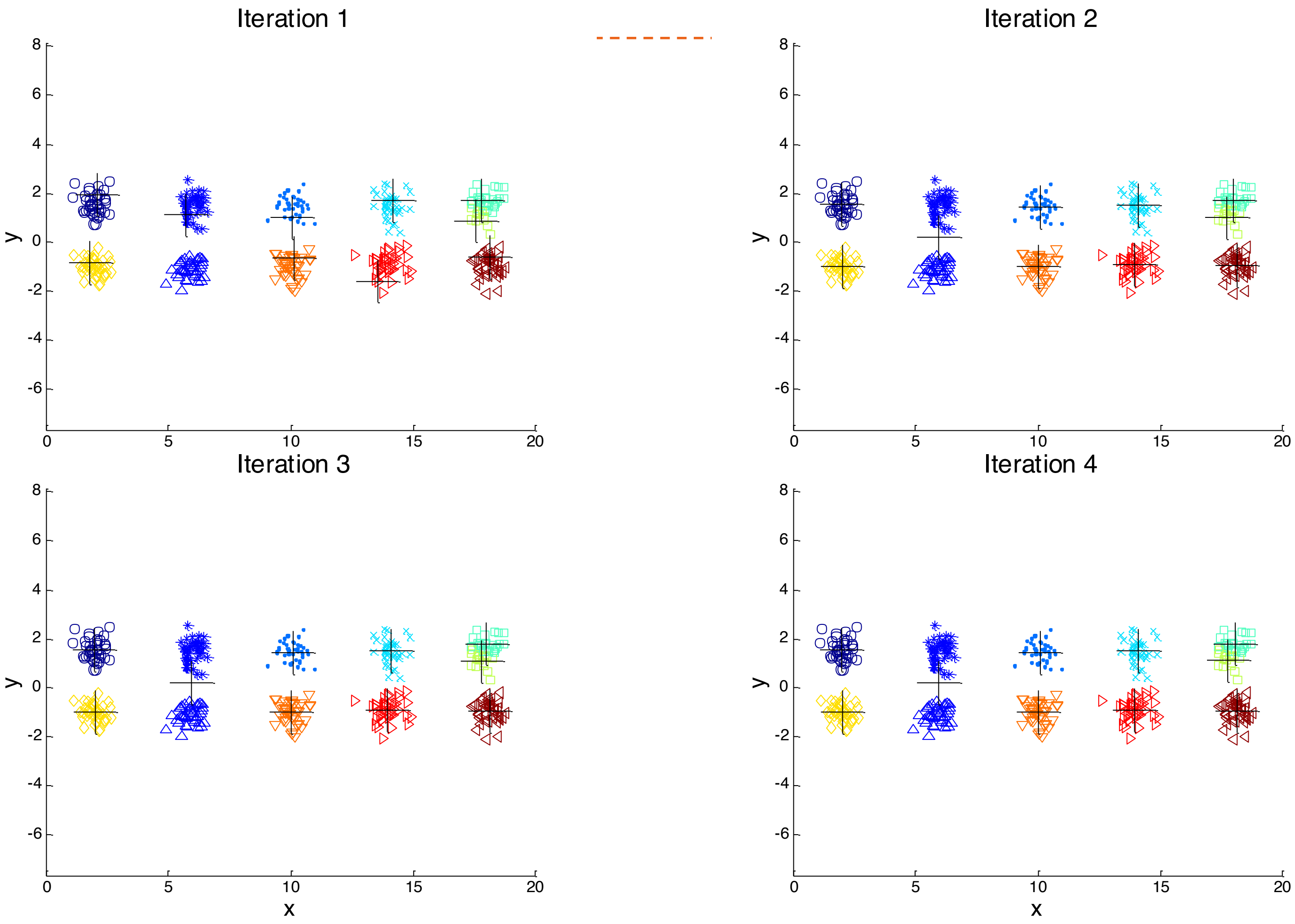
Starting with two initial centroids in one cluster of each pair of clusters

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

- 多次运行 Multiple runs
  - Helps, but probability is not on your side
- 采样并使用层次聚类确定初始质心
- 选择多于k个初始质心，然后在这些初始质心中进行选择
  - Select most widely separated
- 后处理 Postprocessing
  - Generate a larger number of clusters and then perform a hierarchical clustering
- 二分K均值 Bisecting K-means
  - Not as susceptible to initialization issues

