数据挖掘

第6章 聚类

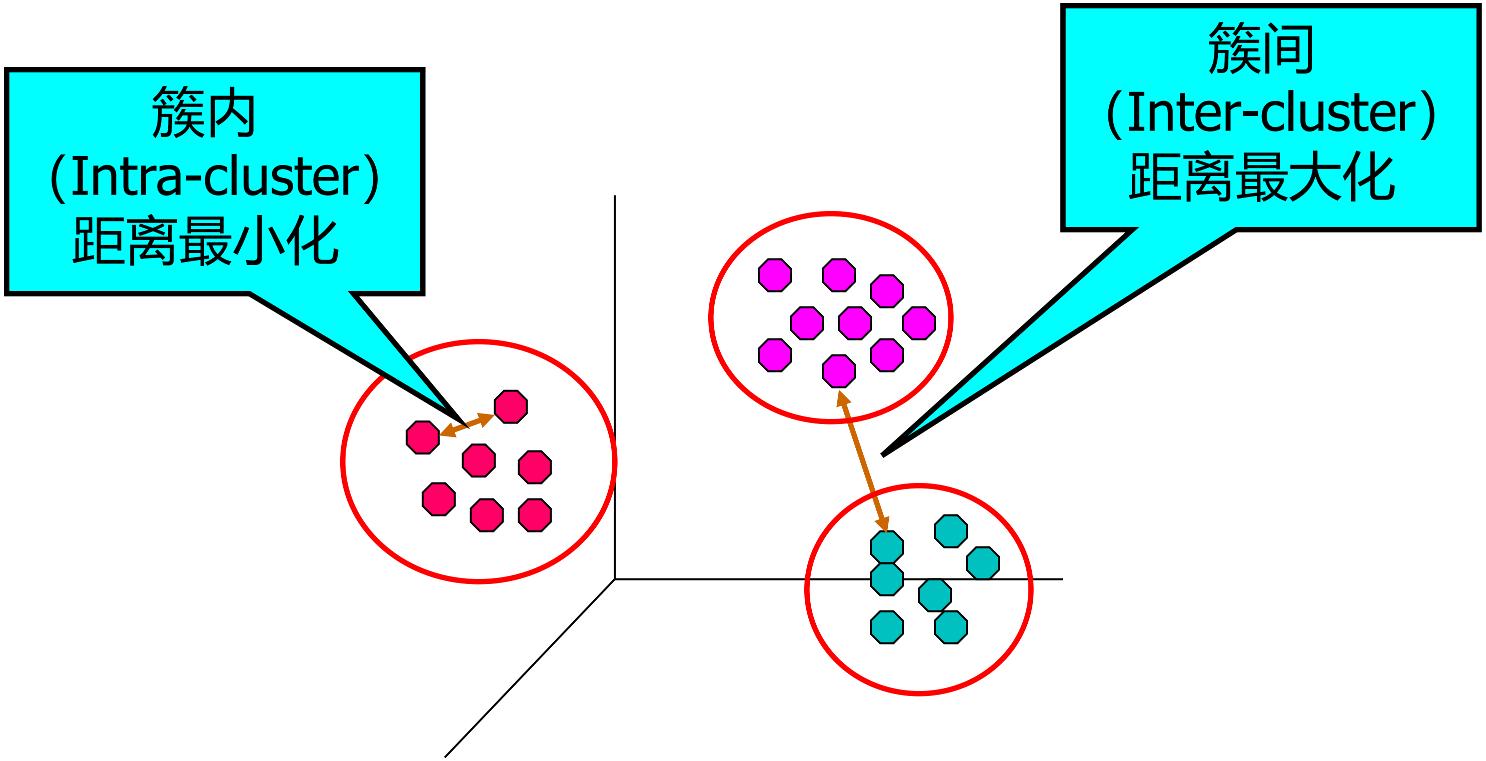
教师：王东京

学院：计算机学院

邮箱：dongjing.wang@hdu.edu.cn

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)



簇内(Intra-cluster)距离最小化

簇间(Inter-cluster)距离最大化

聚类分析应用 Applications of Cluster Analysis

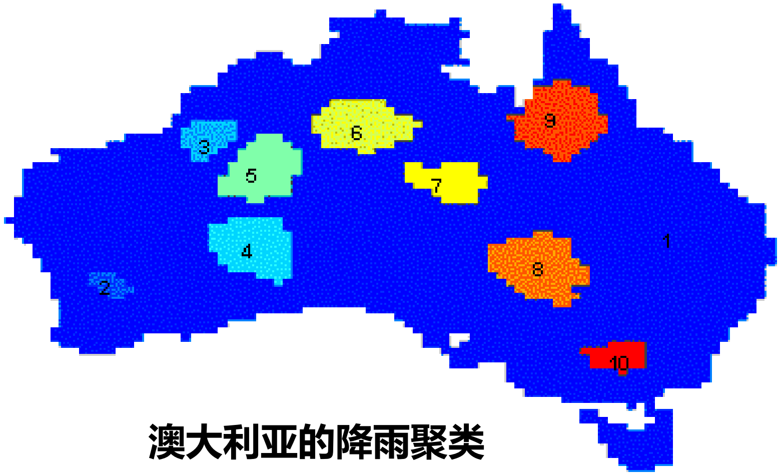
理解 Understanding

- 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

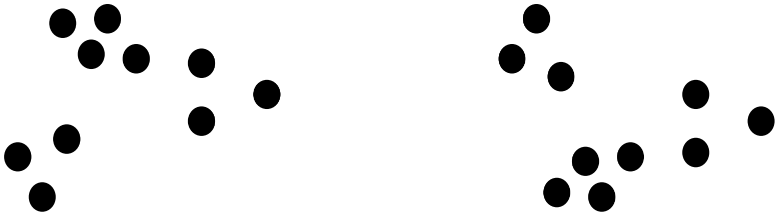
监督分类 Supervised classification

- 有类别标签信息

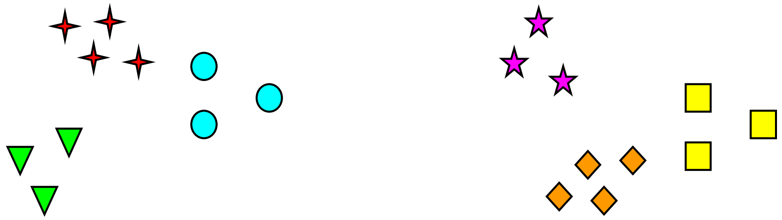
关联分析

- Local vs. global connections

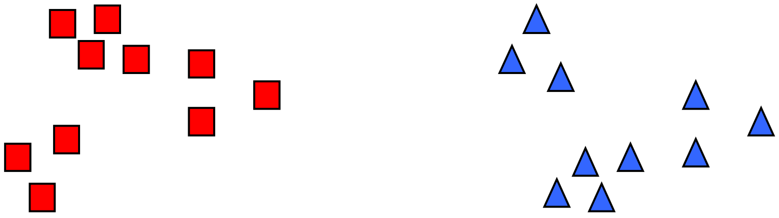
簇的概念可能不明确 Notion of a Cluster can be Ambiguous



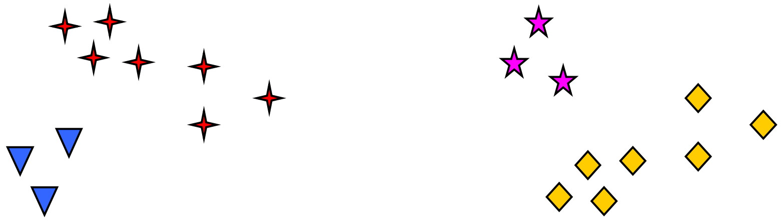
多少簇（cluster）？



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

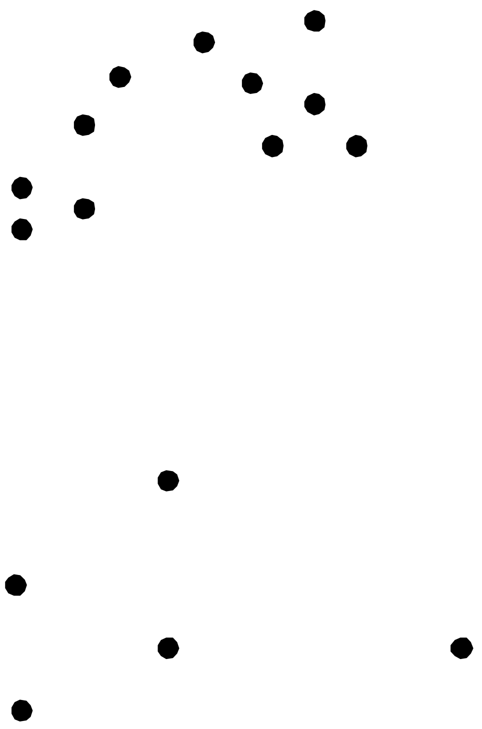
划分聚类 Partitional Clustering

- 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

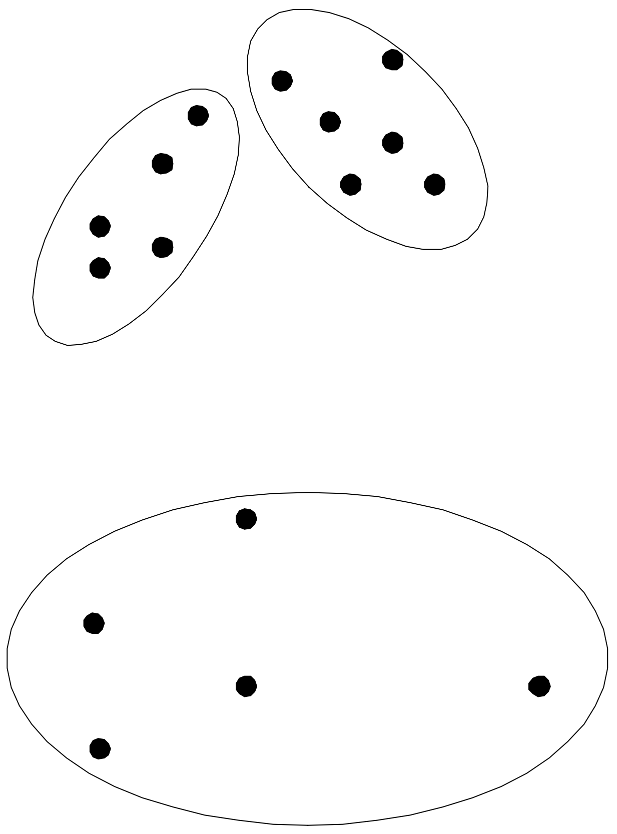
层次聚类 Hierarchical clustering

- 一组嵌套的聚类，组织成一个层次树

划分聚类 Partitional Clustering（非嵌套）

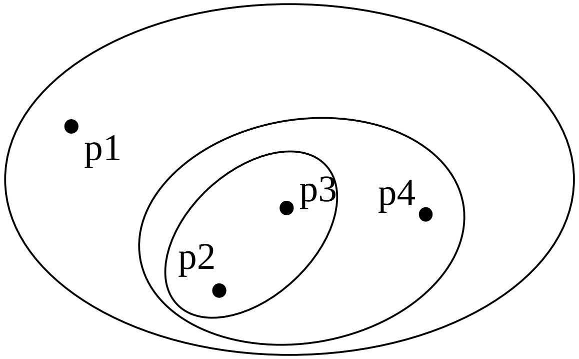


Original Points

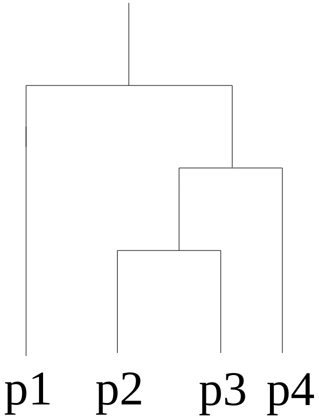


A Partitional Clustering

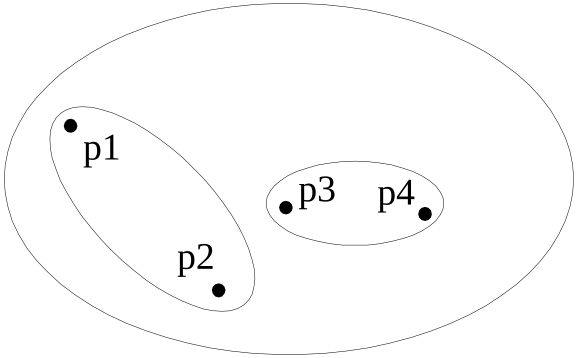
层次聚类 Hierarchical Clustering（嵌套）



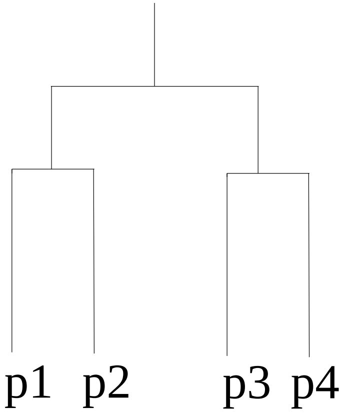
Traditional Hierarchical Clustering



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

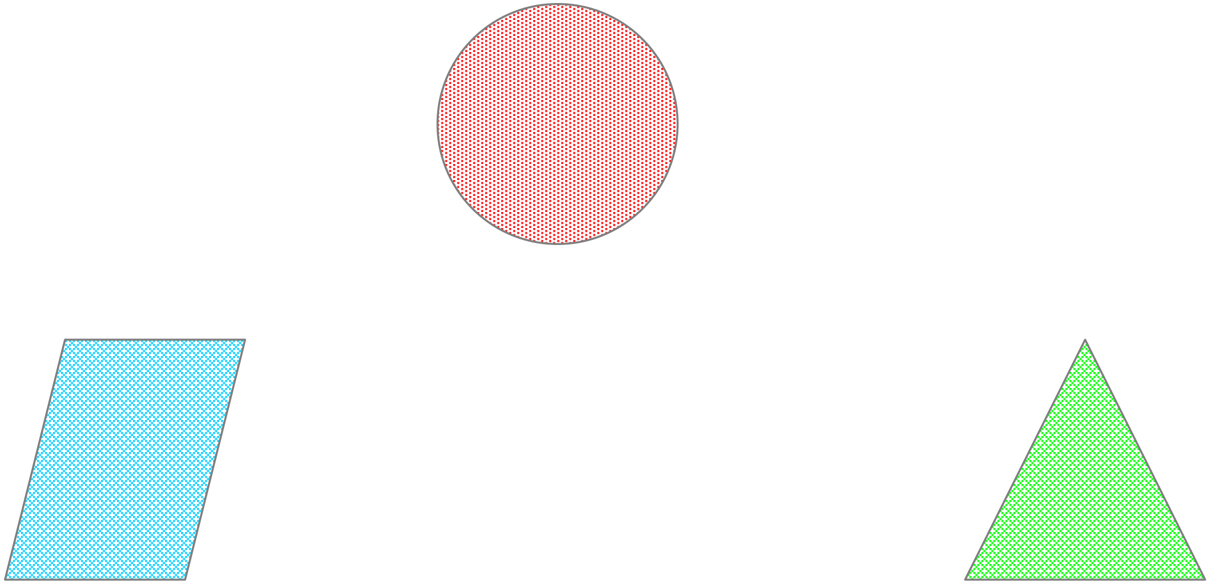
- 具有共同性质的（概念簇）

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）。

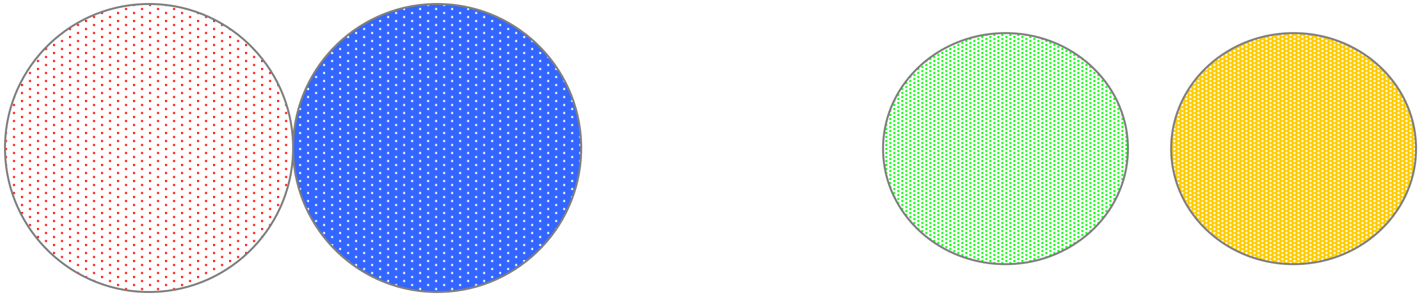


(a) 3 well-separated clusters

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

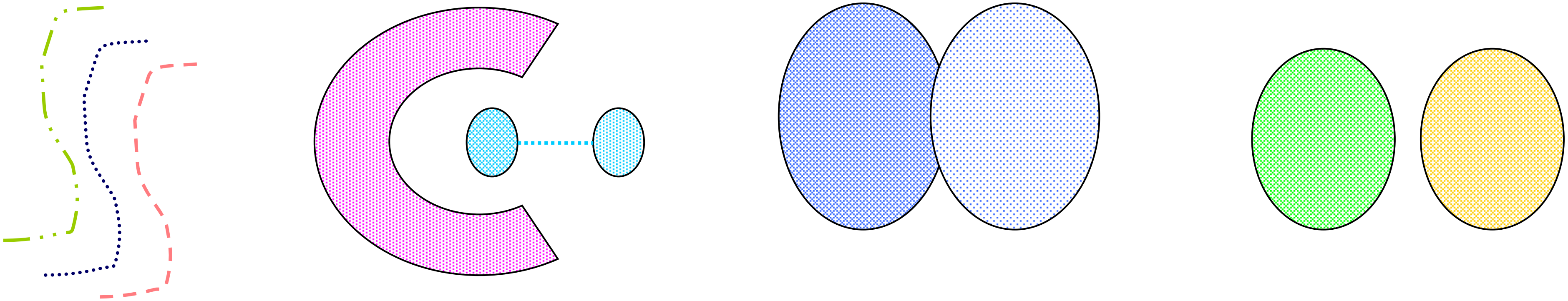


(b) 4 center-based clusters

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.

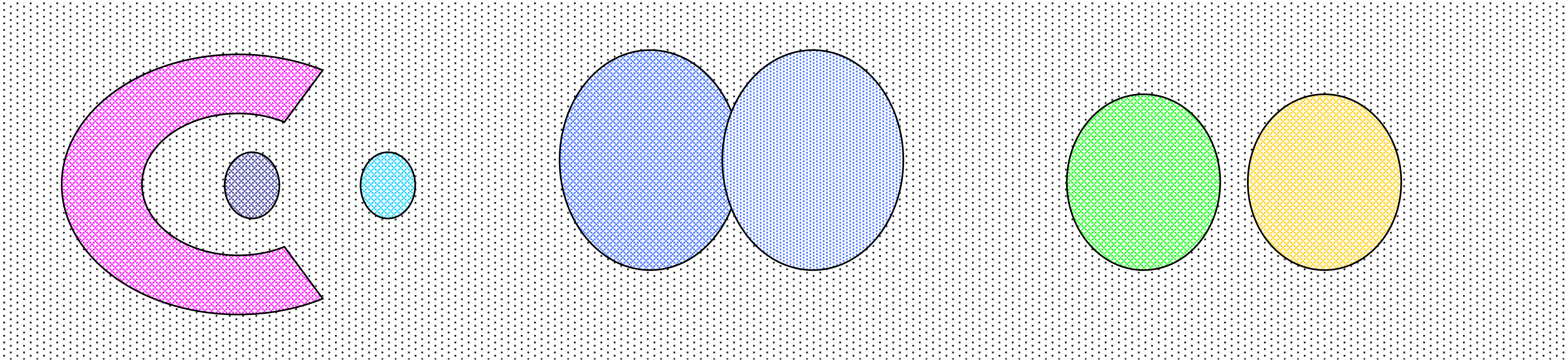


(c) 8 contiguous clusters

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.

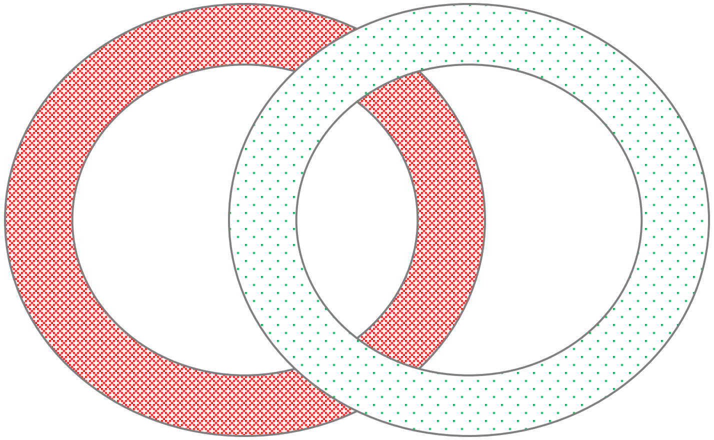


(d) 6 density-based clusters

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

层次聚类 Hierarchical clustering

基于密度的聚类 Density-based clustering

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

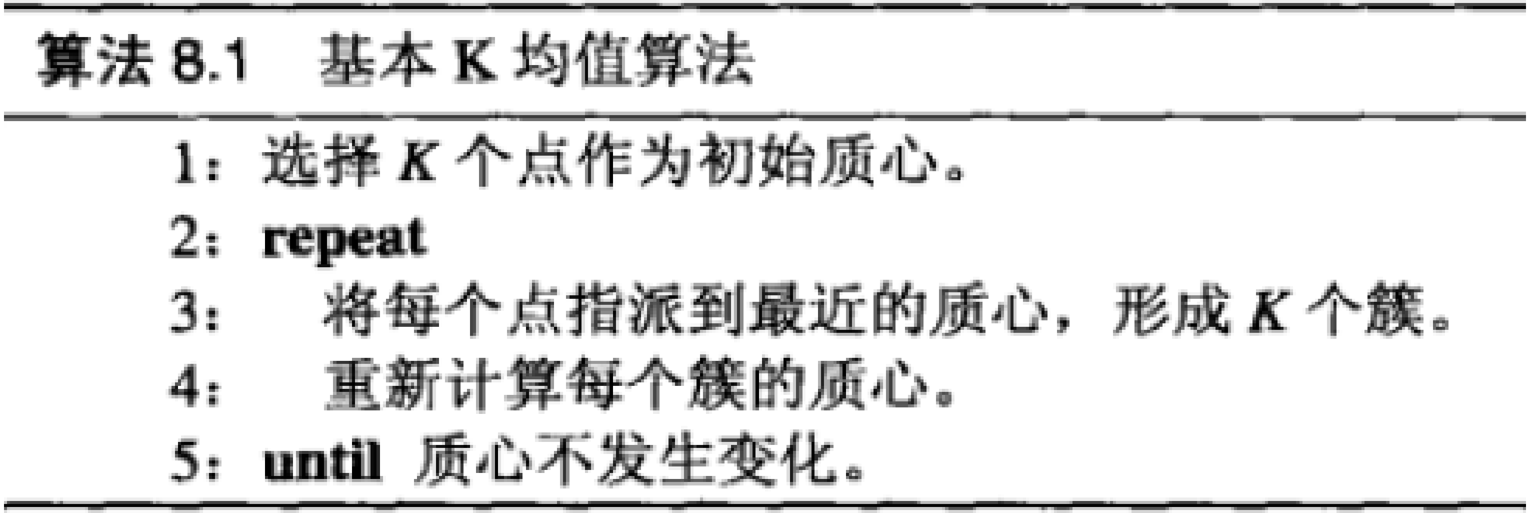
划分聚类（Partitional clustering）方法

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

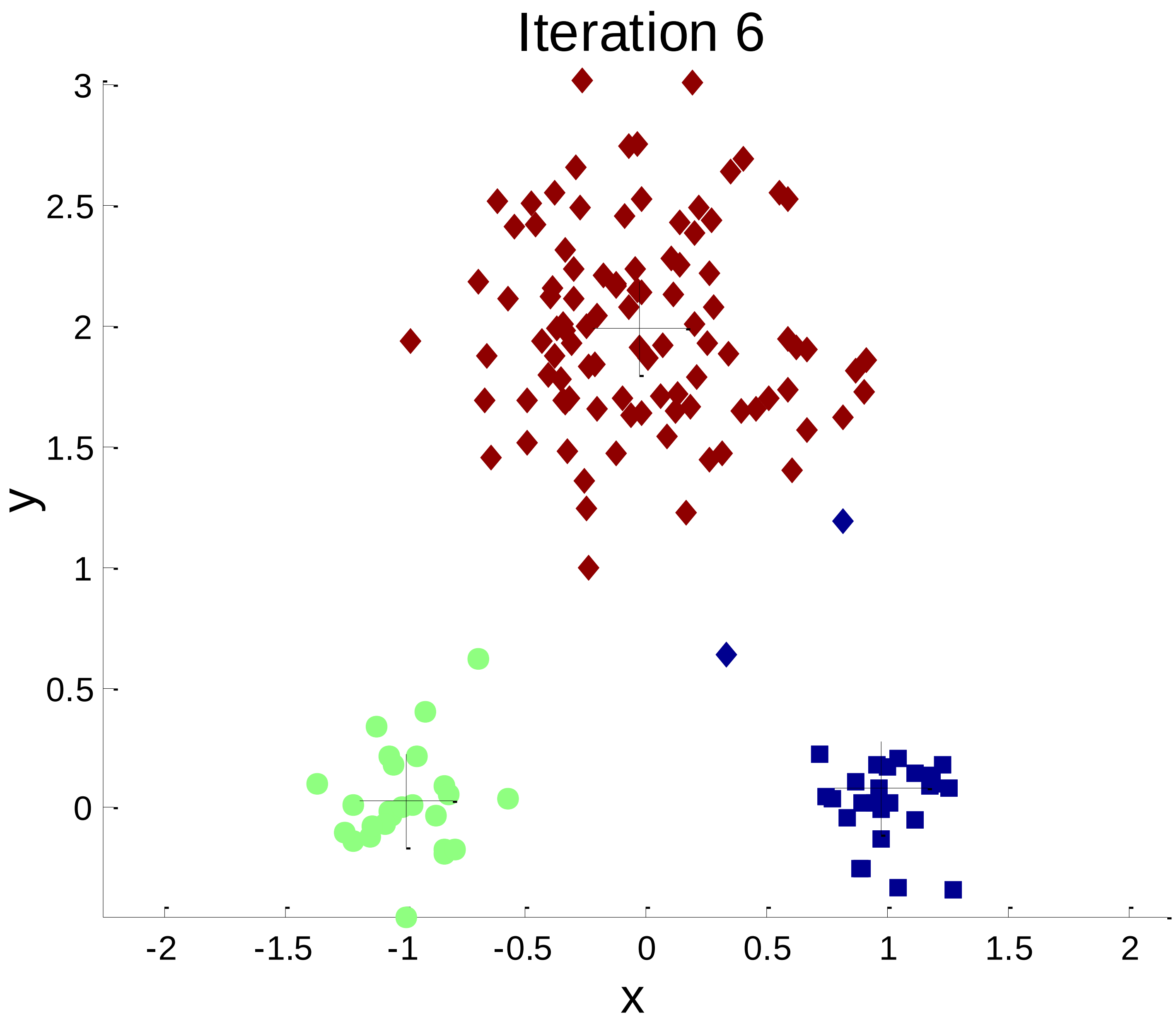
The basic algorithm is very simple



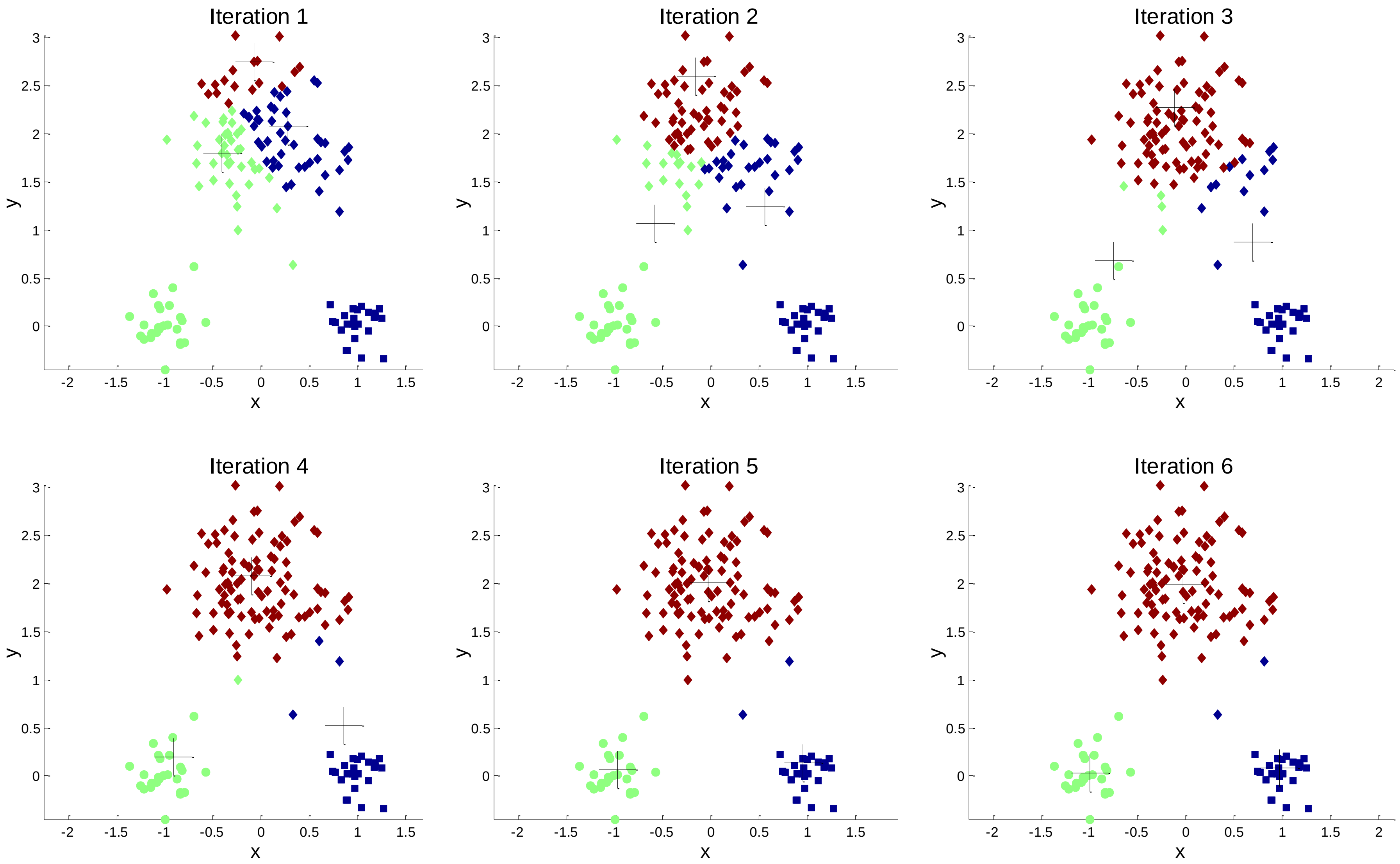
算法 8.1　基本 K 均值算法

```
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



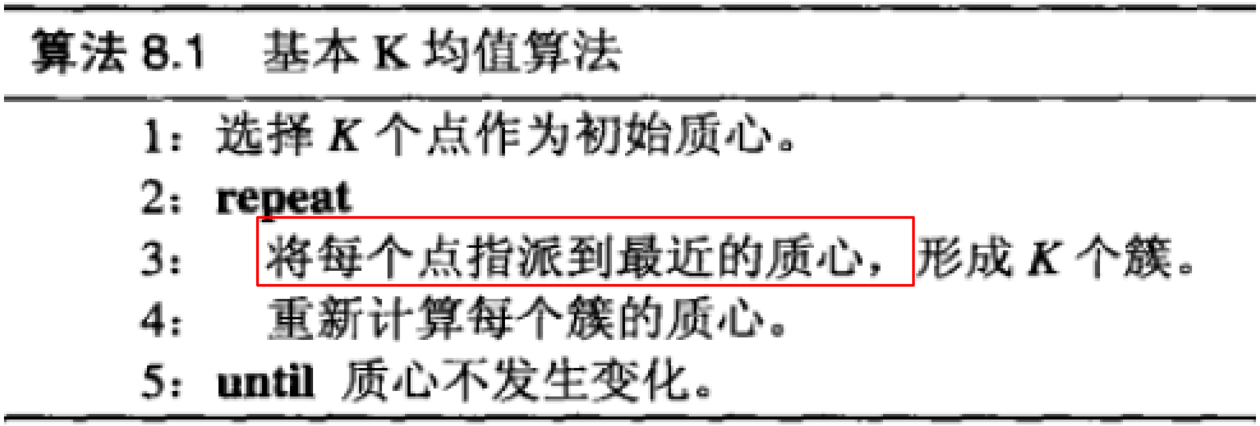
K均值聚类：1. 指派点到最近的质心

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

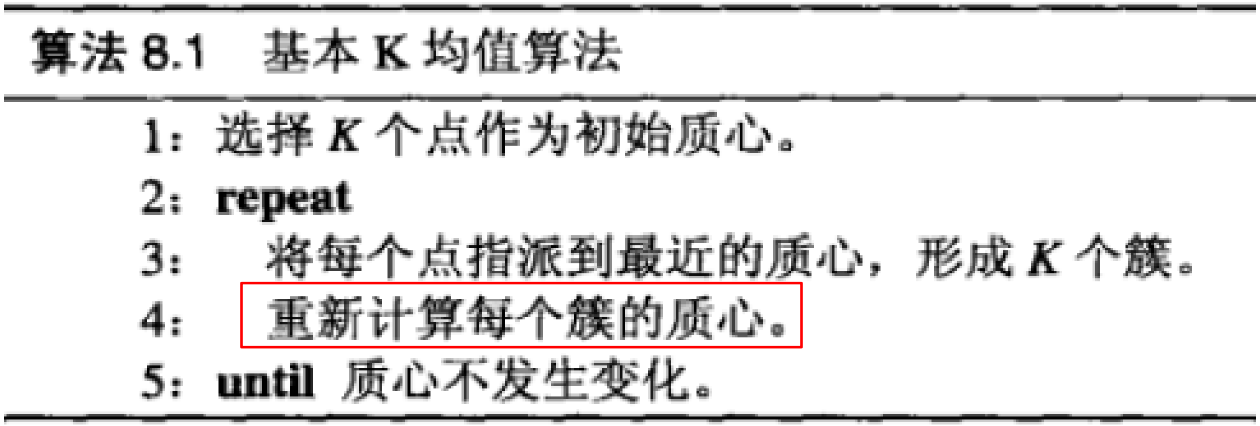
- 误差的平方和（Sum of Squared Error, **SSE**）

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

- 均值!

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

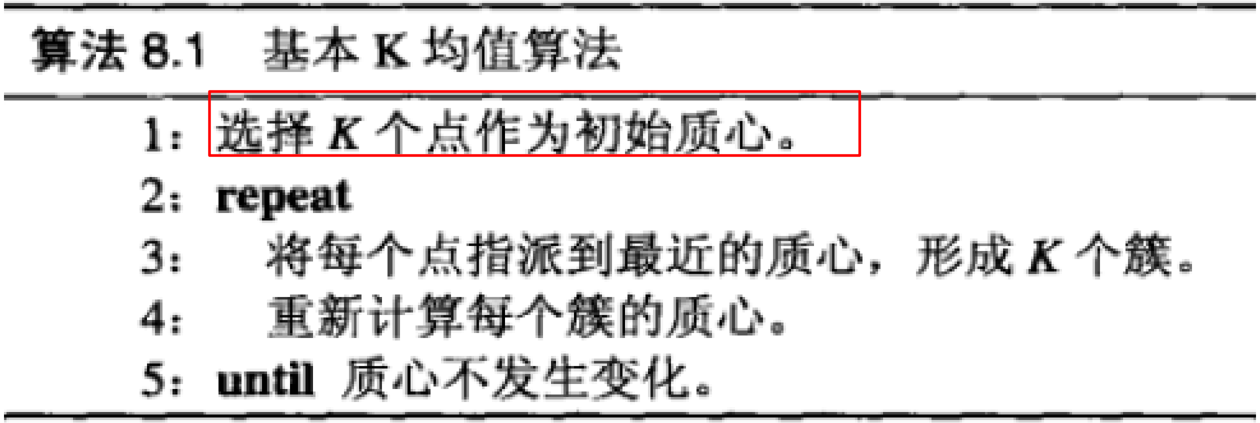


算法 8.1　基本 K 均值算法

```
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?

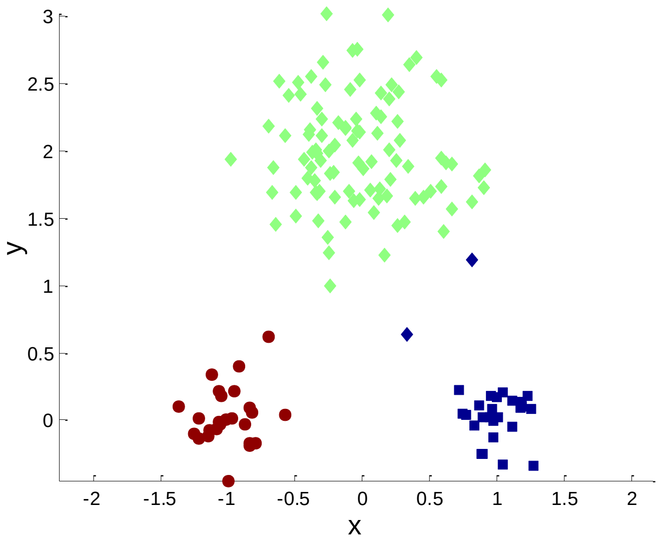


算法 8.1　基本 K 均值算法

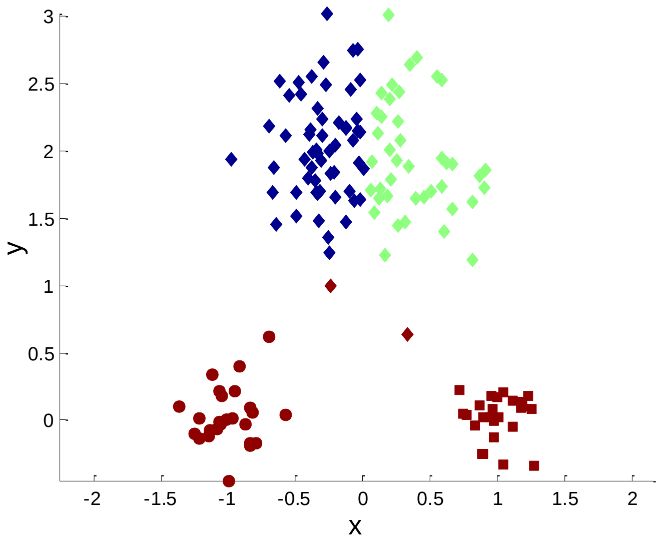
```
选择 $K$ 个点作为初始质心。
repeat
    将每个点指派到最近的质心，形成 $K$ 个簇。
    重新计算每个簇的质心。
until 质心不发生变化。
```