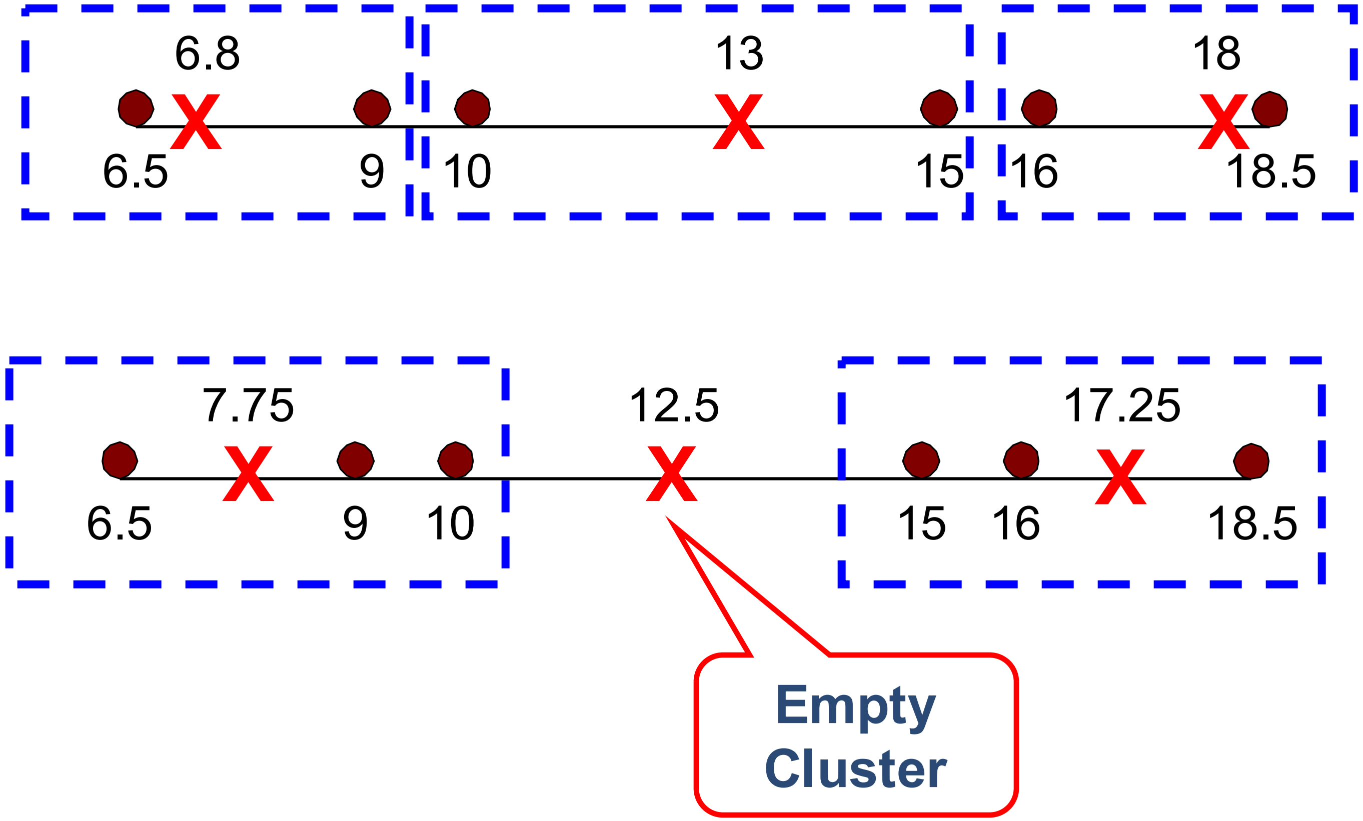
时空复杂度

- **空间复杂度：**
  - $O((m + k)\, n)$
  - m 是样本总数，n 是属性数。

- **时间复杂度：**
- $\mathrm{O}(I * K * m * n)$
  - I 是迭代次数

空簇 Empty Clusters

- K-means can yield empty clusters



处理空簇 Handling Empty Clusters

- 基本的K均值算法可能产生空簇

- 几种策略
  - 选择对SSE贡献最大的点
  - 从具有最高SSE的簇（质量最低的簇）中选择一个点
  - 如果有多个空簇，则可以多次重复上述过程。

降低SSE：Pre-processing and Post-processing

- 预处理 Pre-processing
  - Normalize the data
  - Eliminate outliers
- 后处理 Post-processing
  - Eliminate small clusters that may represent outliers
  - Split ‘loose’ clusters, i.e., clusters with relatively high SSE
  - Merge clusters that are ‘close’ and that have relatively low SSE
  - Can use these steps during the clustering process
    - ISODATA