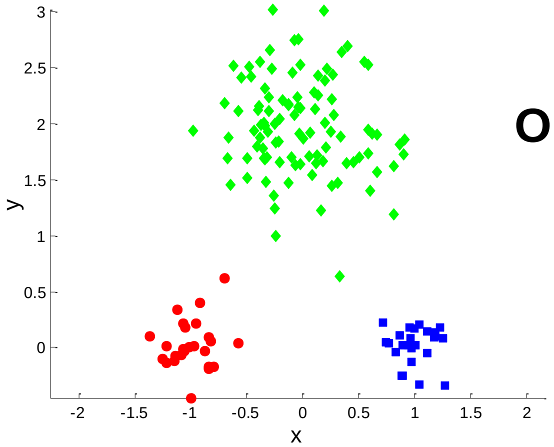
K均值聚类：3. 选择初始质心

Optimal Clustering



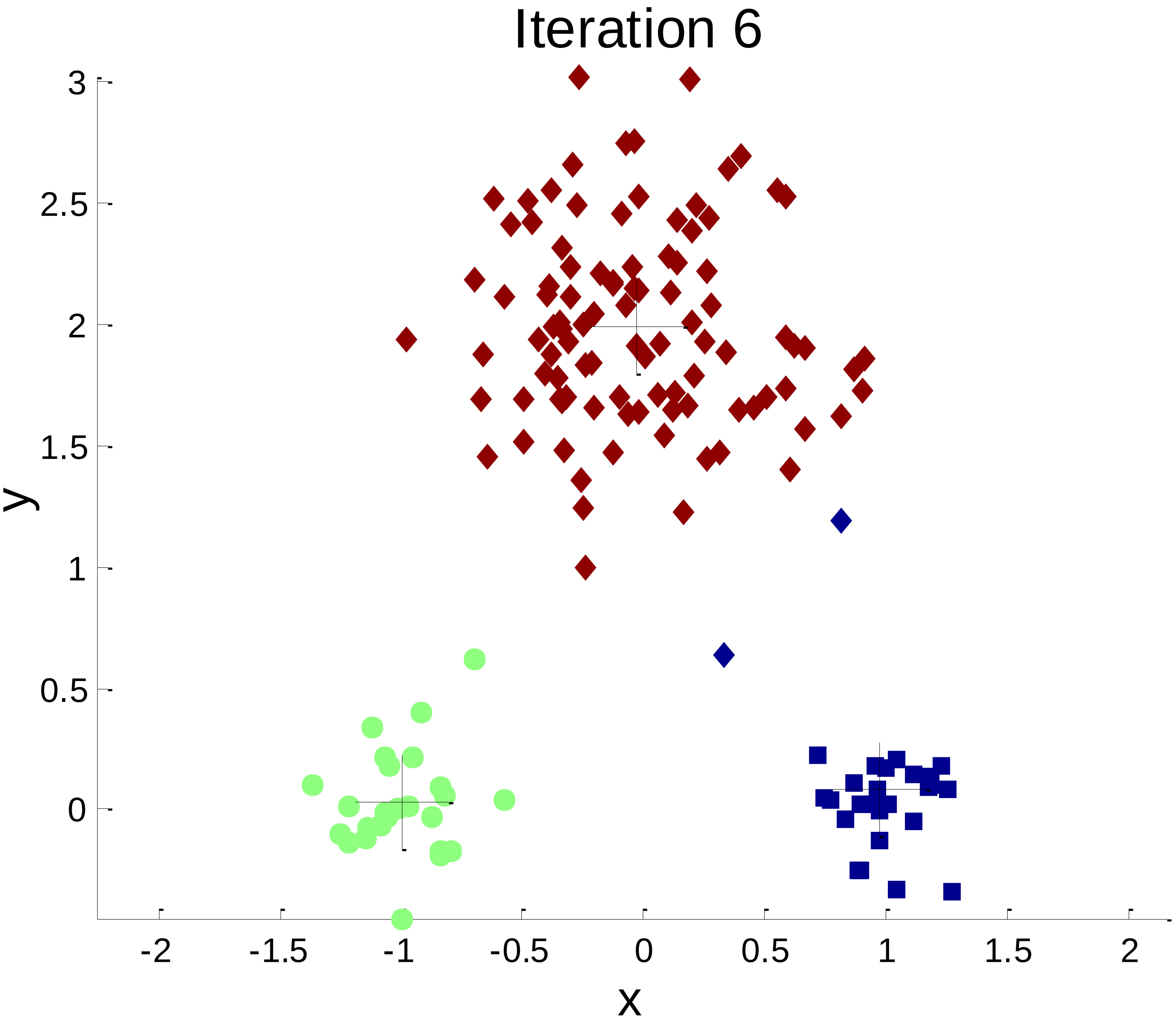
Sub-optimal Clustering



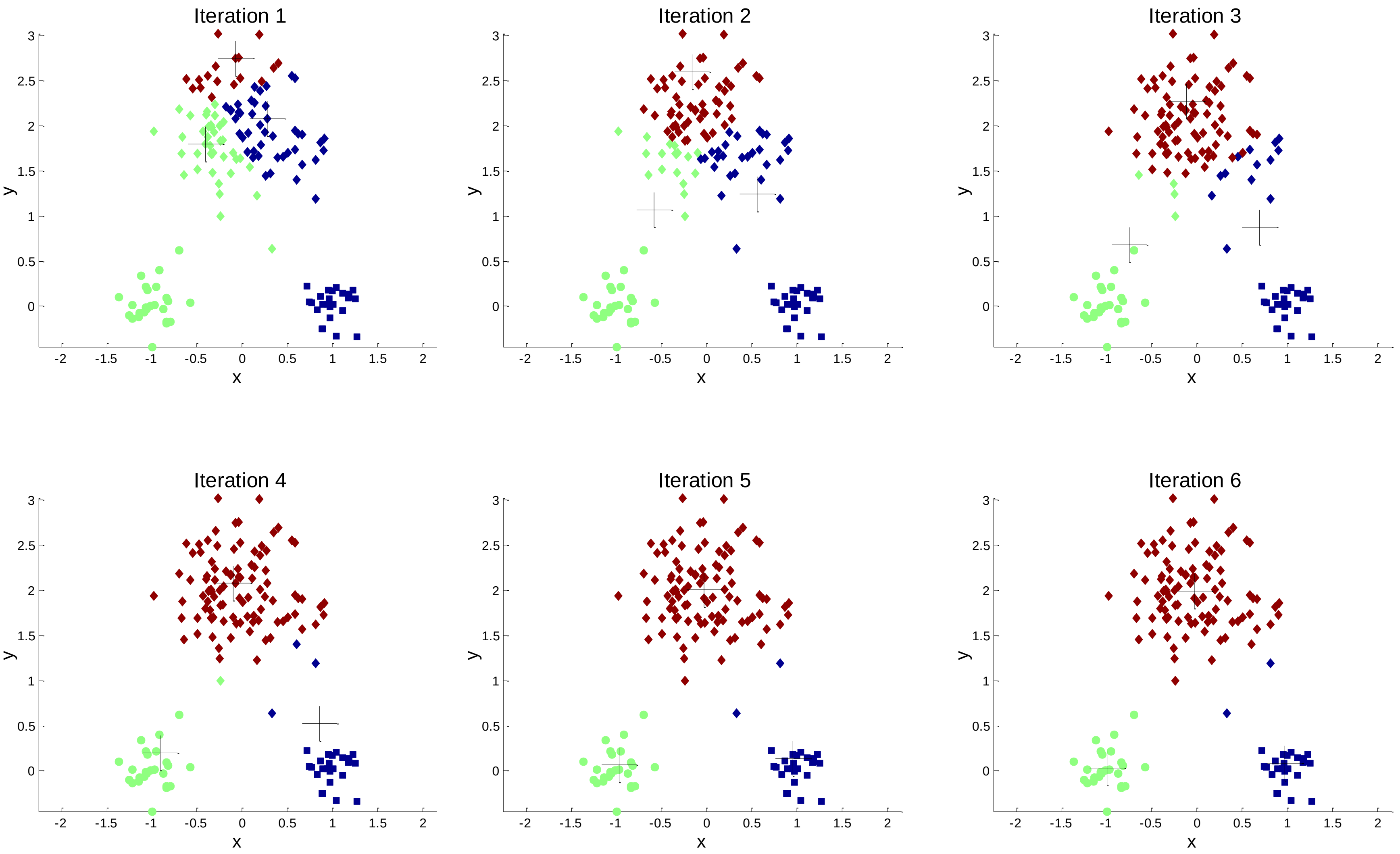


Original Points

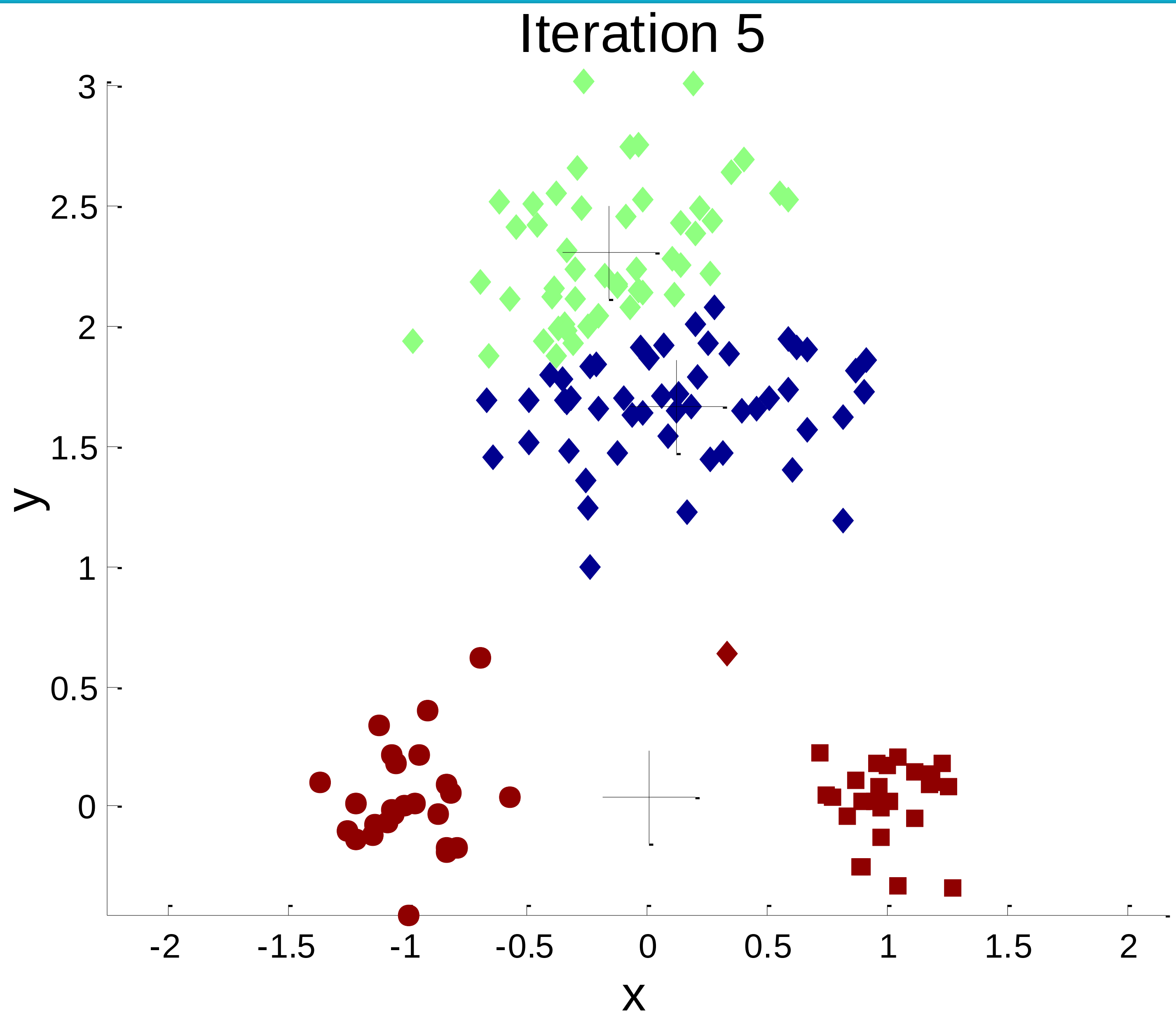
初始质心选择 Importance of Choosing Initial Centroids



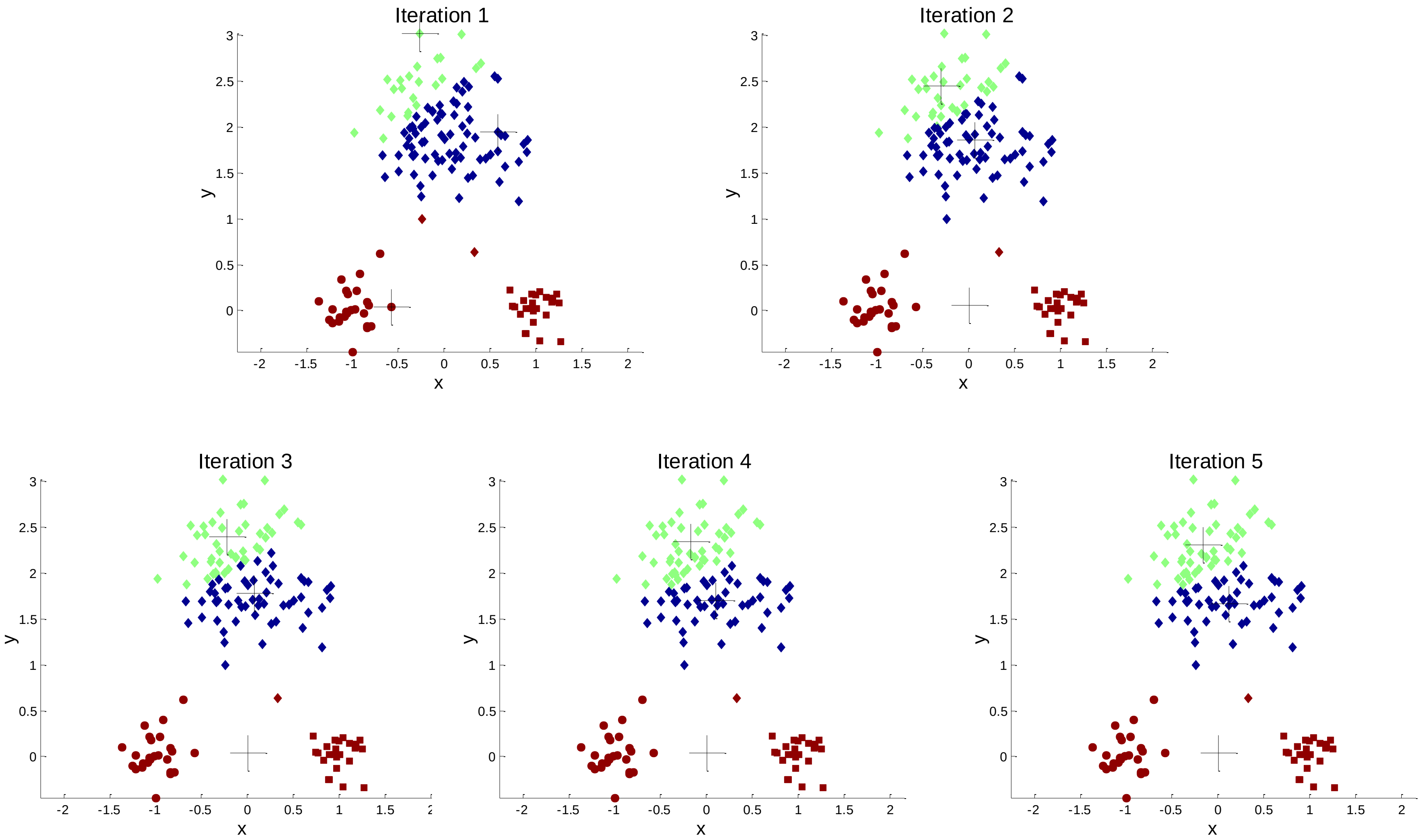
初始质心选择 Importance of Choosing Initial Centroids



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

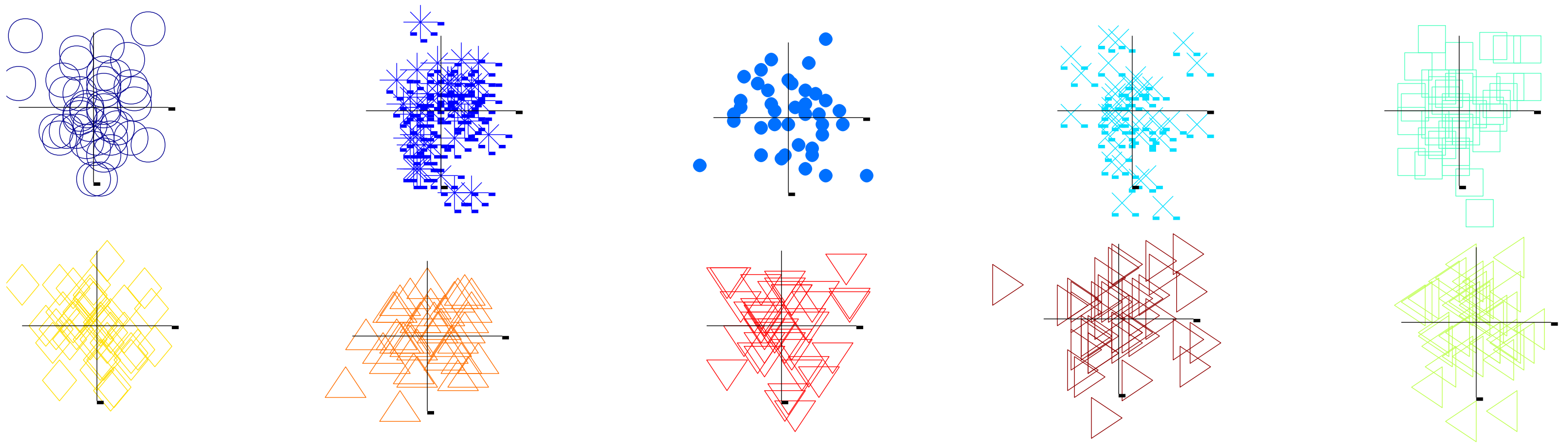
- 这个概率会随着K的增加而降低
- 如果簇的大小都为n，则

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

- 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

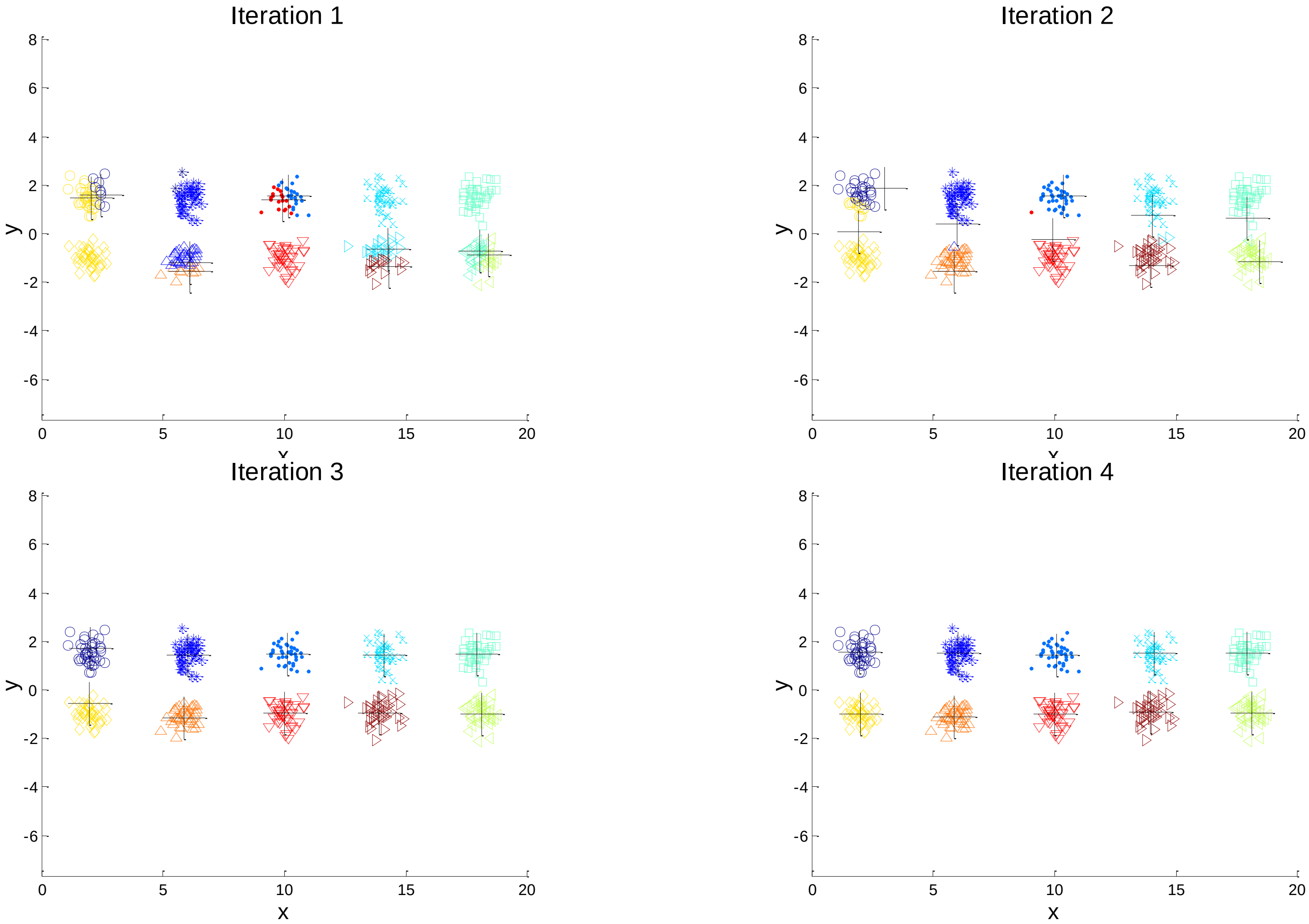
- 考虑**五对**簇（five pairs of clusters）的例子

10 Clusters Example



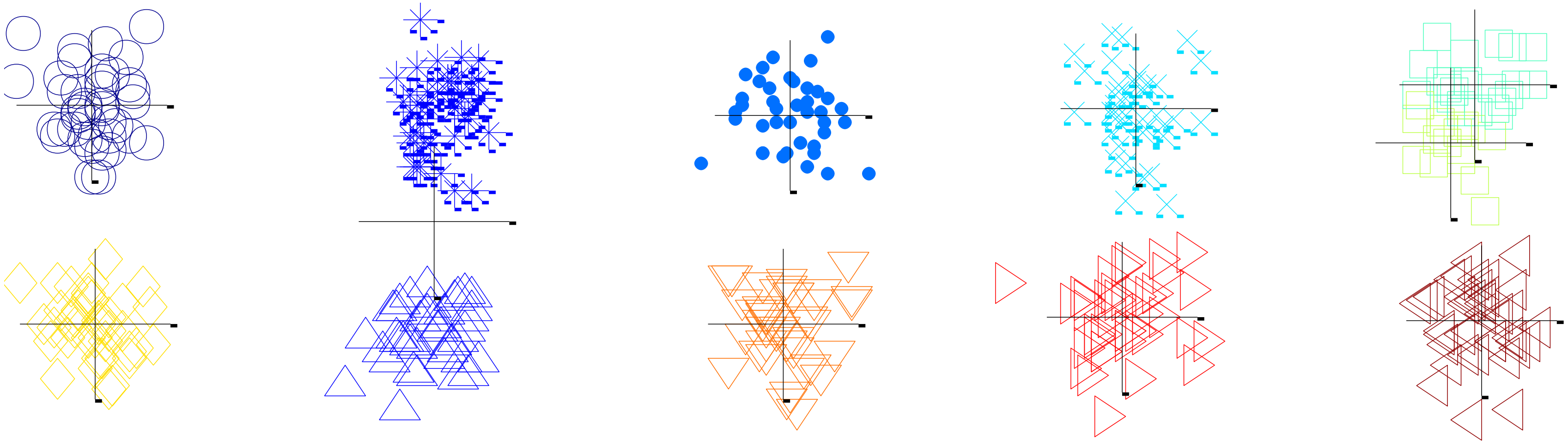
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example



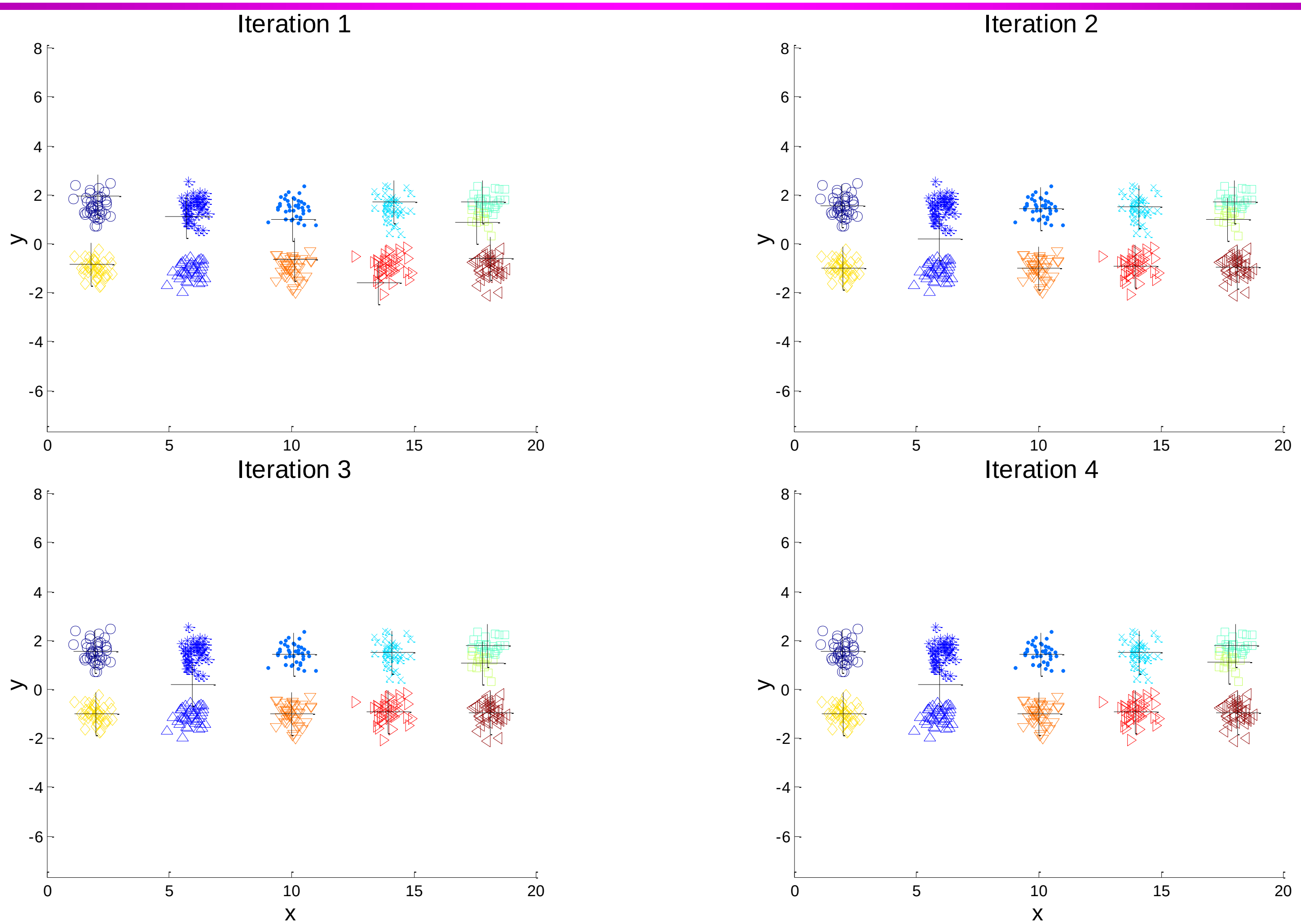
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example



Starting with some pairs of clusters having three initial centroids, while other have only one.

10 Clusters Example



Starting with <span>some pairs of clusters having three initial centroids</span>, 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。

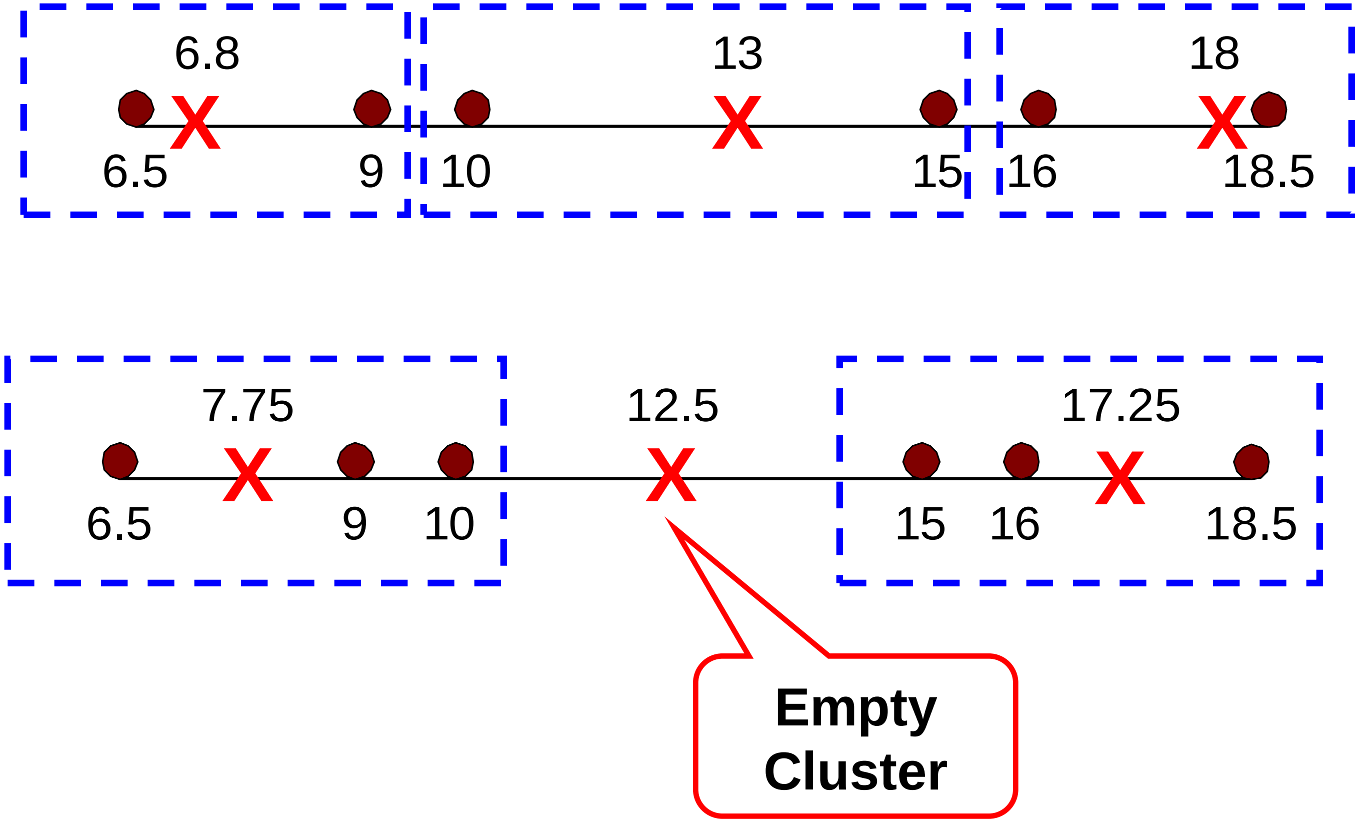
时间复杂度：

$O(I * K * m * n)$

- 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

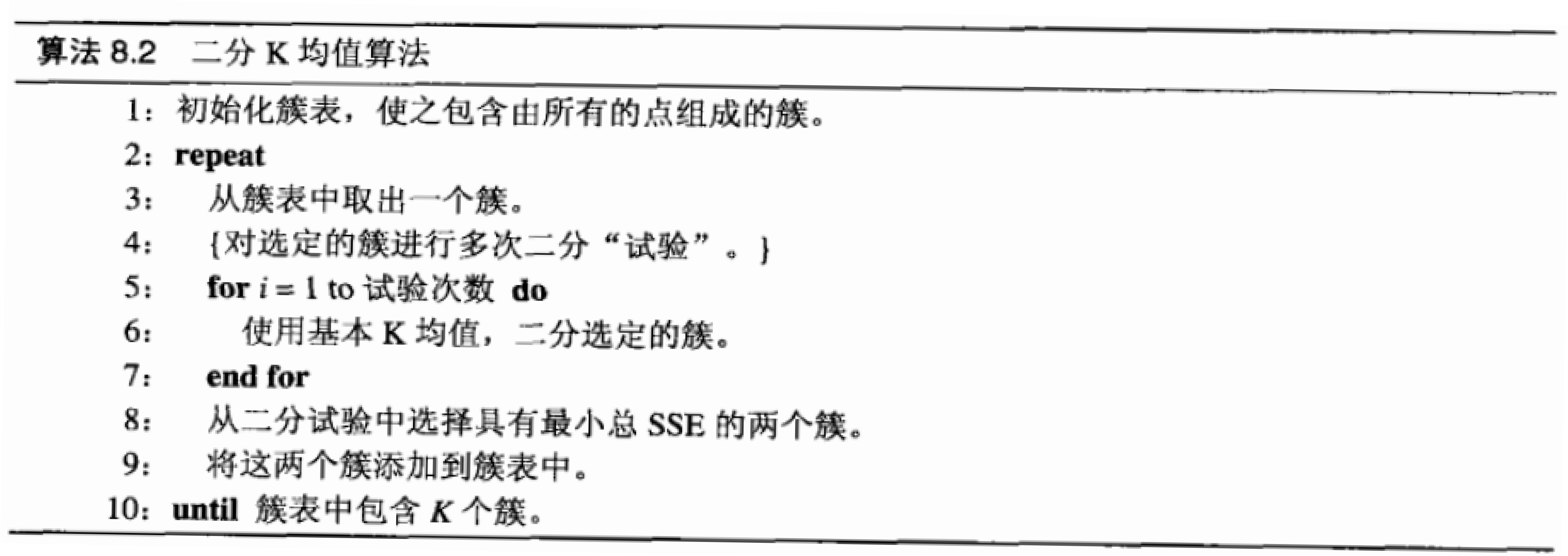
另一种方法是在每次分配（每个点的分配）后更新质心（增量方法 incremental approach ）

- 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

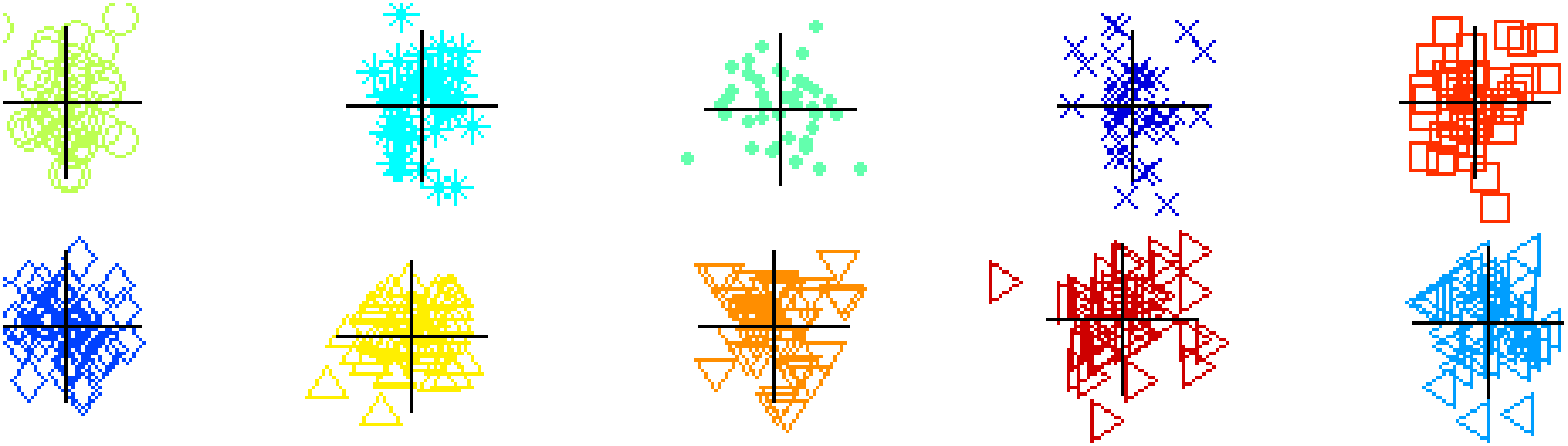


算法 8.2 二分 K 均值算法

```
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



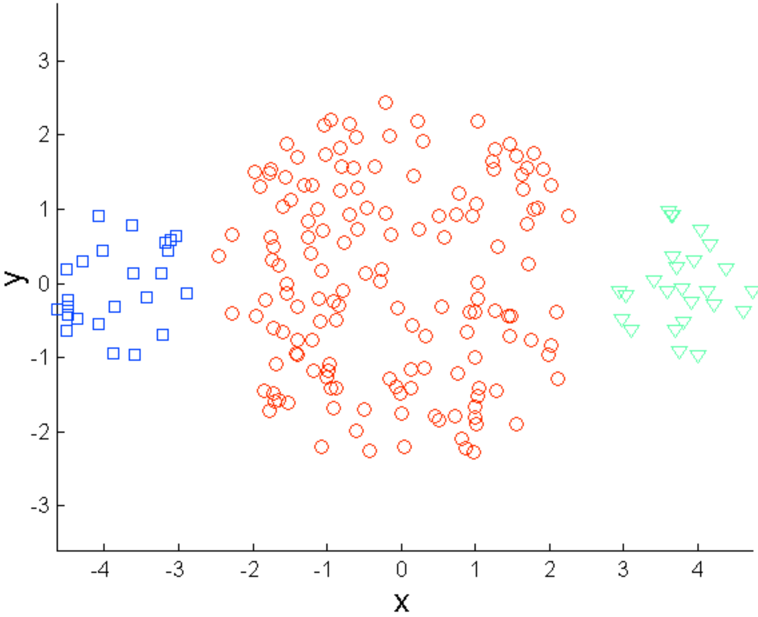
K均值聚类局限性 Limitations of K-means

K均值在以下情况时会遇到问题：

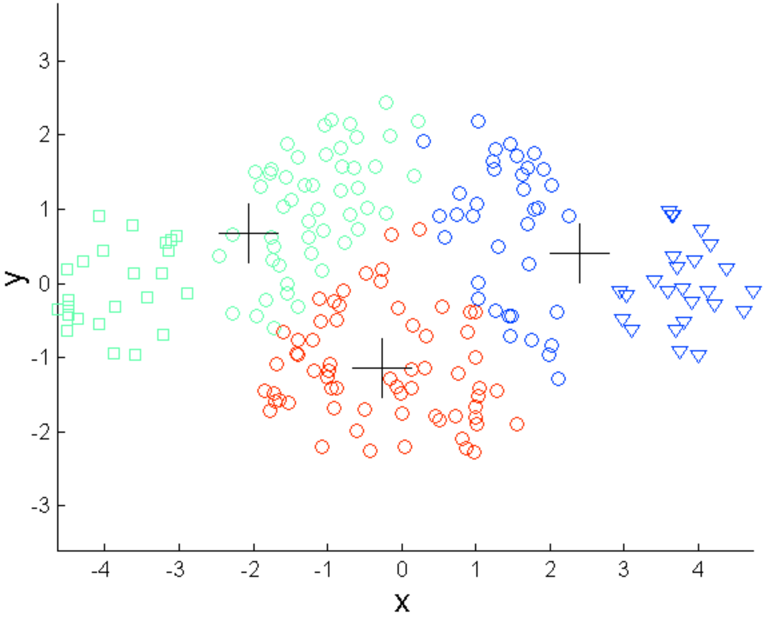
- 簇具有不同的大小 Sizes
- 簇具有不同的密度 Densities
- 簇不是球形的 Non-globular shapes

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

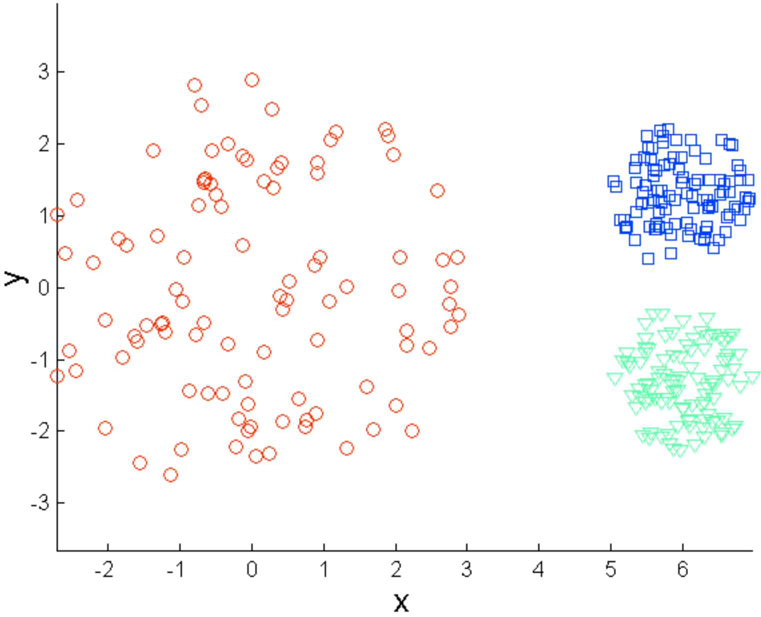


Original Points

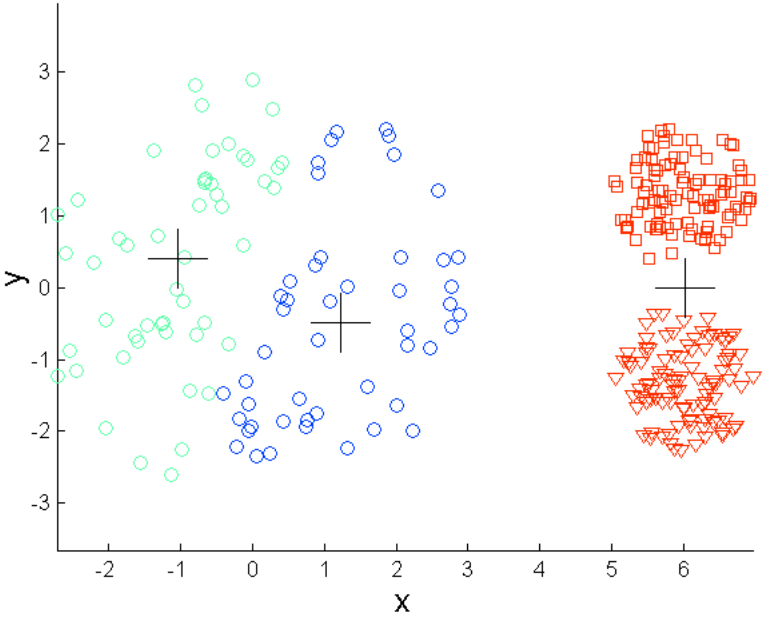


K-means (3 Clusters)

不同密度 Limitations of K-means: Differing Density

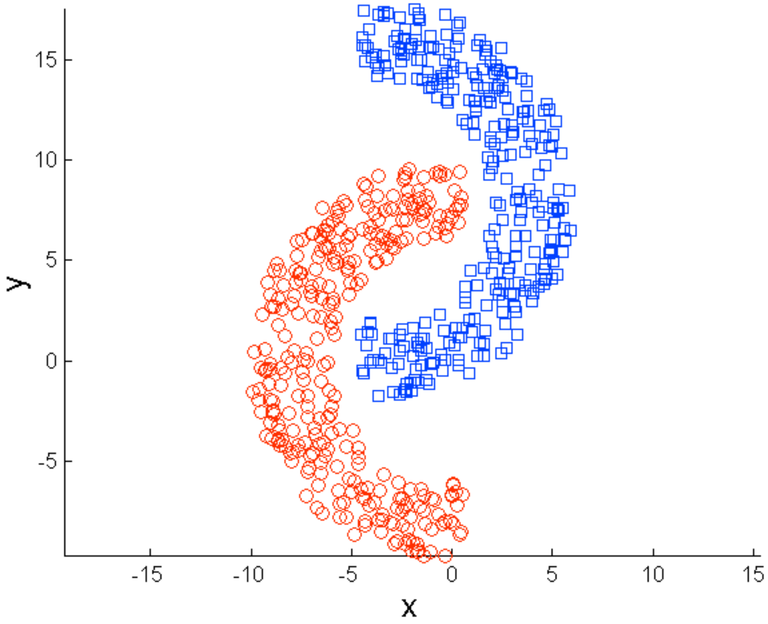


Original Points

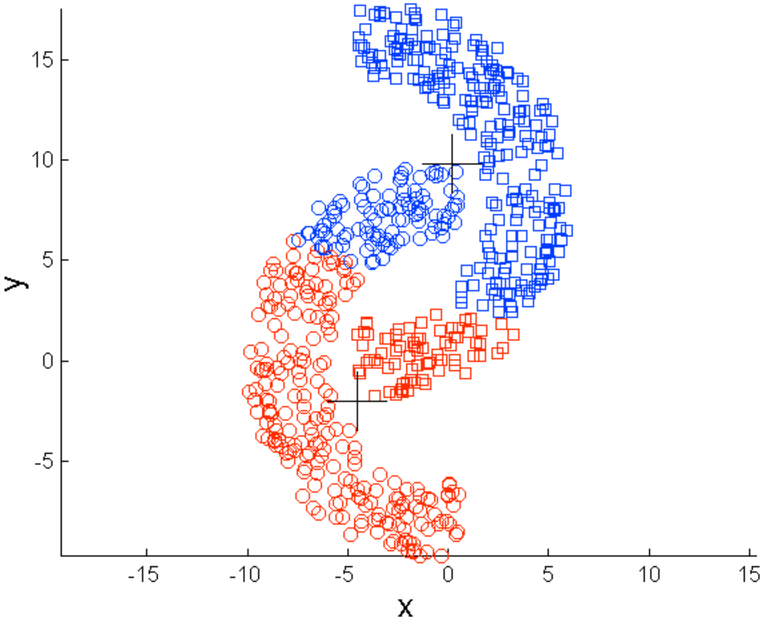


K-means (3 Clusters)

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

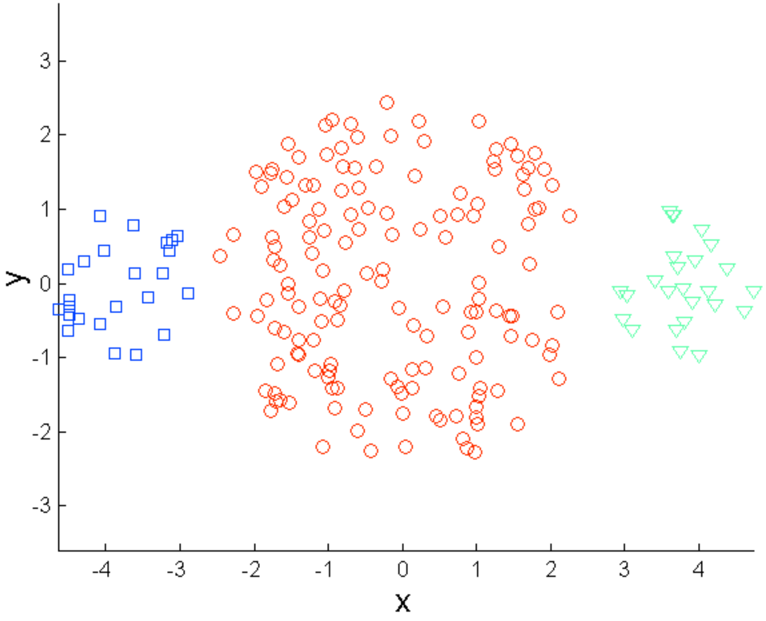


Original Points

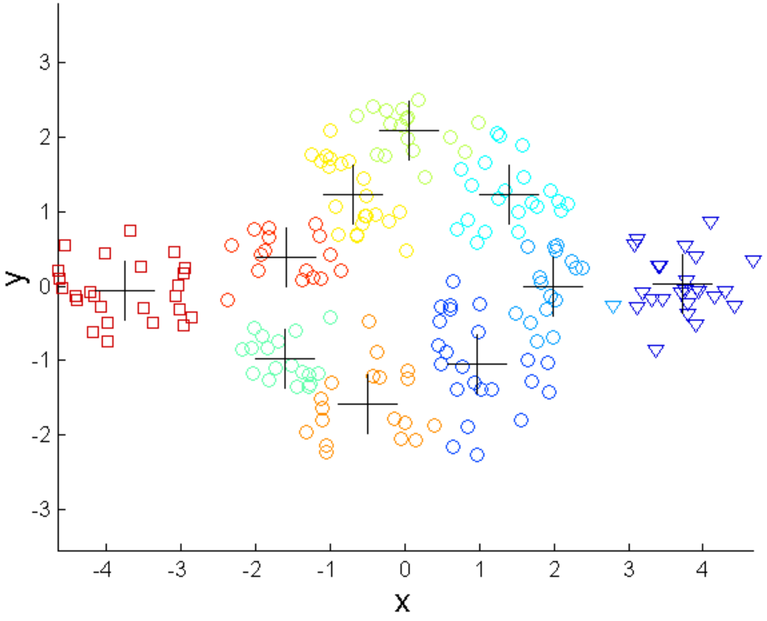


K-means (2 Clusters)

克服其局限性 Overcoming K-means Limitations



Original Points

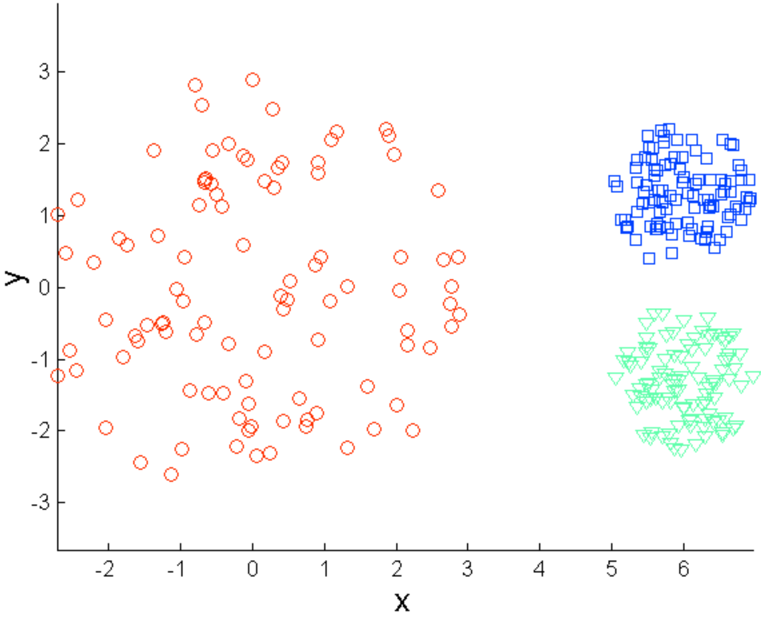


K-means Clusters

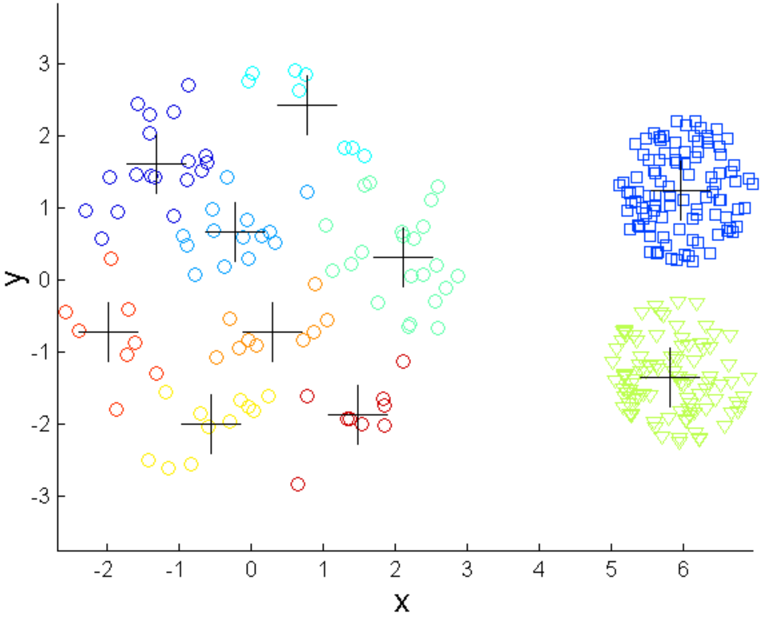
One solution is to use many clusters.

Find parts of clusters, but need to **put together**.

克服其局限性 Overcoming K-means Limitations

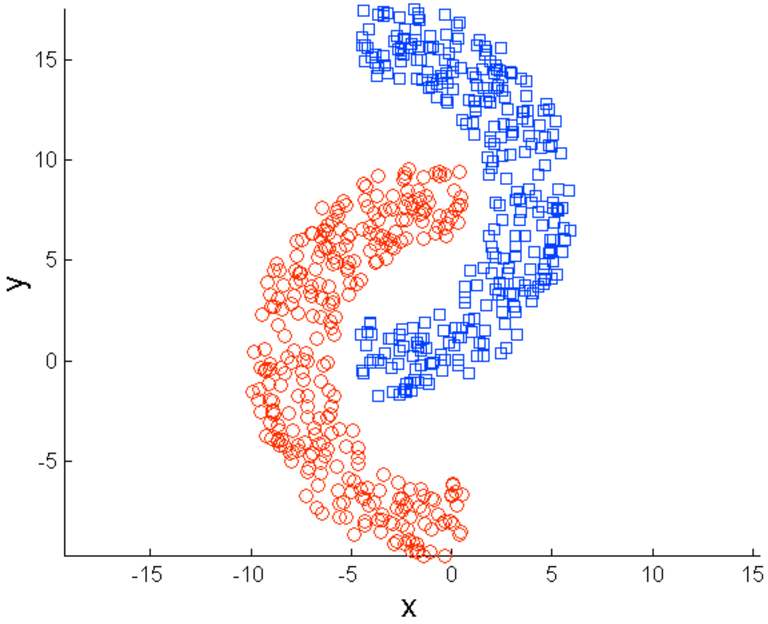


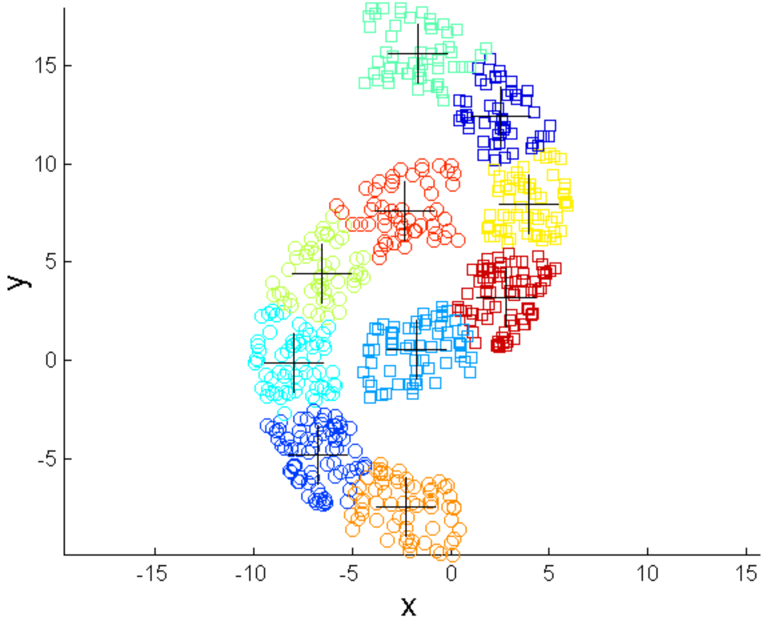
Original Points



K-means Clusters

克服其局限性 Overcoming K-means Limitations





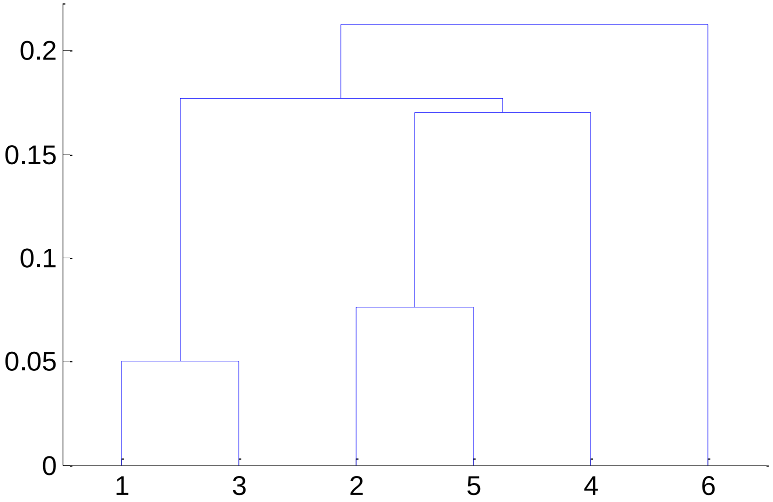
Original Points

K-means Clusters

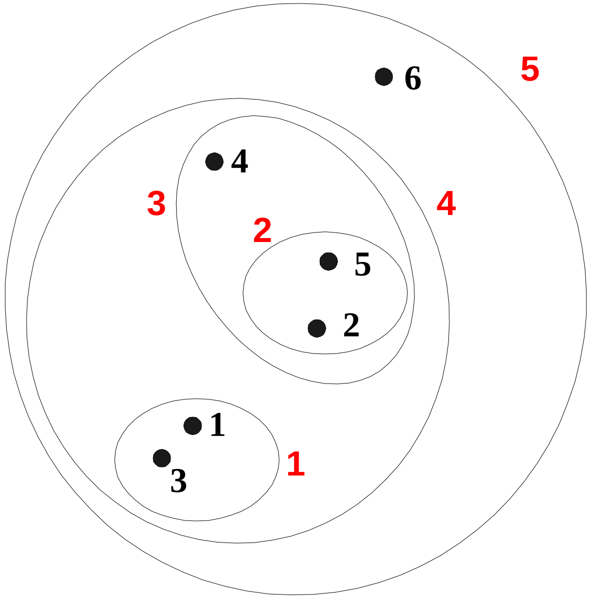
层次聚类 Hierarchical Clustering

产生一组嵌套的簇，这些簇被组织成一个层次树可以可视化为树状图（dendrogram）

- 类似于图的树，记录合并或拆分操作的序列。A tree like diagram that records the sequences of merges or splits



树状图（dendrogram）



嵌套簇图（nested cluster diagram）

优势：Strengths of Hierarchical Clustering

不必假设特定数量的簇（K值）

- 通过将树状图“切割（cutting）”到适当的水平可以获得任意数量的簇。

结果可能对应于有意义的分类法（taxonomies）

- 生物科学中的分类(e.g., animal kingdom, phylogeny reconstruction, …)

层次聚类方法 Hierarchical Clustering

层次聚类的两种主要类型

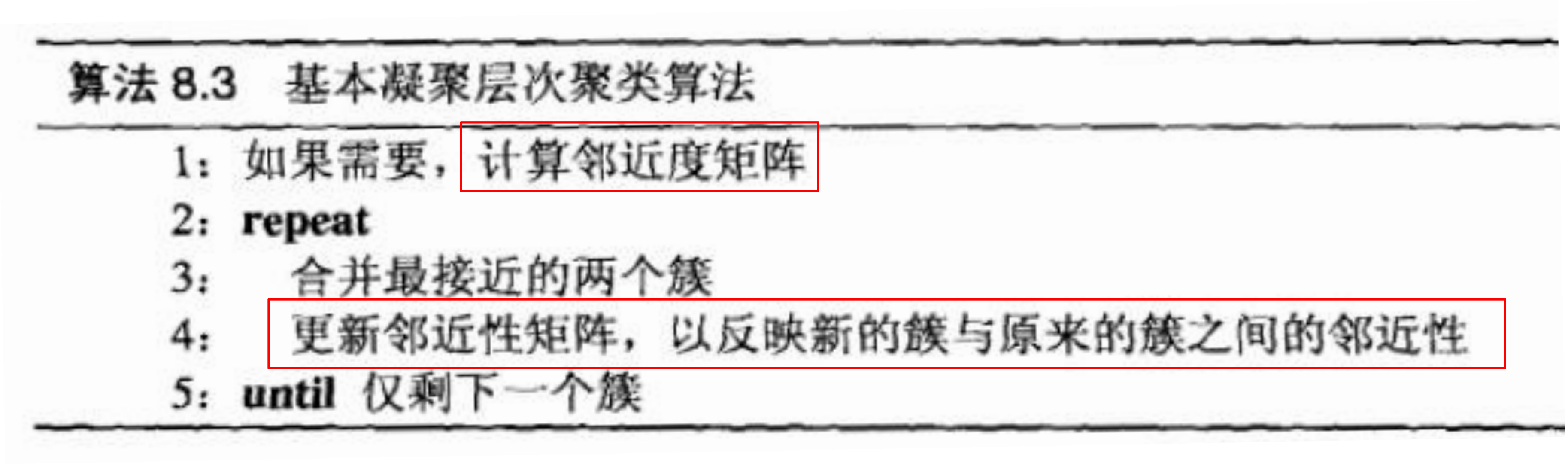
- **凝聚的 Agglomerative:**
  - Start with the points as individual clusters
  - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left

- **分裂的 Divisive:**
  - Start with one, all-inclusive cluster
  - At each step, split a cluster until each cluster contains an individual point (or there are k clusters)

Traditional hierarchical algorithms use a similarity or distance matrix

- Merge or split one cluster at a time

凝聚聚类 Agglomerative Clustering Algorithm

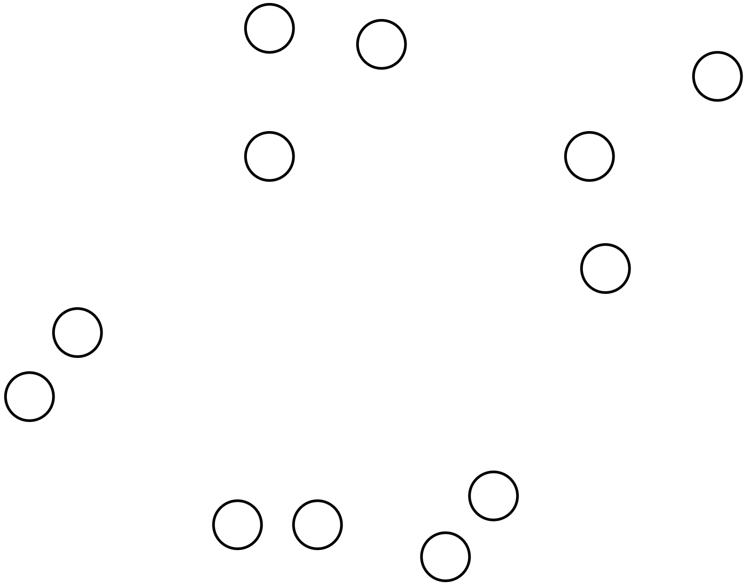


算法 8.3　基本凝聚层次聚类算法