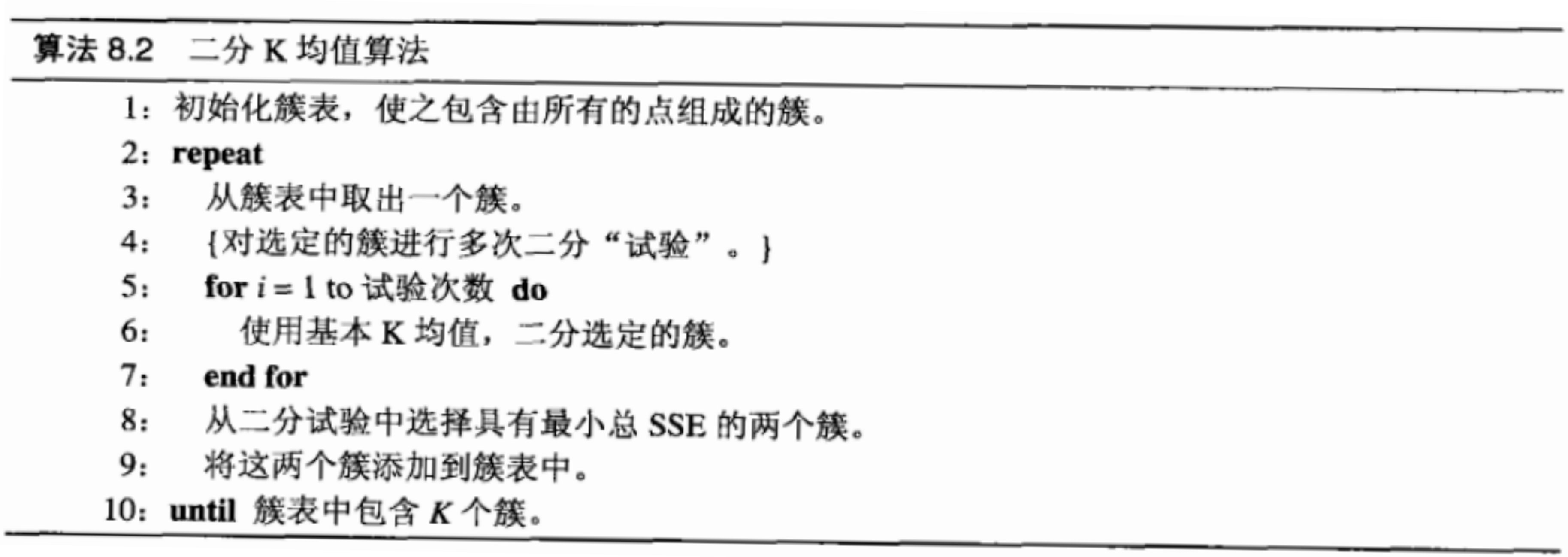
增量更新质心 Updating Centers Incrementally

- 在基本K均值算法中，在将所有点分配给质心之后需要更新质心

- 另一种方法是在每次分配（每个点的分配）后更新质心（增量方法 incremental approach ）
  - 每个分配更新零个或两个质心
  - 开销大 More expensive
  - 次序依赖 Introduces an order dependency
  - Never get an empty cluster
  - Can use “weights” to change the impact

二分 K 均值 Bisecting K-means

- 二分 K 均值算法： Bisecting K-means algorithm
  - 可以产生分区或层次聚类的K均值的变体。Variant of K-means that can produce a partitional or a hierarchical clustering
  - 缓解初始化问题