```
如果需要，计算邻近度矩阵
repeat
    合并最接近的两个簇
    更新邻近性矩阵，以反映新的簇与原来的簇之间的邻近性
until 仅剩下一个簇
```

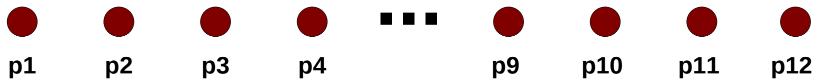
起始 Starting Situation

从单个点的簇和邻近矩阵（proximity matrix）开始



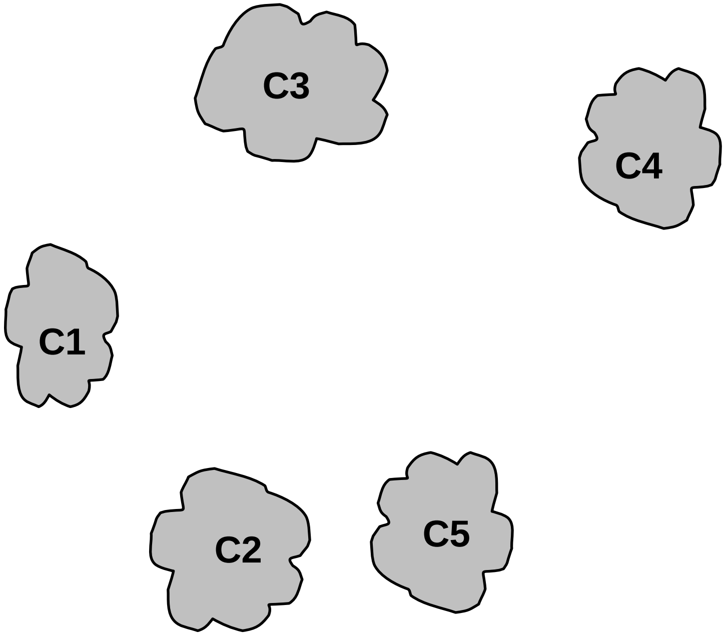
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix



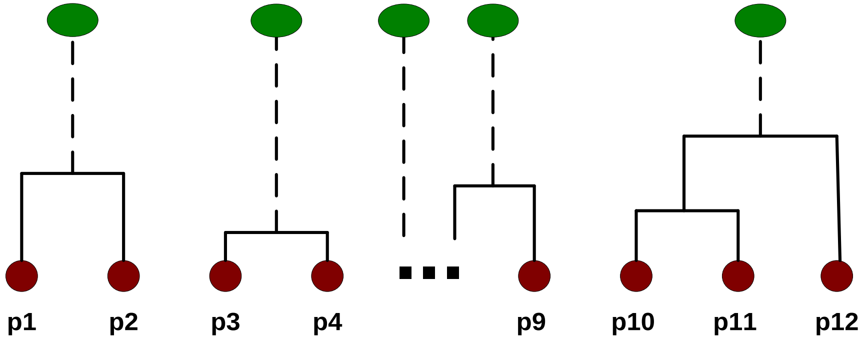
中间状态 Intermediate Situation

经过一些合并（merging），得到了一些簇（some clusters）



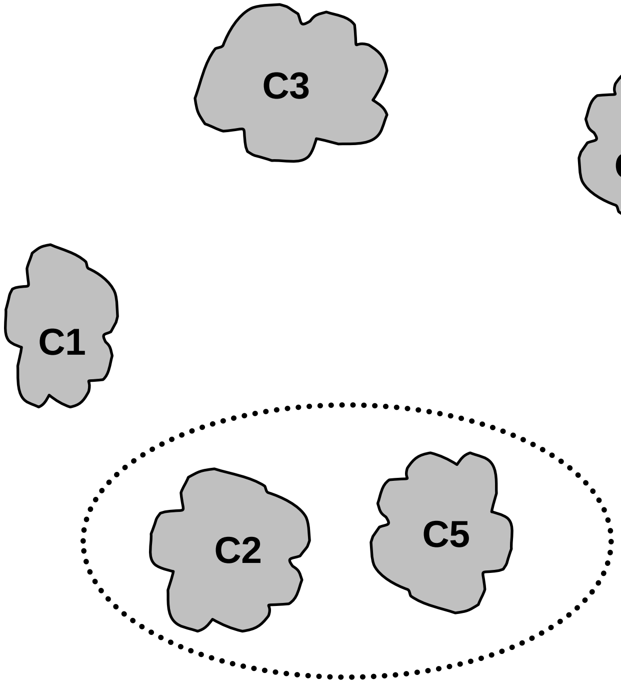
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

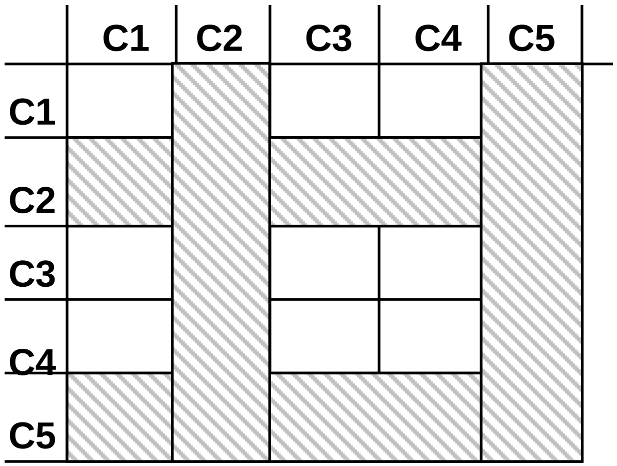
Proximity Matrix



中间状态 Intermediate Situation

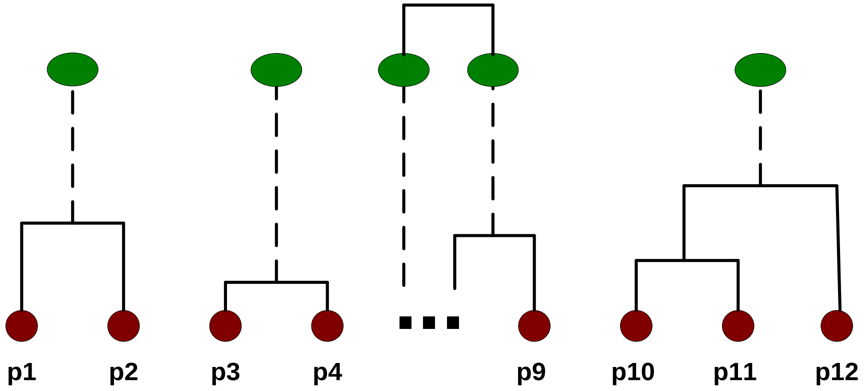
合并最近的簇（C2和C5）并更新邻近矩阵





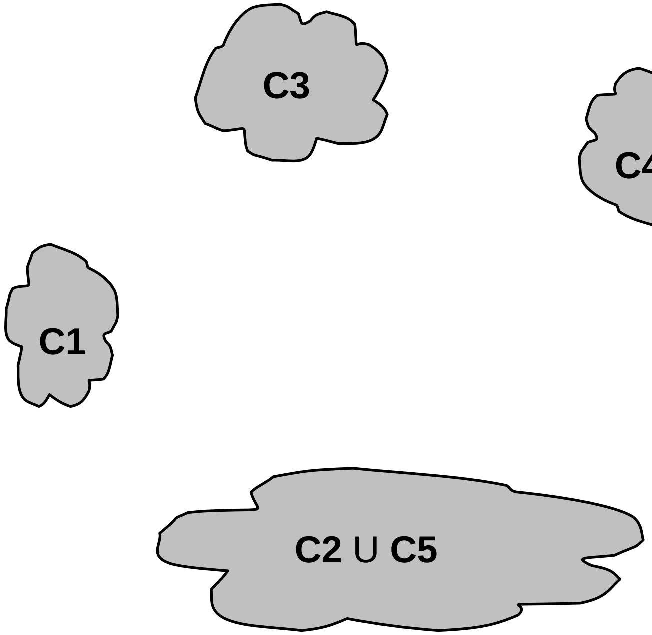
	C1	C2	C3	C4	C5
C1		▨			▨
C2	▨	▨	▨	▨	▨
C3		▨			▨
C4		▨			▨
C5	▨	▨	▨	▨	▨

Proximity Matrix



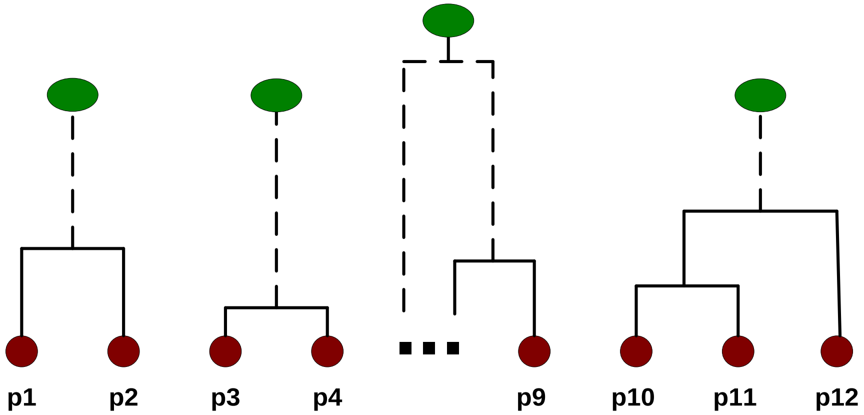
合并后 After Merging

如何合并邻近矩阵（簇之间的邻近性）

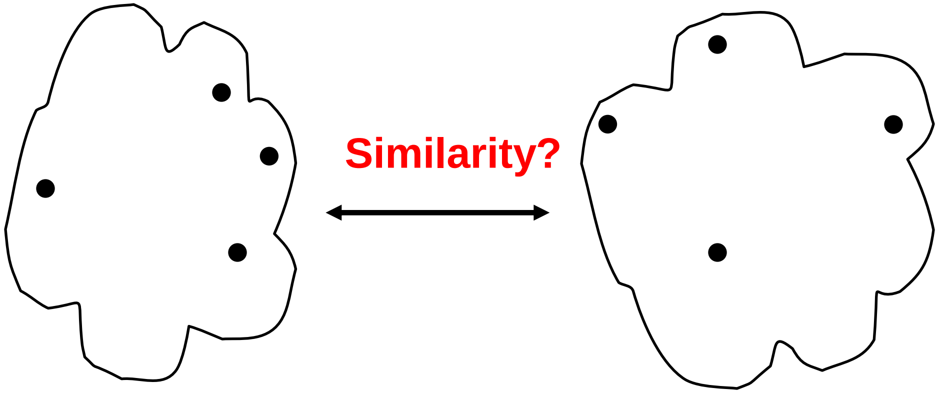


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



簇间距离：How to Define Inter-Cluster Distance

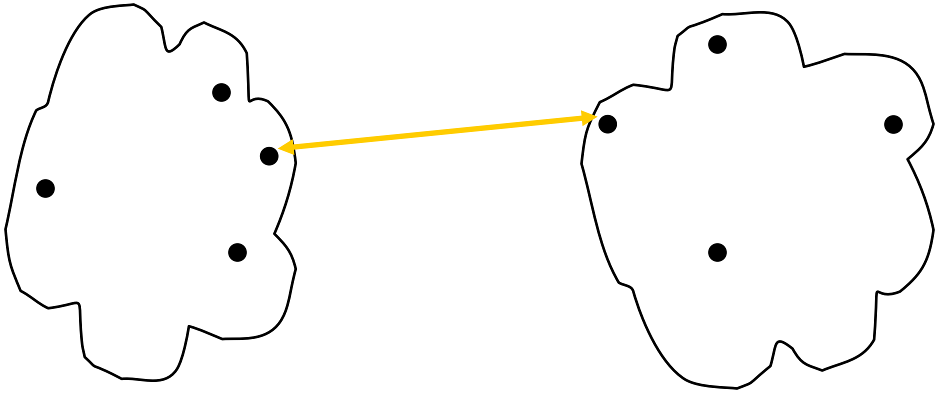


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
. . .						

Proximity Matrix

1. MIN
2. MAX
3. Group Average
4. Distance Between Centroids
5. Other methods driven by an objective function
   – Ward’s Method uses squared error

簇间相似：How to Define Inter-Cluster Similarity



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

.

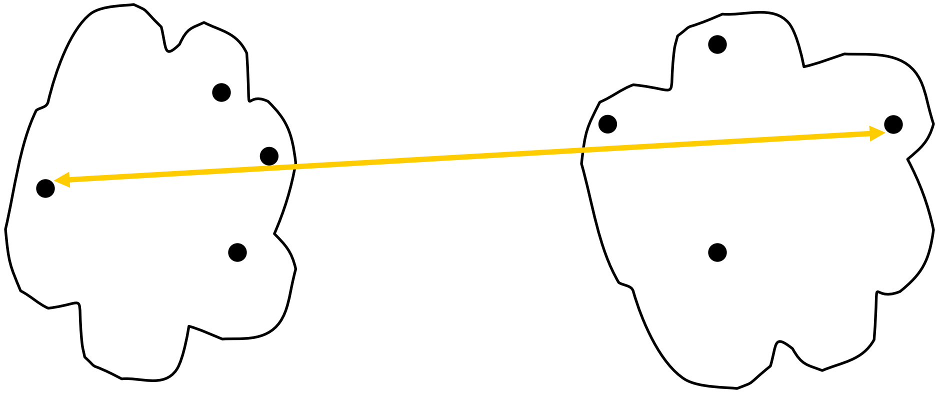
.

.

Proximity Matrix

1. MIN
2. MAX
3. Group Average
4. Distance Between Centroids
5. Other methods driven by an objective function
   – Ward’s Method uses squared error

How to Define Inter-Cluster Similarity

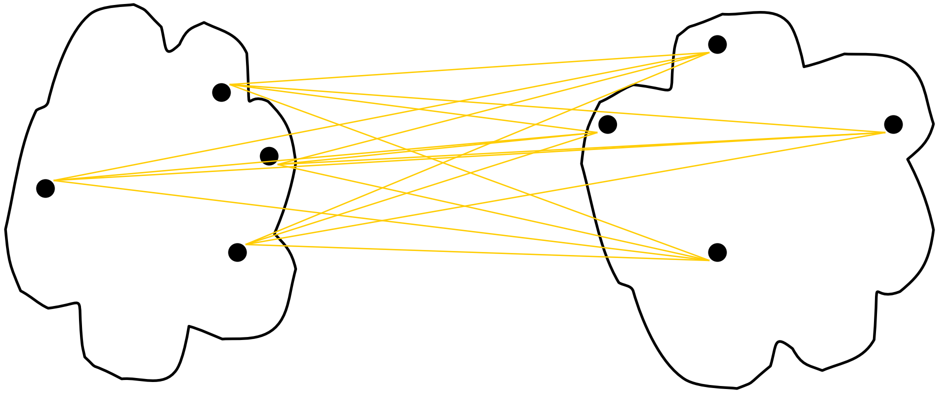


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
. . .						

Proximity Matrix

1. MIN
2. MAX
3. Group Average
4. Distance Between Centroids
5. Other methods driven by an objective function
   – Ward’s Method uses squared error

How to Define Inter-Cluster Similarity

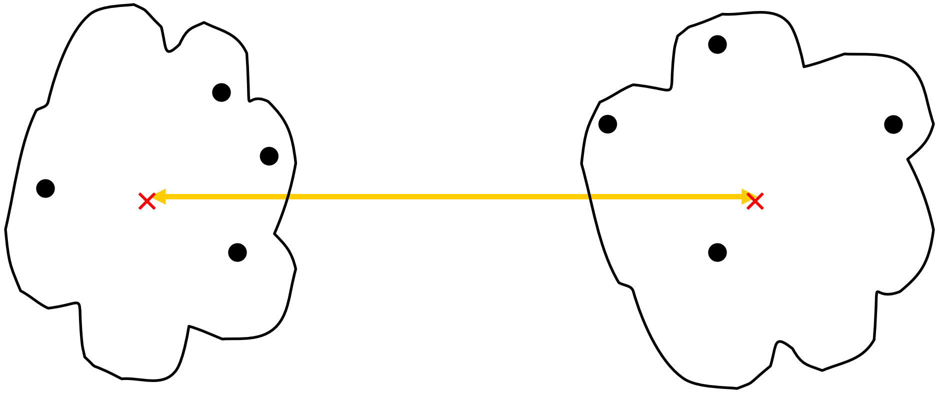


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
⋮						

Proximity Matrix

1. MIN
2. MAX
3. Group Average
4. Distance Between Centroids
5. Other methods driven by an objective function
   - Ward’s Method uses squared error

How to Define Inter-Cluster Similarity



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

Proximity Matrix

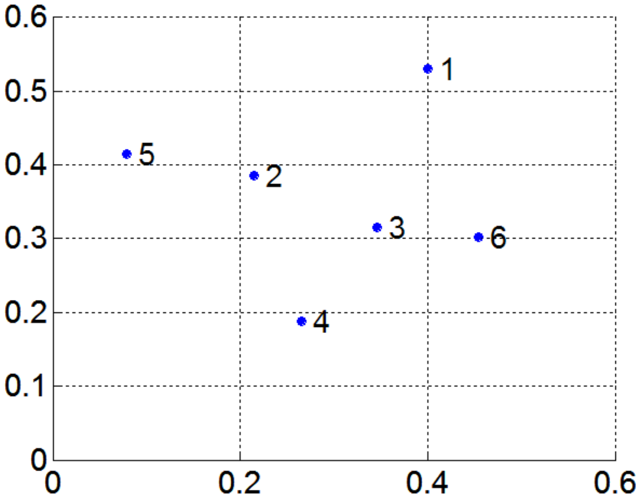
1. MIN
2. MAX
3. Group Average
4. Distance Between Centroids
5. Other methods driven by an objective function
   – Ward’s Method uses squared error

MIN or Single Link（单链）

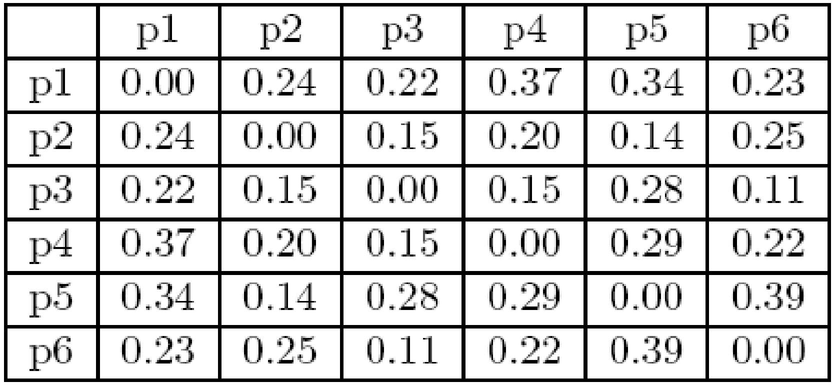
两个簇的邻近度基于同簇中的两个最接近点

- 由一对点确定，即由邻近图（proximity graph）中的一个链接（link）确定

示例:



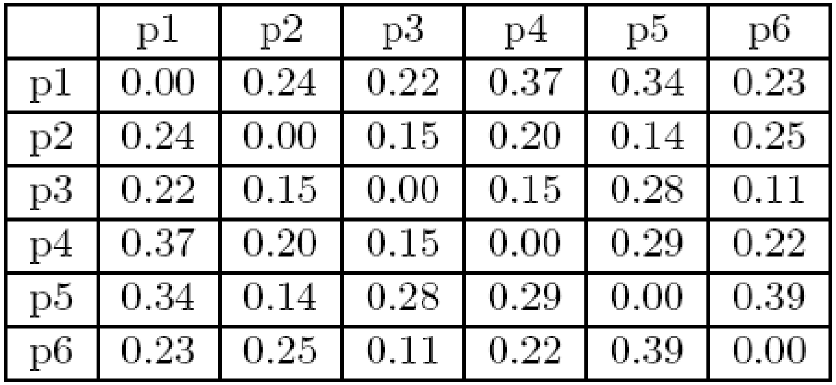
Distance Matrix:



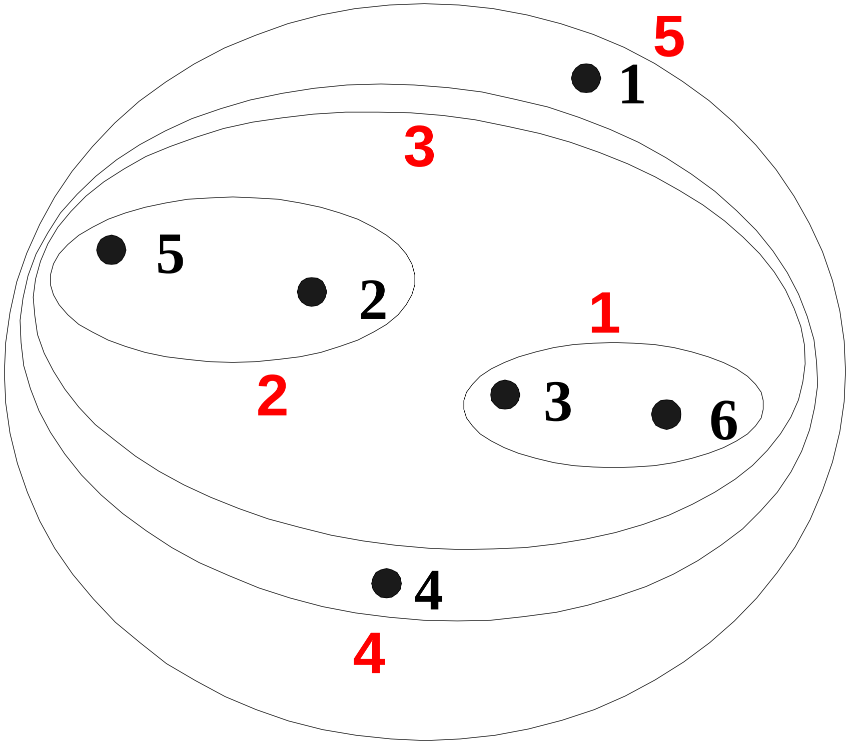
	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Hierarchical Clustering: MIN

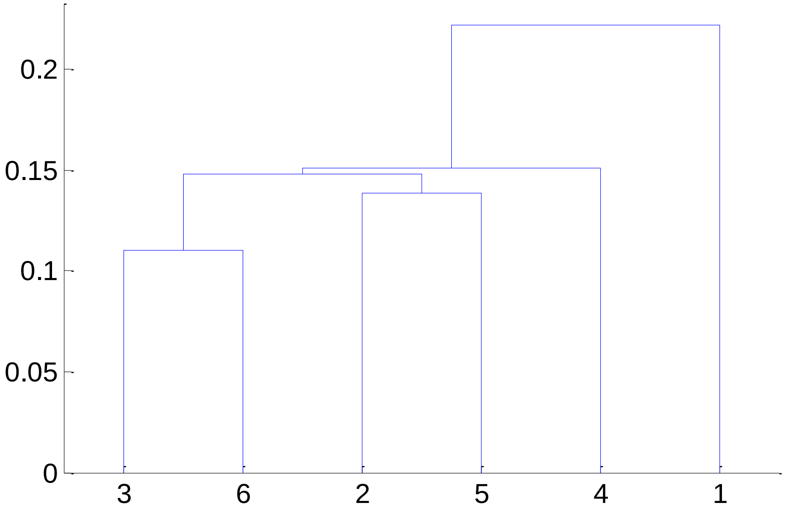
$$\begin{aligned} dist(\{3,6\},\{2,5\}) &= \min(dist(3,2), dist(6,2), dist(3,5), dist(6,5)) \\ &= \min(0.15, 0.25, 0.28, 0.39) \\ &= 0.15 \end{aligned}$$



	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

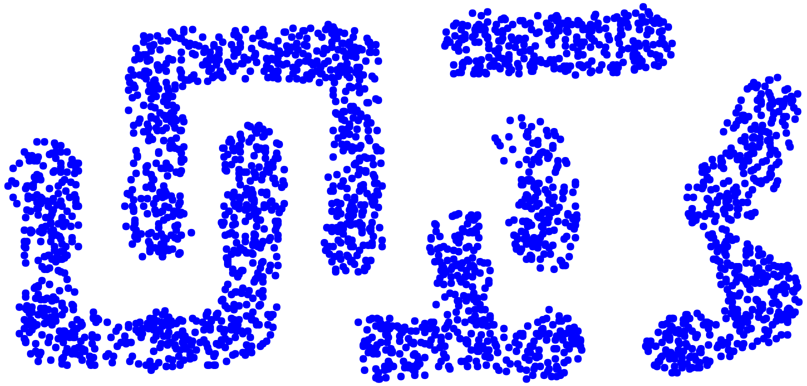


(a) Nested Clusters

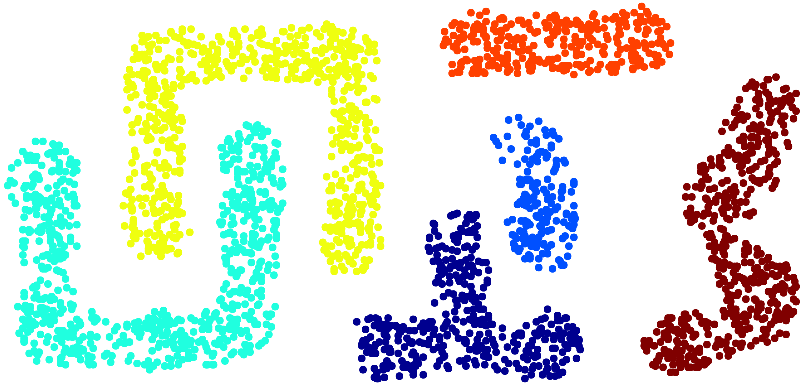


(b) Dendrogram（树状图）

优势：Strength of MIN



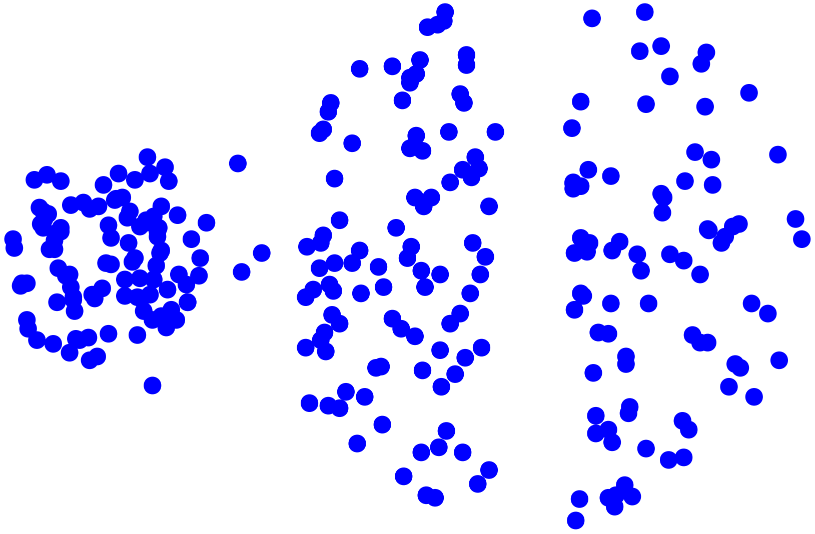
Original Points



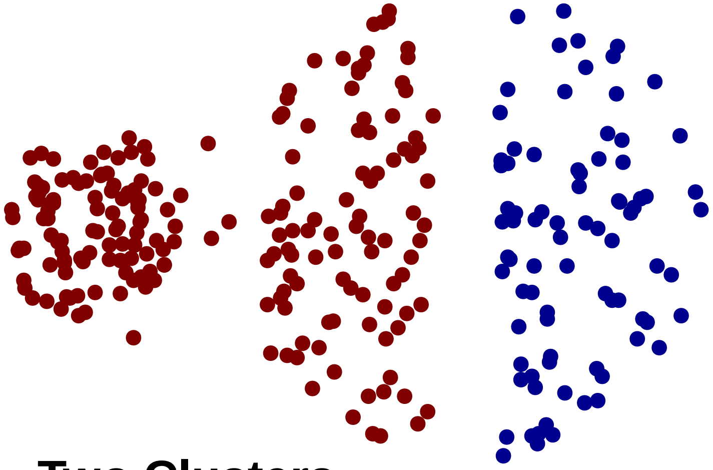
Six Clusters

- 单链技术擅长于处理非椭圆形状的簇。
- *Can handle non-elliptical shapes*

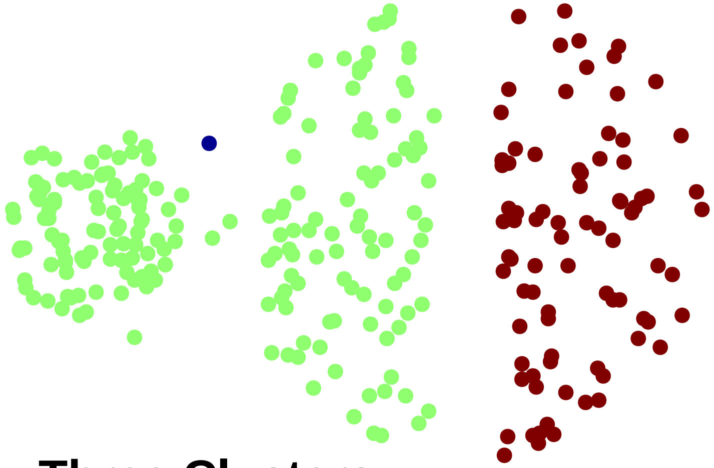
劣势：Limitations of MIN



Original Points



Two Clusters



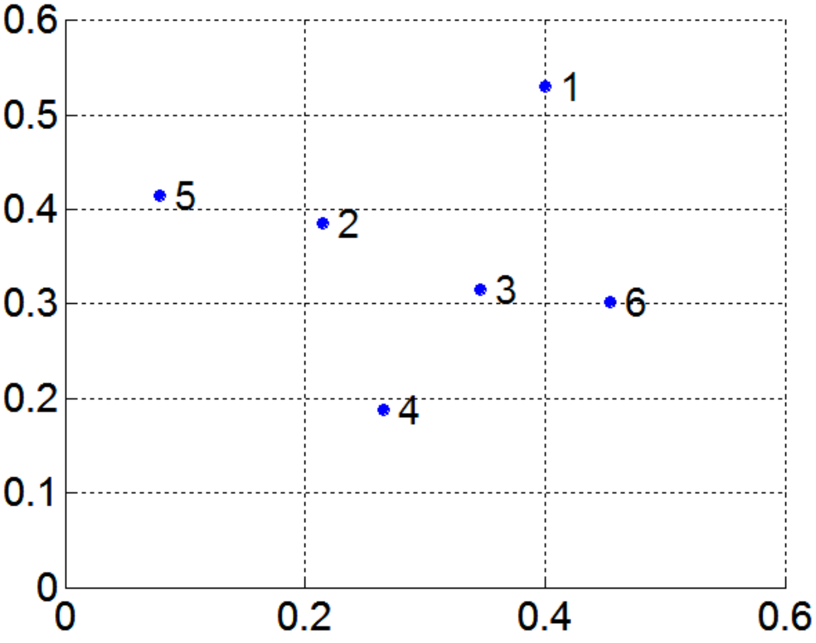
Three Clusters

- 对噪声点和离群点很敏感
- *Sensitive to noise and outliers*

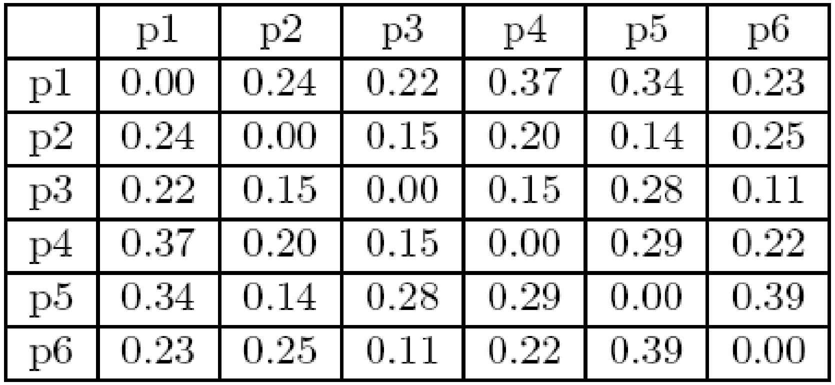
MAX or Complete Linkage（全链）

两个簇的邻近度定义为两个不同簇中任意两点之间的最长距离（最小相似度）

- 由两个簇中的所有点对决定。Determined by all pairs of points in the two clusters

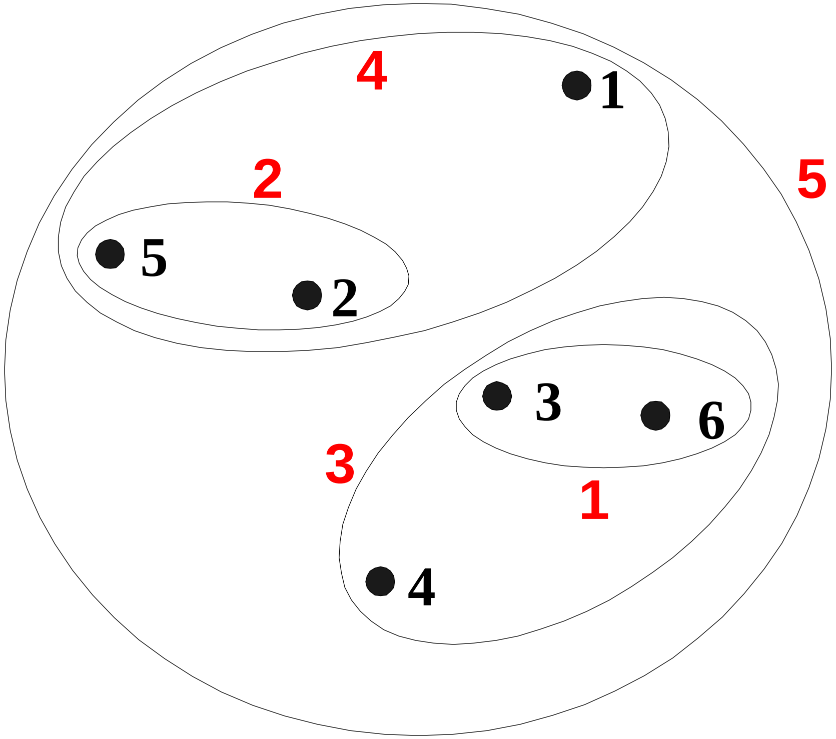


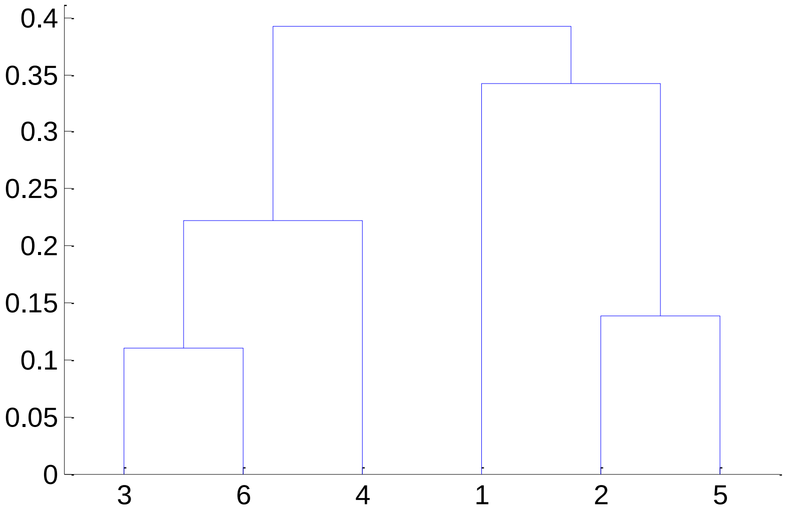
Distance Matrix:



	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Hierarchical Clustering: MAX

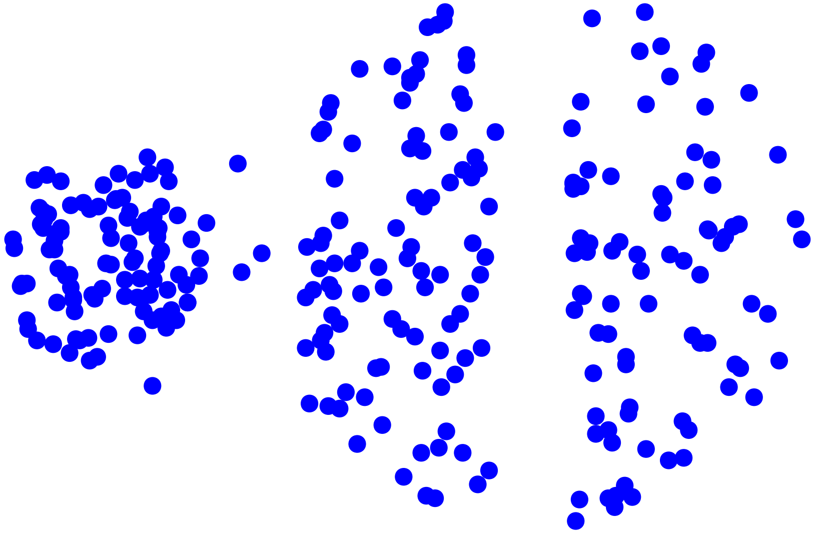




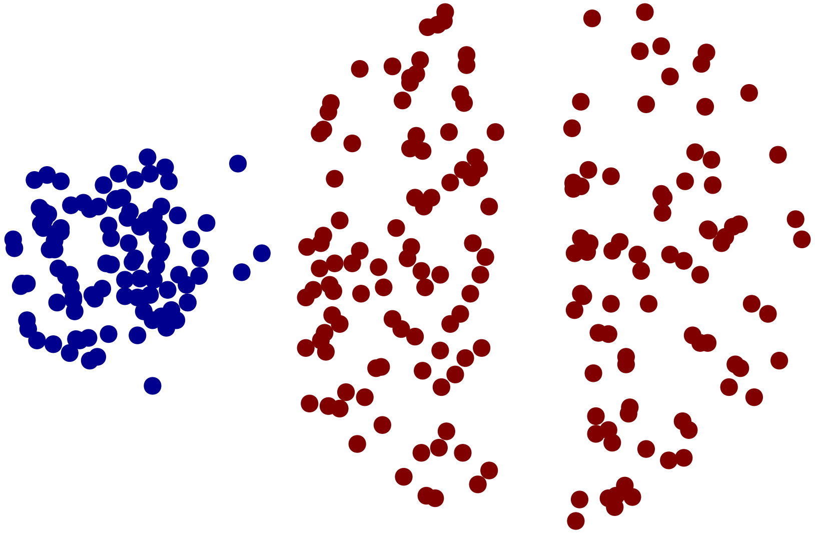
Nested Clusters

Dendrogram

优势：Strength of MAX



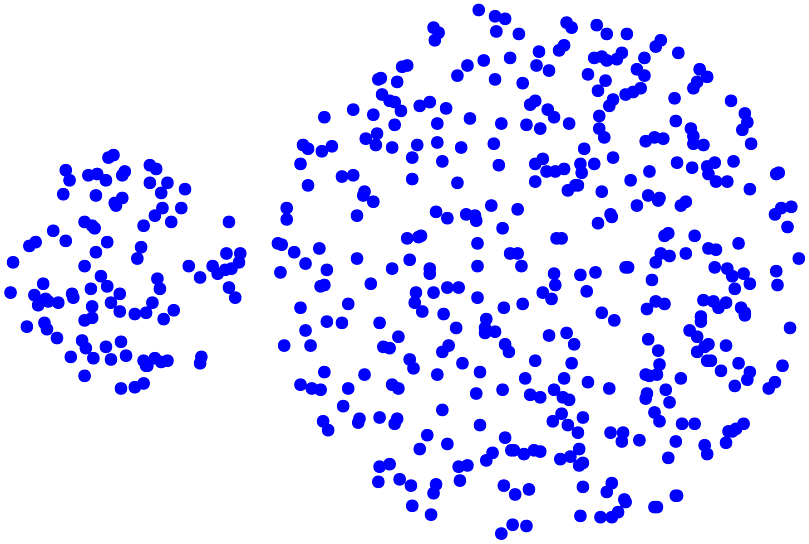
Original Points



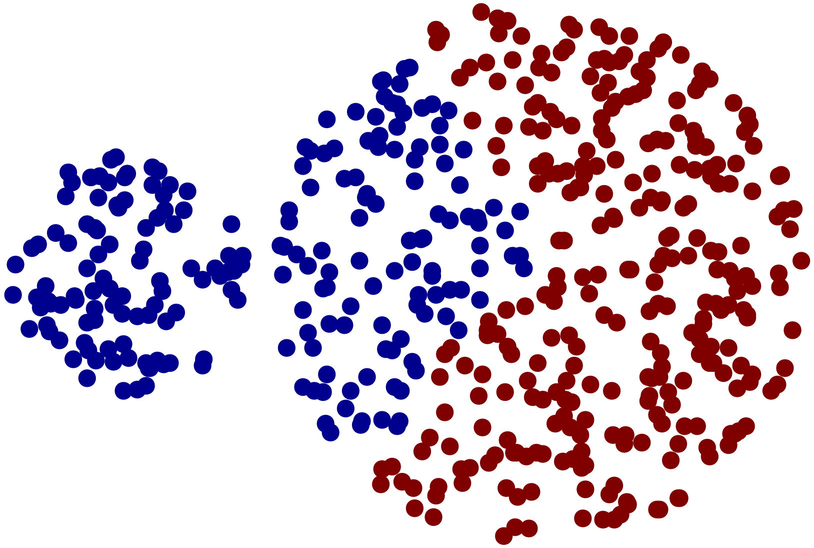
Two Clusters

- 全链对噪声和离群点不太敏感
- *Less susceptible to noise and outliers*

Limitations of MAX



Original Points



Two Clusters

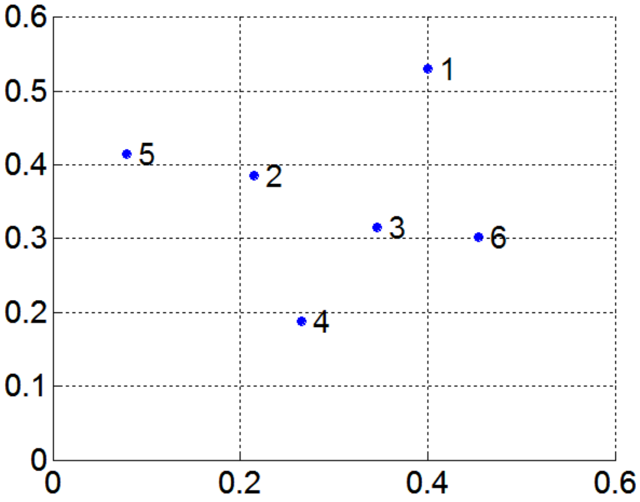
- 可能使大的簇破裂。Tends to break large clusters
- 偏好球形。Biased towards globular clusters

组平均 Group Average

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

$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum\limits_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| \times |\text{Cluster}_j|}$$

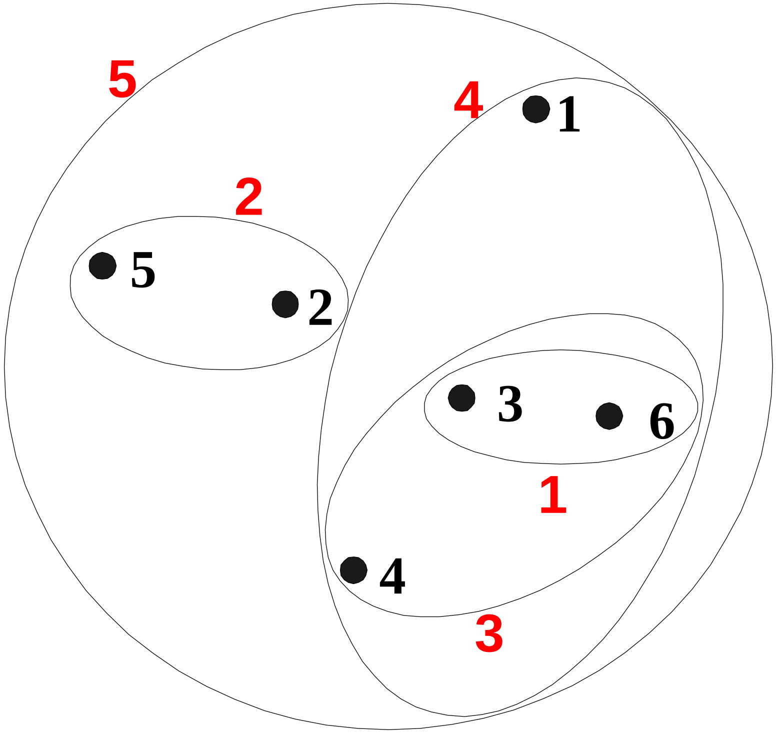
Need to use average connectivity for scalability since total proximity favors large clusters

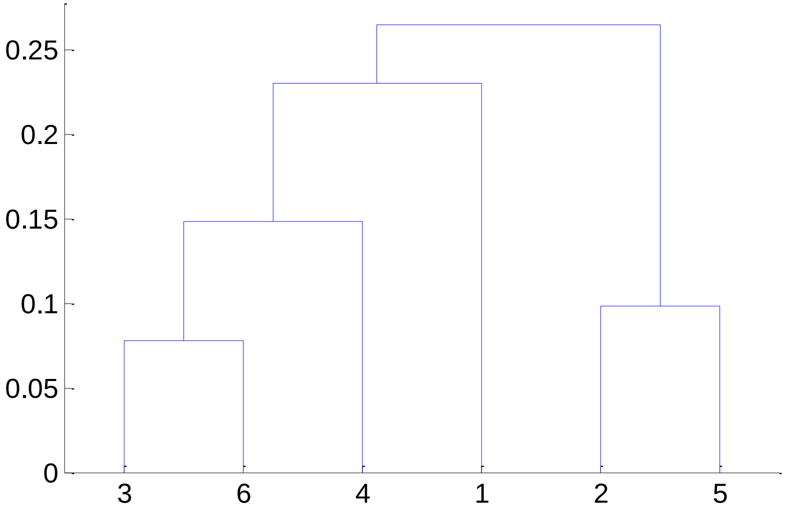


Distance Matrix:

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Hierarchical Clustering: Group Average





Nested Clusters