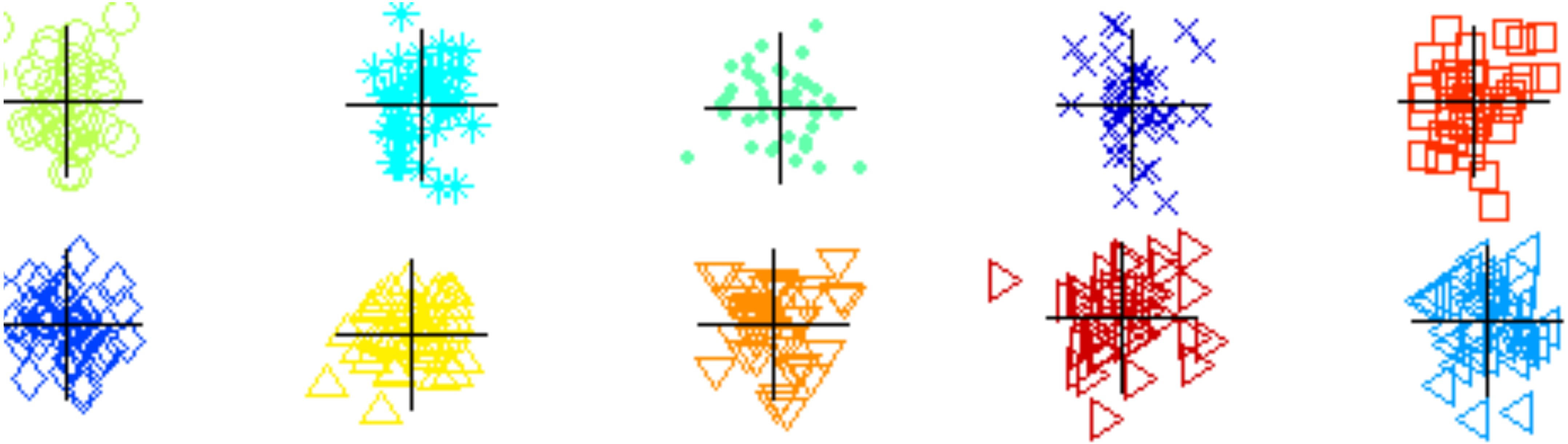


算法 8.2 二分 K 均值算法

```
初始化簇表，使之包含由所有的点组成的簇。
repeat
    从簇表中取出一个簇。
    {对选定的簇进行多次二分“试验”。}
    for $i = 1$ to 试验次数 do
        使用基本 K 均值，二分选定的簇。
    end for
    从二分试验中选择具有最小总 SSE 的两个簇。
    将这两个簇添加到簇表中。
until 簇表中包含 $K$ 个簇。
```

CLUTO: http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview

示例 Bisecting K-means Example

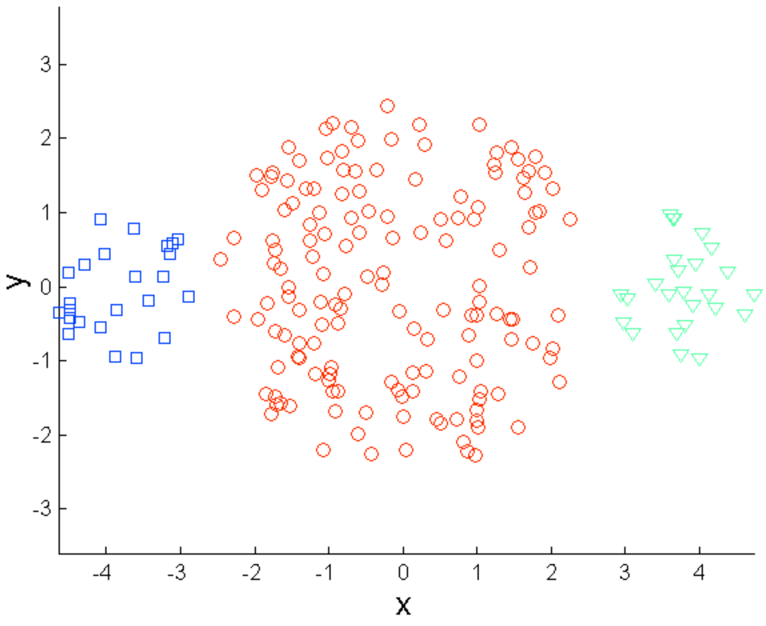


K均值聚类局限性 Limitations of K-means

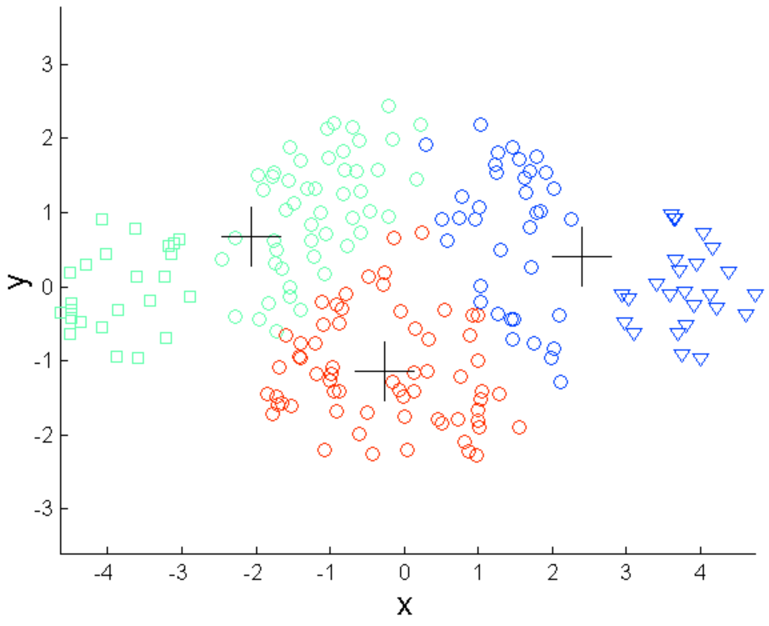
- K均值在以下情况时会遇到问题：
  - 簇具有不同的大小 Sizes
  - 簇具有不同的密度 Densities
  - 簇不是球形的 Non-globular shapes

- 处理包含离群点的数据时也有问题

不同大小 ：Limitations of K-means: Differing Sizes

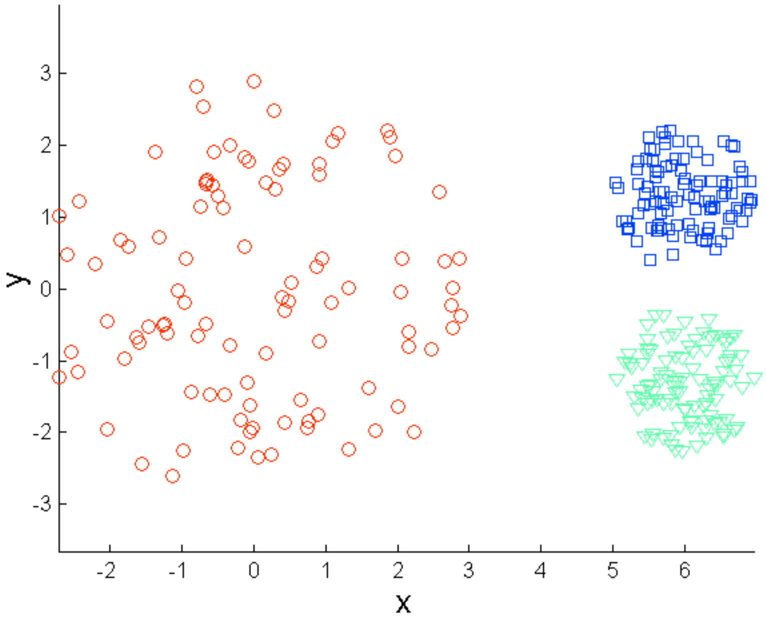


Original Points

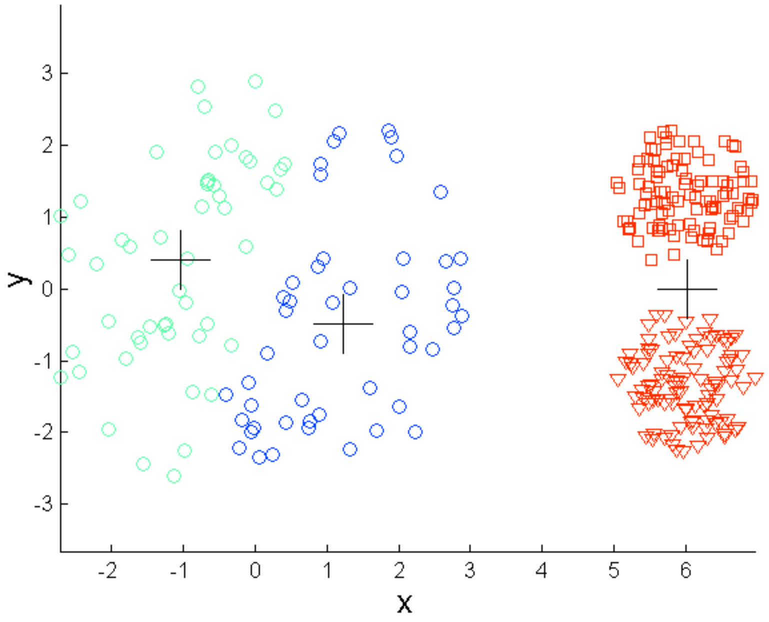


K-means (3 Clusters)