Dendrogram

Hierarchical Clustering: Group Average

单链和全链之间的妥协

优势

- 噪声不敏感。Less susceptible to noise and outliers

劣势

- 倾向于形成球形簇。Biased towards globular clusters

Cluster Similarity: Ward’s Method（自学）

Similarity of two clusters is based on the increase in squared error when two clusters are merged

- Similar to group average if distance between points is distance squared

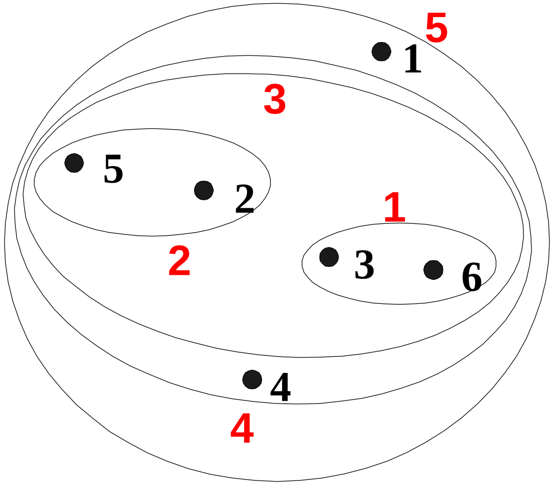
Less susceptible to noise and outliers

Biased towards globular clusters

Hierarchical analogue of K-means

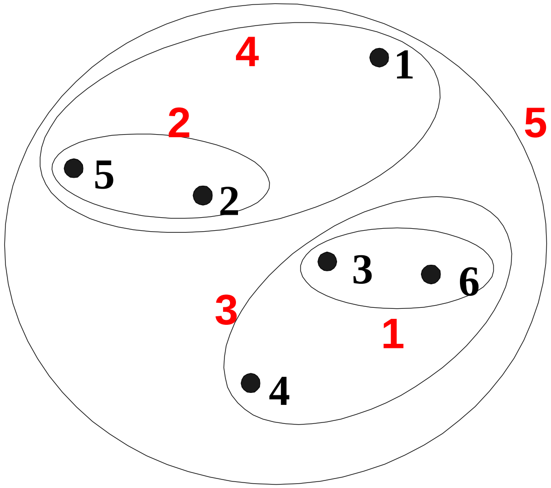
- Can be used to initialize K-means

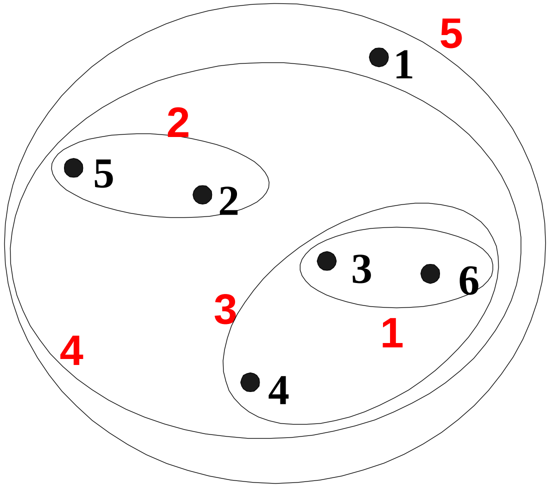
Hierarchical Clustering: Comparison



MIN

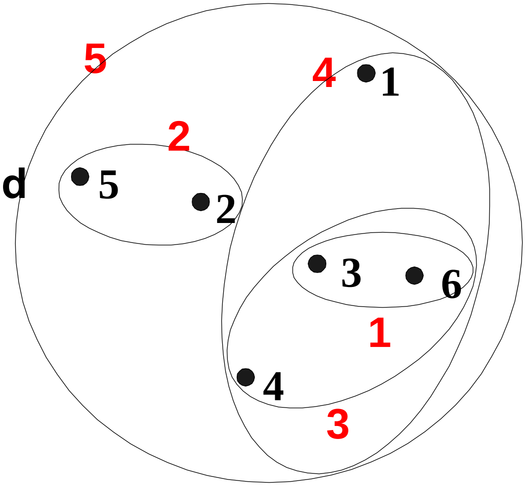
MAX





Group Average

Ward’s Method



层次聚类：时空复杂度
Hierarchical Clustering: Time and Space requirements

- 空间复杂度：O(N^2) space since it uses the proximity matrix.
  - N is the number of points.

- 时间复杂度：O(N^3) time in many cases
  - There are N steps and at each step the size, N^2, proximity matrix must be updated and searched
  - Complexity can be reduced to O(N^2 log(N)) time with some cleverness

层次聚类的空间和时间复杂度严重地限制了它所能够处理的数据集的大小

层次聚类：问题与局限性
Hierarchical Clustering: Problems and Limitations

合并决策不可撤销：Once a decision is made to combine two clusters, it cannot be undone

无全局目标函数：No global objective function is directly minimized

其他问题：Different schemes have problems with one or more of the following:

- Sensitivity to noise and outliers
- Difficulty handling clusters of different sizes and non-globular shapes
- Breaking large clusters

基于密度的聚类：DBSCAN

DBSCAN 是基于密度（density-based）的算法

- 基于中心的密度（半径）
- Density = number of points within a specified radius (Eps)

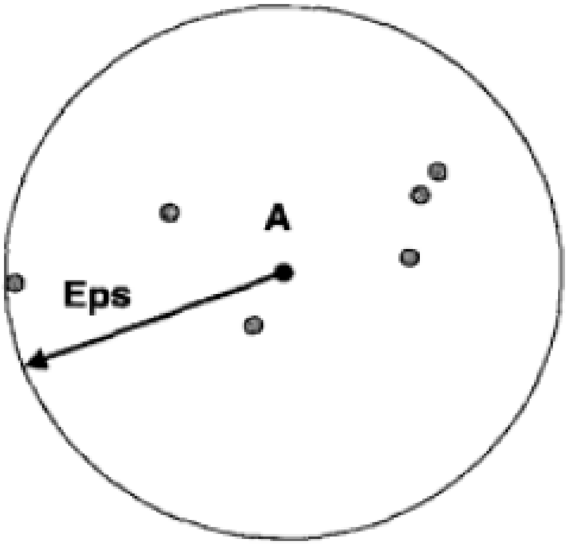


图 8-20　基于中心的密度

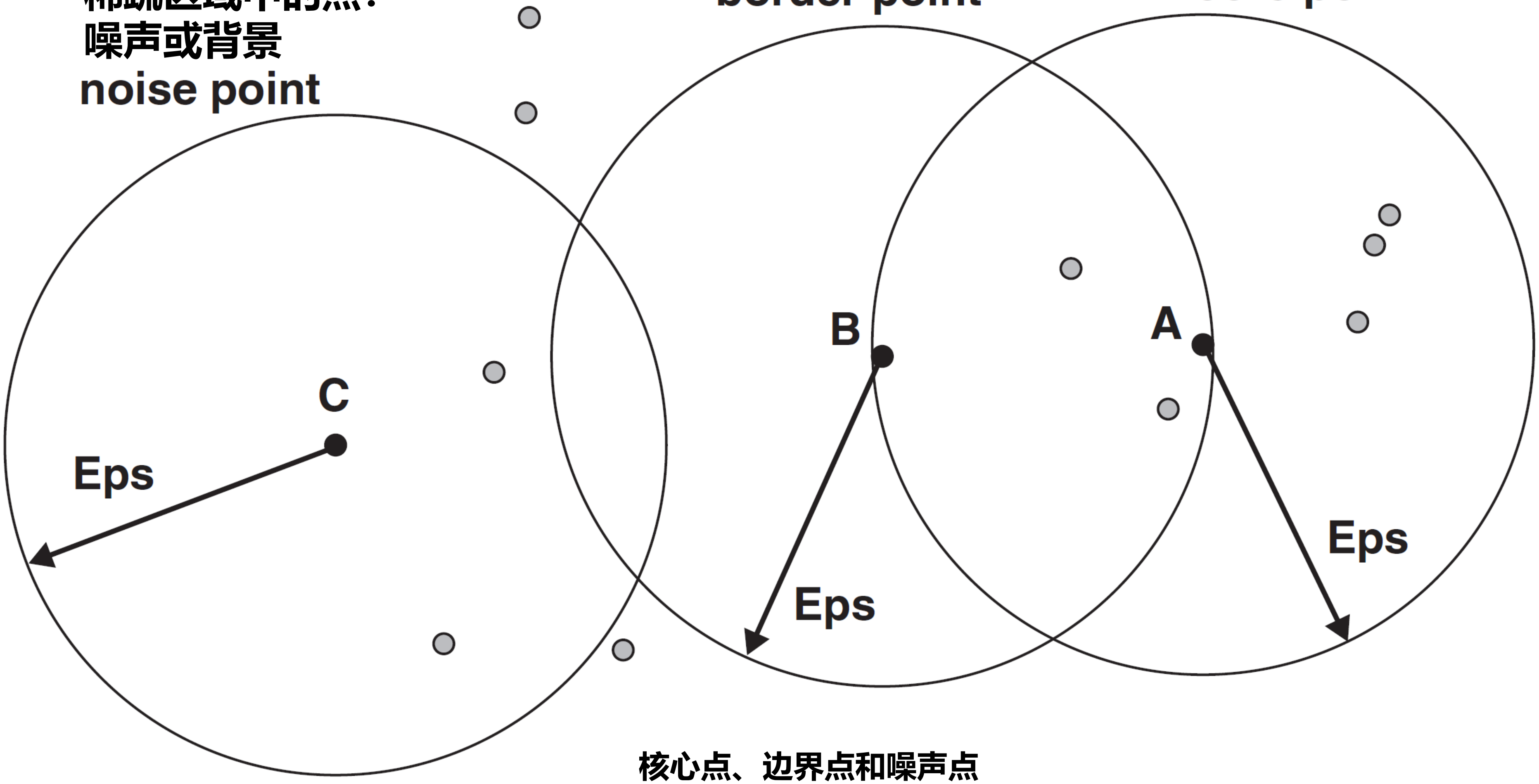
DBSCAN: Core, Border, and Noise Points

MinPts = 7

稀疏区域中的点：
噪声或背景
noise point

稠密区域边缘上的
点：边界点
border point

稠密区域内部的
点：核心点
core point



核心点、边界点和噪声点

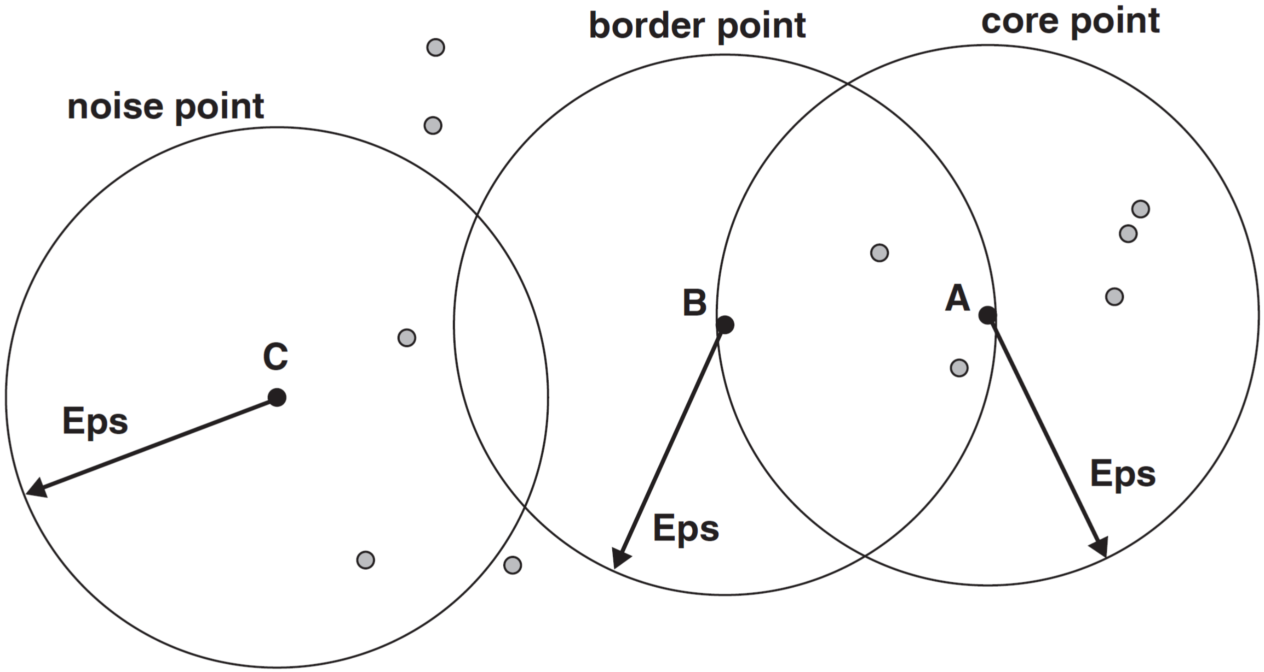
基于密度的聚类：DBSCAN

核心点：A point is a core point if it has at least a specified number of points (MinPts) within Eps

- These are points that are at the interior of a cluster
- Counts the point itself

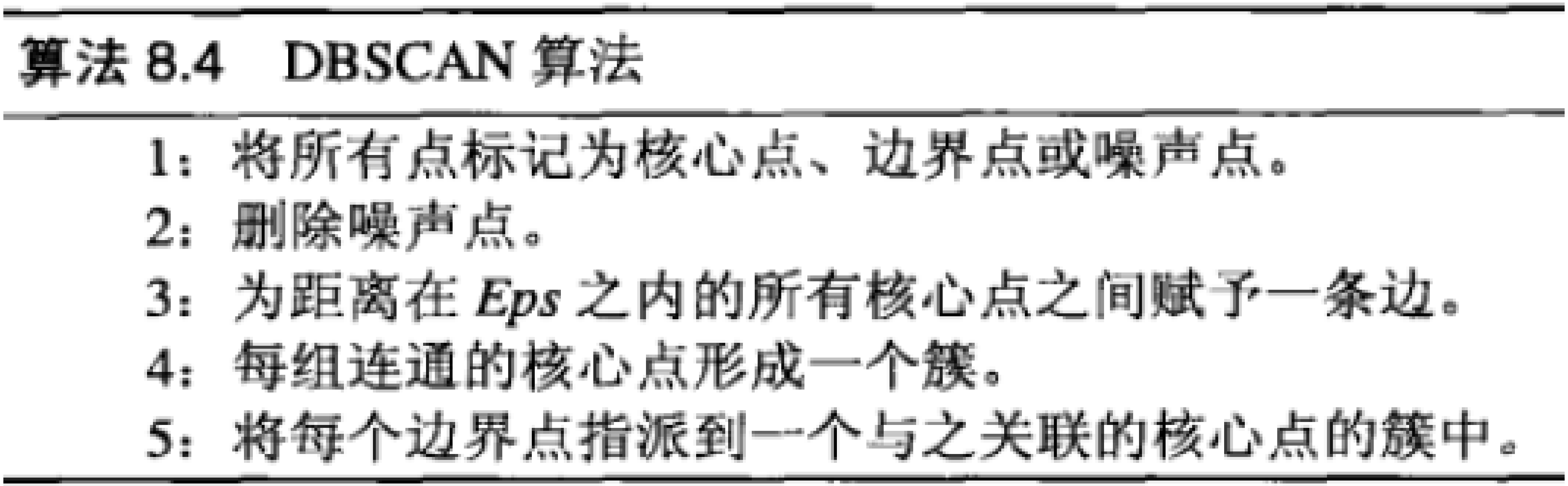
边界点：A border point is not a core point, but is in the neighborhood of a core point

噪声点：A noise point is any point that is not a core point or a border point



MinPts = 7

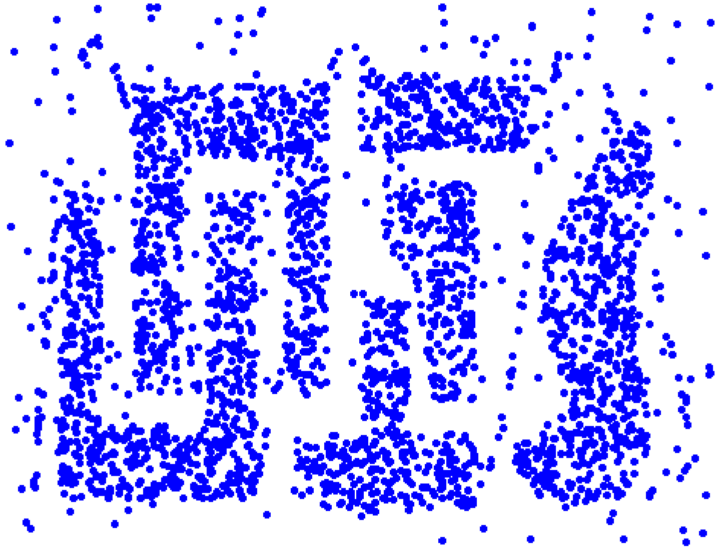
DBSCAN 算法



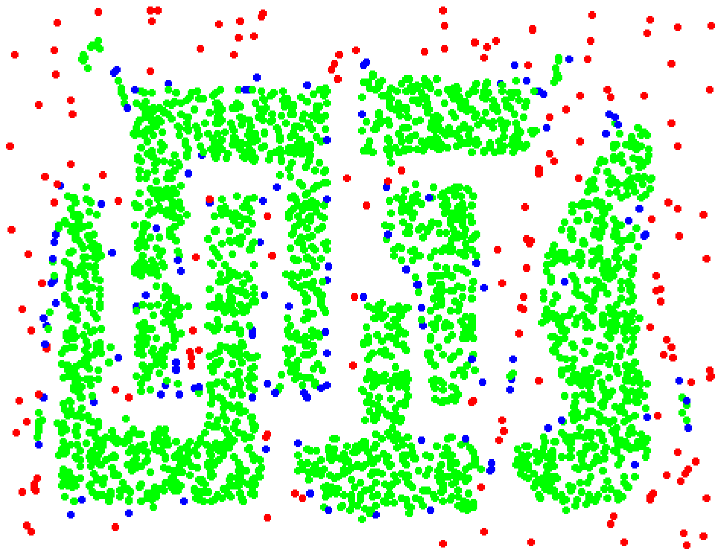
算法 8.4　DBSCAN 算法