不同密度 Limitations of K-means: Differing Density

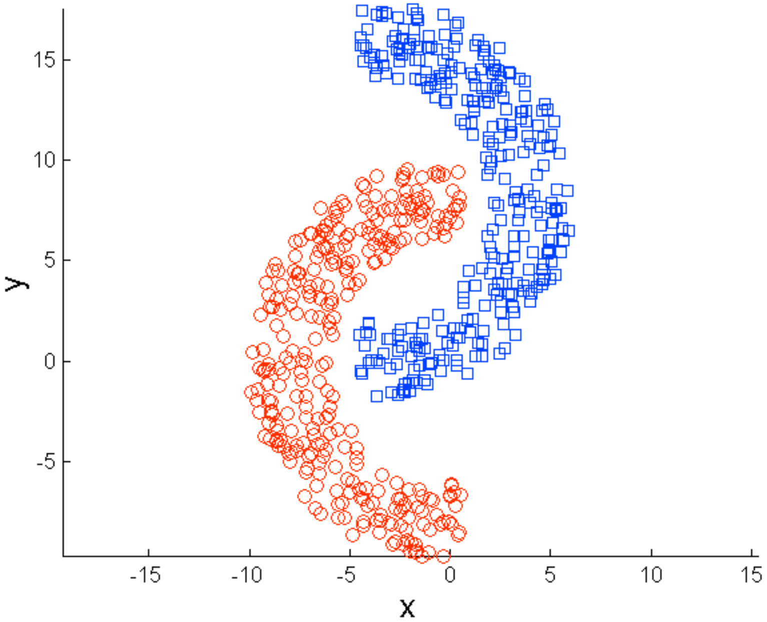


Original Points

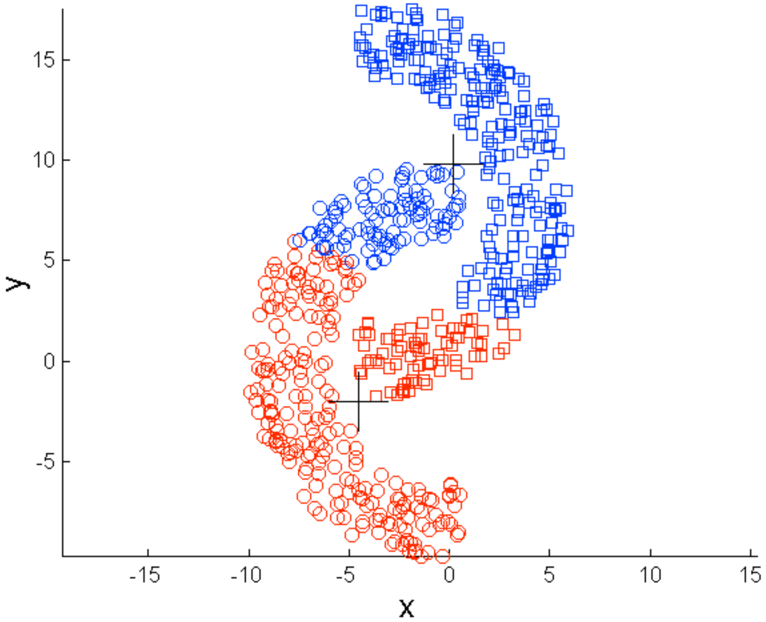


K-means (3 Clusters)

非球形 Limitations of K-means: Non-globular Shapes

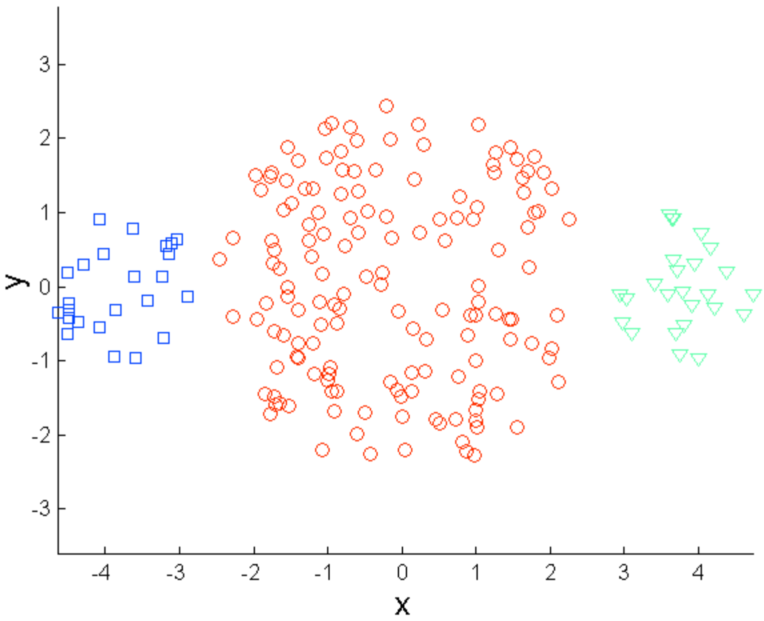


Original Points

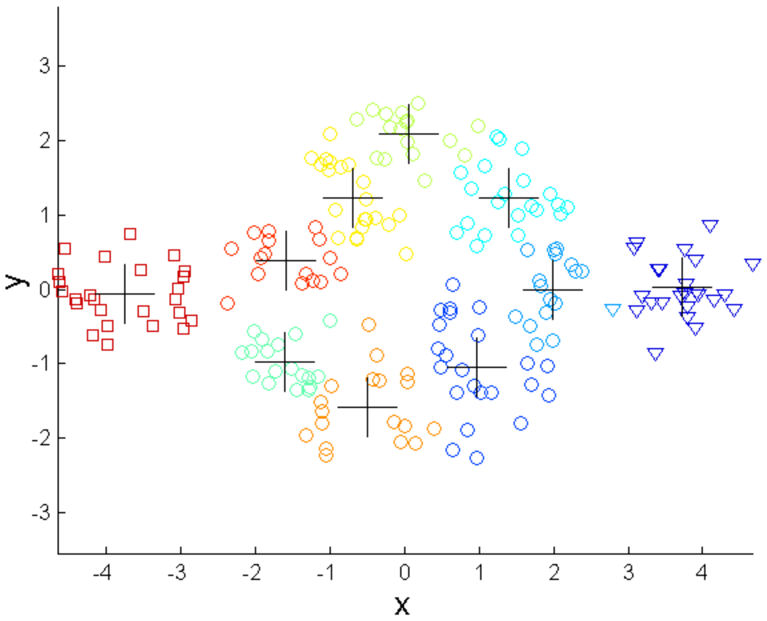


K-means (2 Clusters)

克服其局限性 Overcoming K-means Limitations



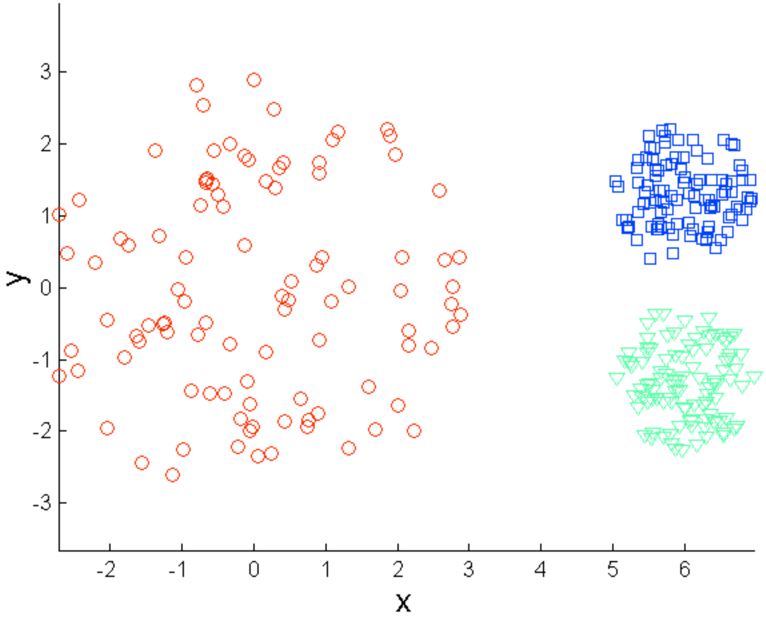
Original Points



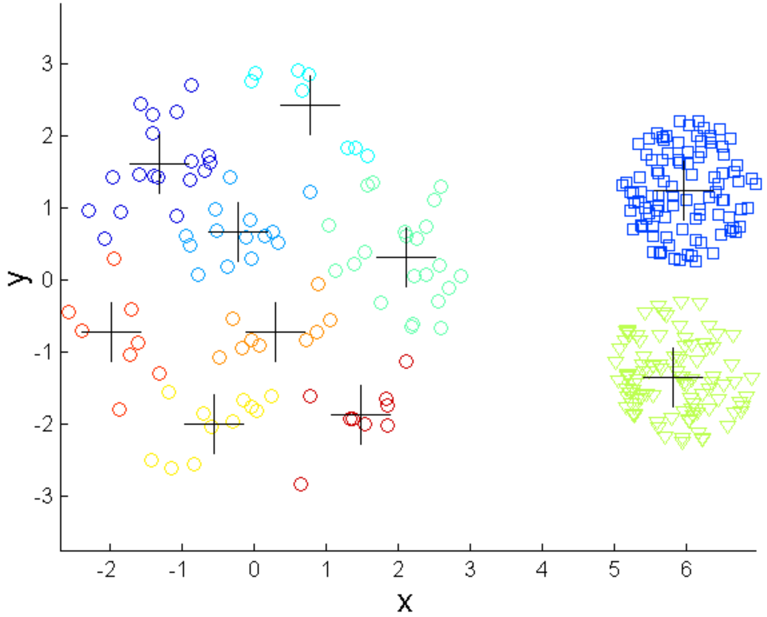
K-means Clusters

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

克服其局限性 Overcoming K-means Limitations

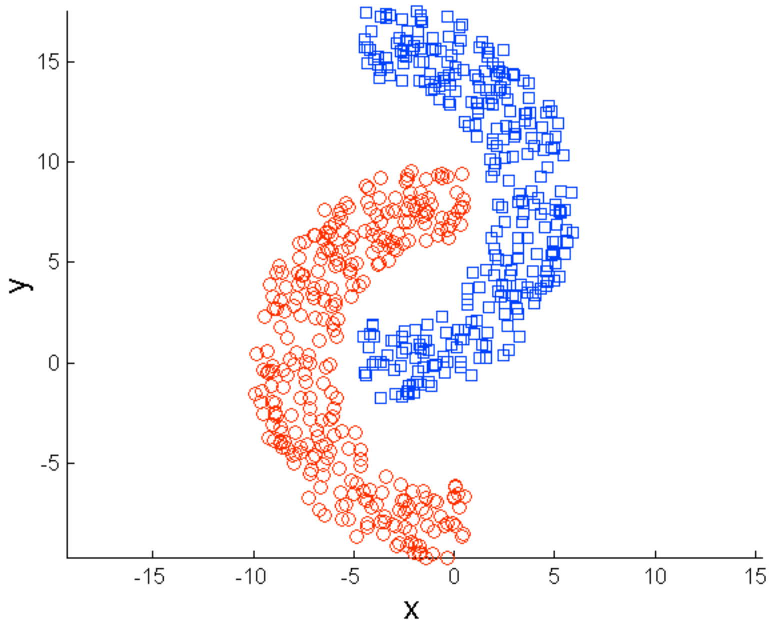


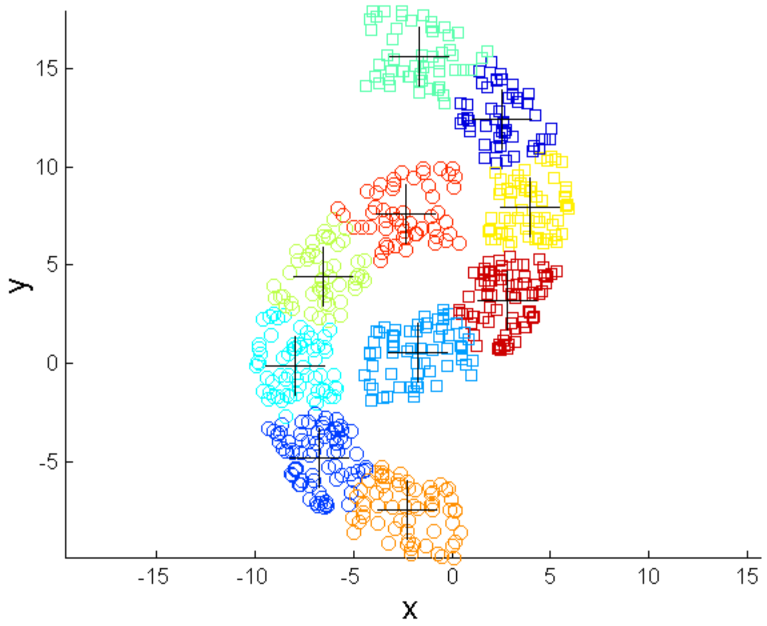
Original Points



K-means Clusters

克服其局限性 Overcoming K-means Limitations





Original Points

K-means Clusters

单选题 1分

⚙ 设置

下面关于聚类说法**错误**的是？

A 聚类分析是无监督的，对聚类分析结果往往是主观的

B 聚类分析能够帮助我们理解数据的分组/分布情况，也可以用于减小需要处理的数据集的大小。

[✓ C] 在K均值算法中，增加簇的个数可以减小SSE，因此簇的个数越多越好

D K均值算法算法需要设置簇的个数

提交