```
将所有点标记为核心点、边界点或噪声点。
删除噪声点。
为距离在 Eps 之内的所有核心点之间赋予一条边。
每组连通的核心点形成一个簇。
将每个边界点指派到一个与之关联的核心点的簇中。
```

DBSCAN: Core, Border and Noise Points



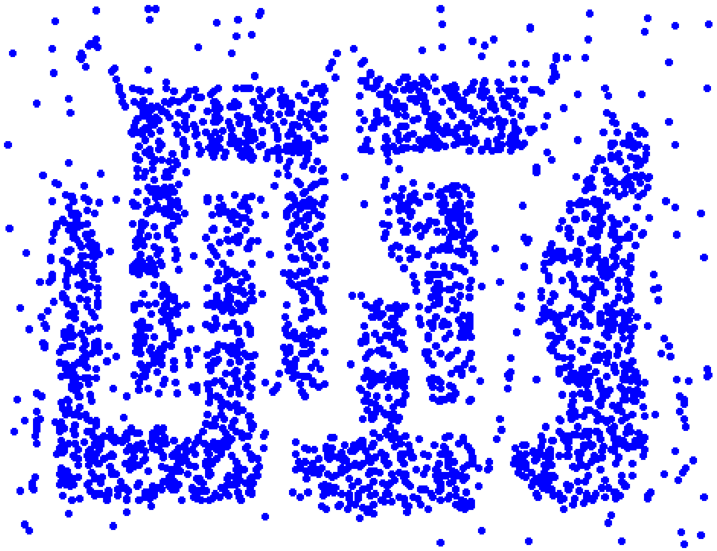
Original Points



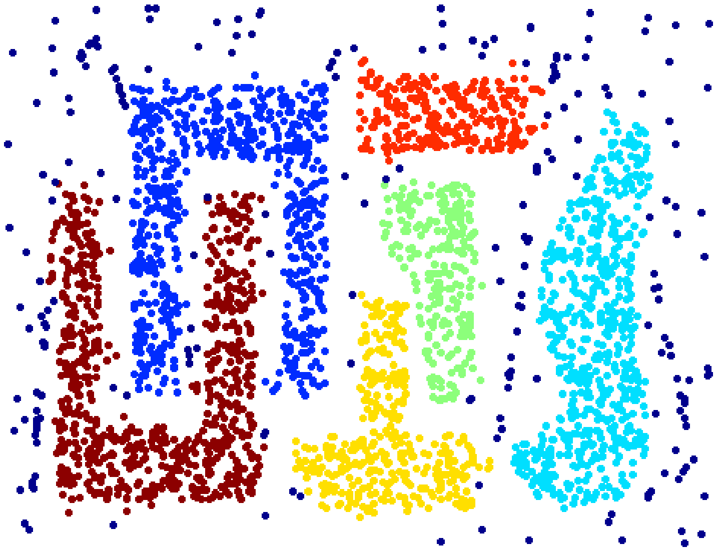
Point types: core, border and noise

Eps = 10, MinPts = 4

When DBSCAN Works Well



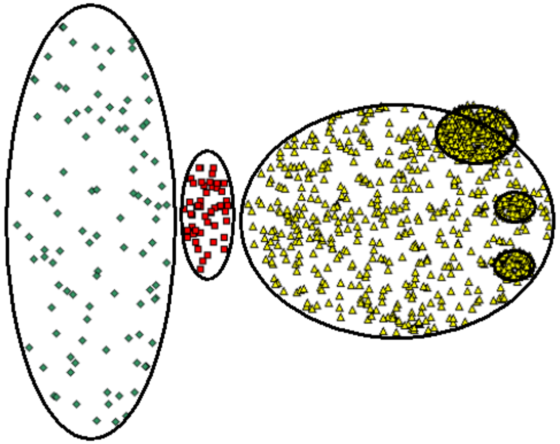
Original Points



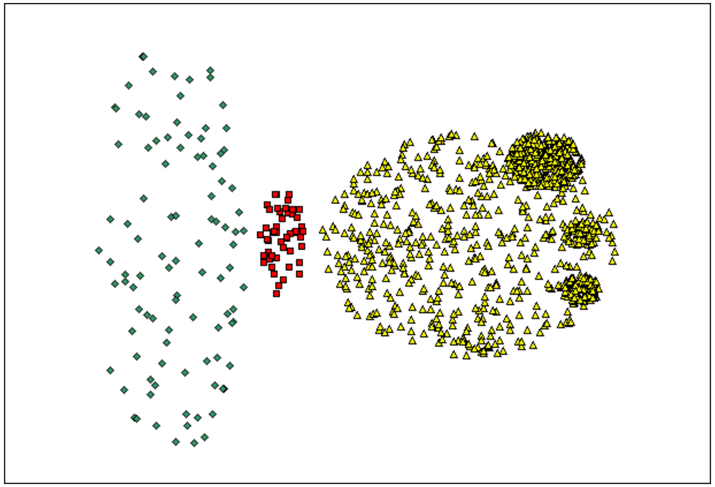
Clusters

- 抗噪声。Resistant to Noise
- 处理不同形状和大小。Can handle clusters of different shapes and sizes

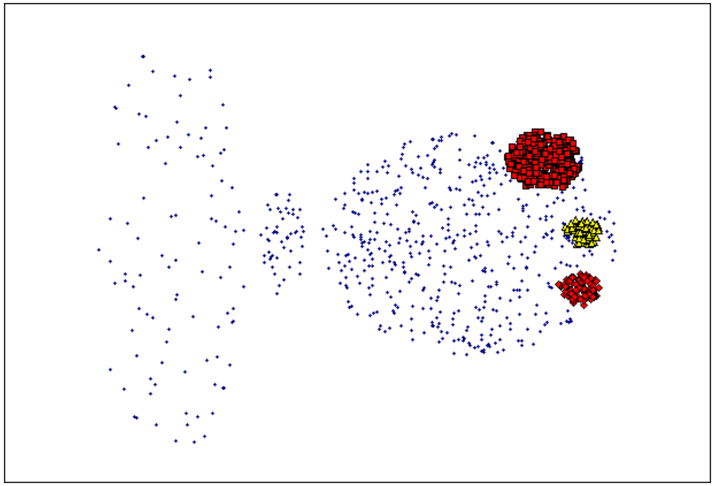
When DBSCAN Does NOT Work Well



Original Points



(MinPts=4, Eps=9.75).



(MinPts=4, Eps=9.92)

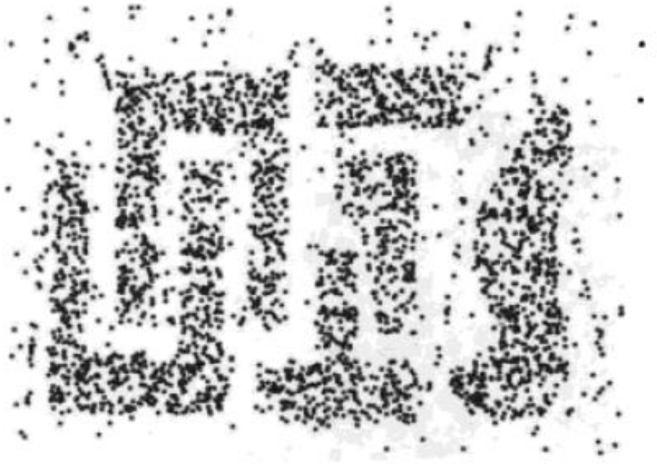
- 密度变化太大。Varying densities
- 高维数据。High-dimensional data

参数设置：DBSCAN: Determining EPS and MinPts

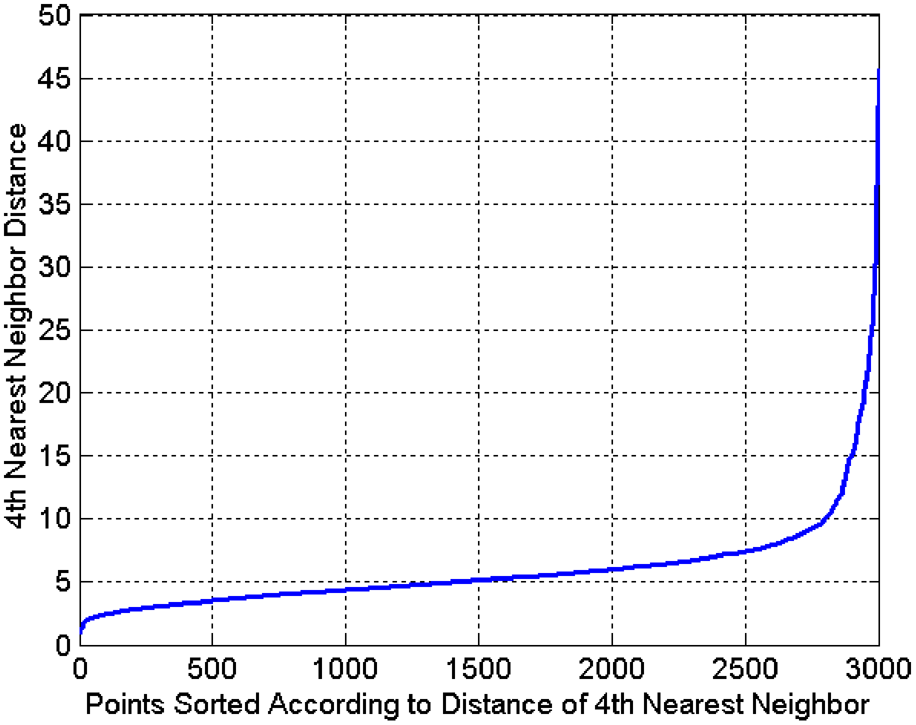
Idea is that for points in a cluster, their k^{th} nearest neighbors are at roughly the same distance

Noise points have the k^{th} nearest neighbor at farther distance

So, plot sorted distance of every point to its k^{th} nearest neighbor



样本数据



k-距离图

簇的有效性 Cluster Validity

对于监督分类（supervised classification），我们有多种方法可以评估模型的质量

- 准确性，精确度，召回率：Accuracy, precision, recall

对于聚类分析，类似的问题是如何评估所得聚类好不好?

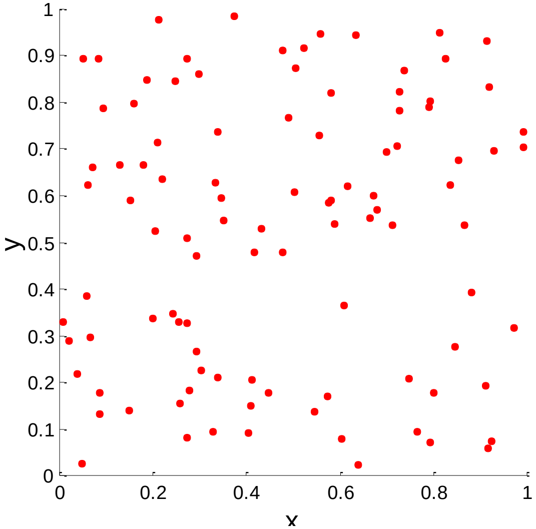
- 主观性很强

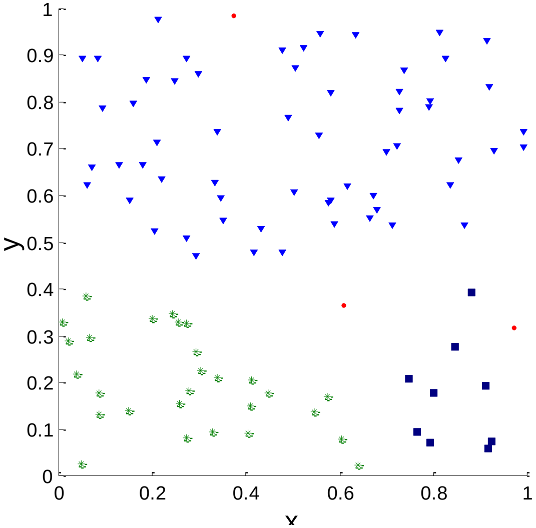
Then why do we want to evaluate them?

- To avoid finding patterns in noise
- To compare clustering algorithms
- To compare two sets of clusters
- To compare two clusters

Clusters found in Random Data

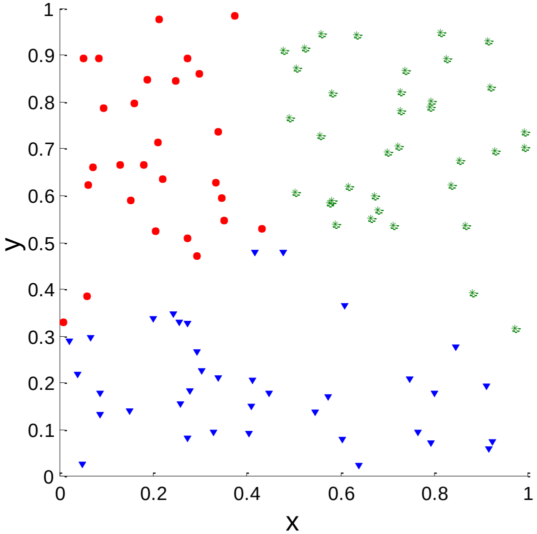
Random Points

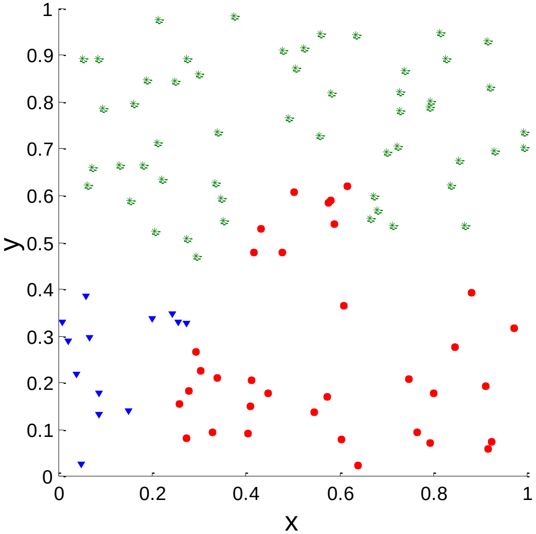




DBSCAN

K-means





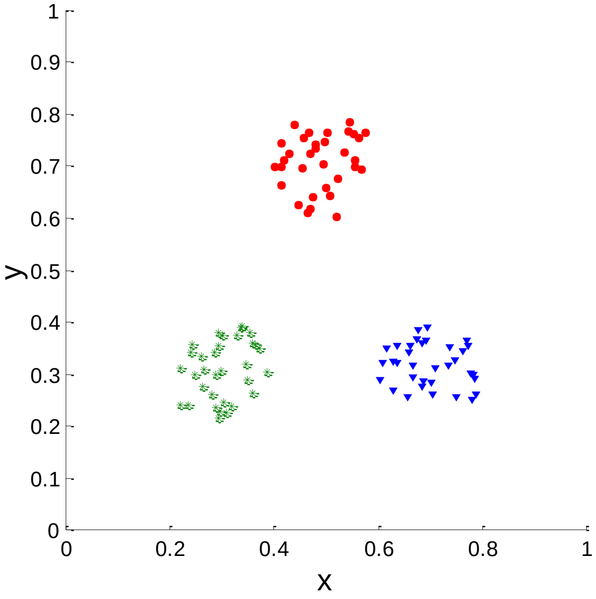
Complete Link

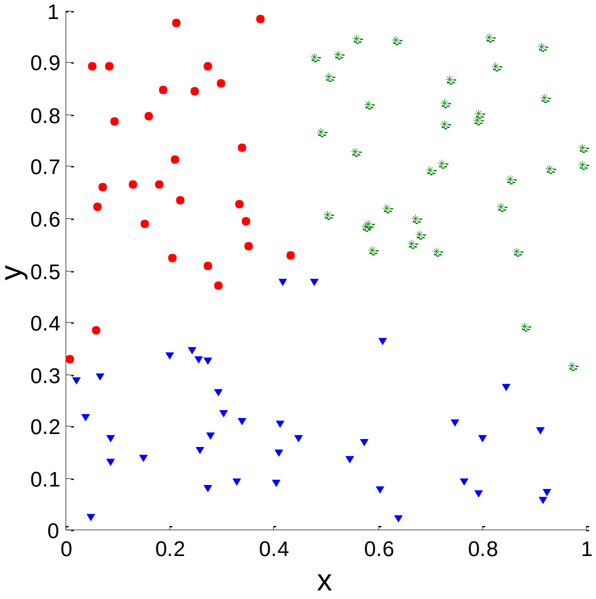
Different Aspects of Cluster Validation

1. 确定数据集的聚类趋势（ clustering tendency），即识别数据中是否实际存在非随机结构
2. 确定正确的簇个数。
3. 不参考附加的信息，评估聚类分析结果对数据拟合情况。
4. 将聚类分析结果与已知的客观结果（如，外部提供的类标号）比较。
5. *比较两个簇集，确定哪个更好。*

相关性：Measuring Cluster Validity Via Correlation

理想和实际相似度矩阵之间相关性比较



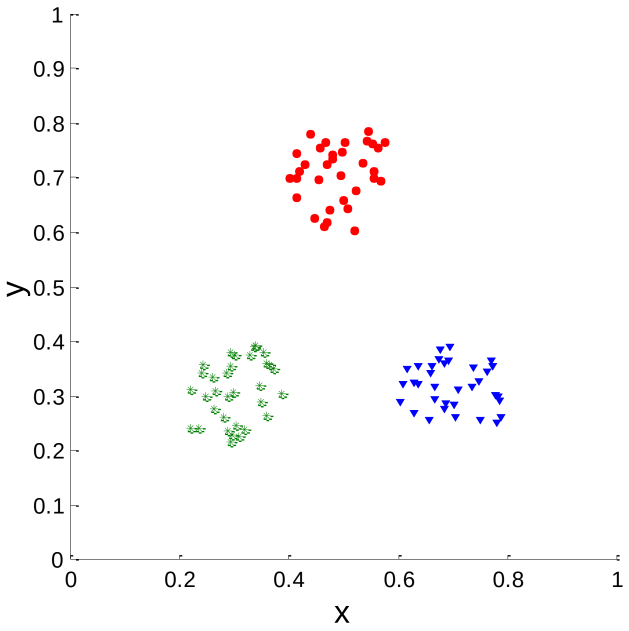


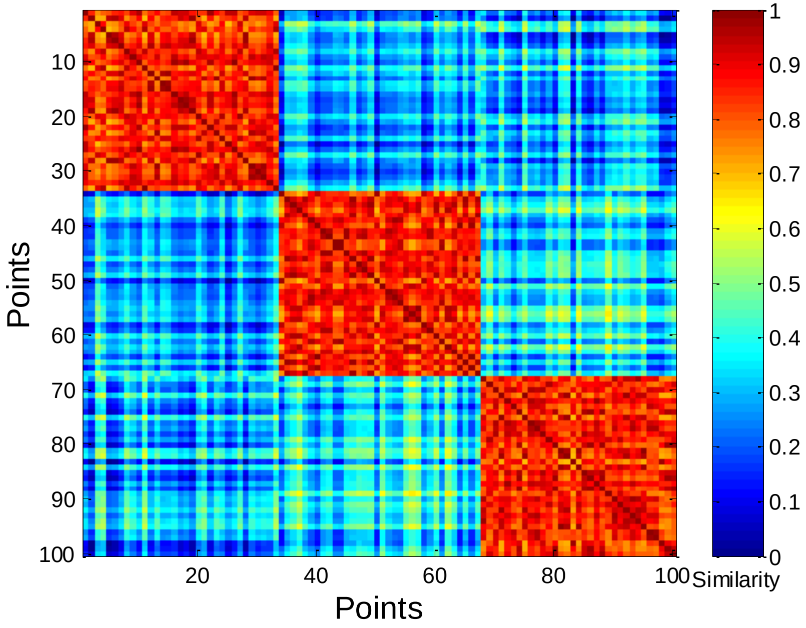
Corr = -0.9235

Corr = -0.5810

相似度矩阵：Using Similarity Matrix for Cluster Validation

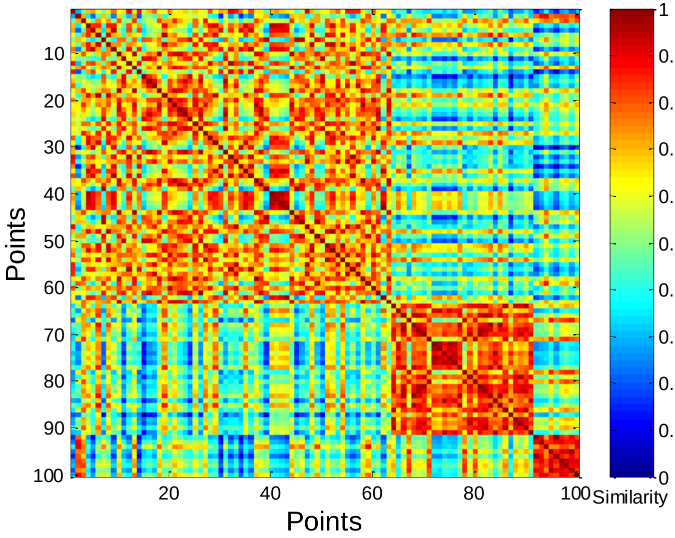
Order the similarity matrix with respect to cluster labels and inspect visually.

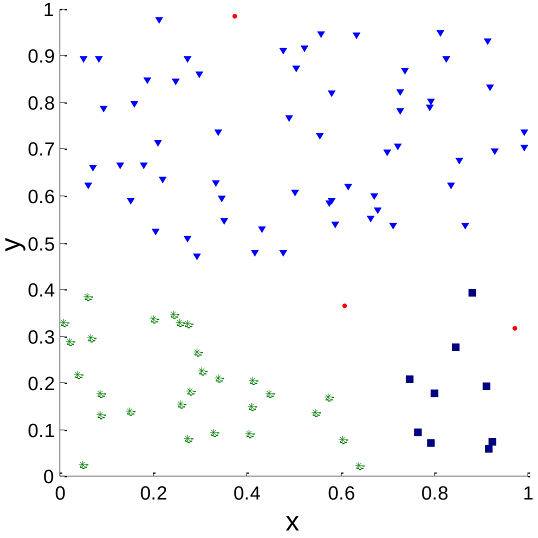




Using Similarity Matrix for Cluster Validation

Clusters in random data are not so crisp

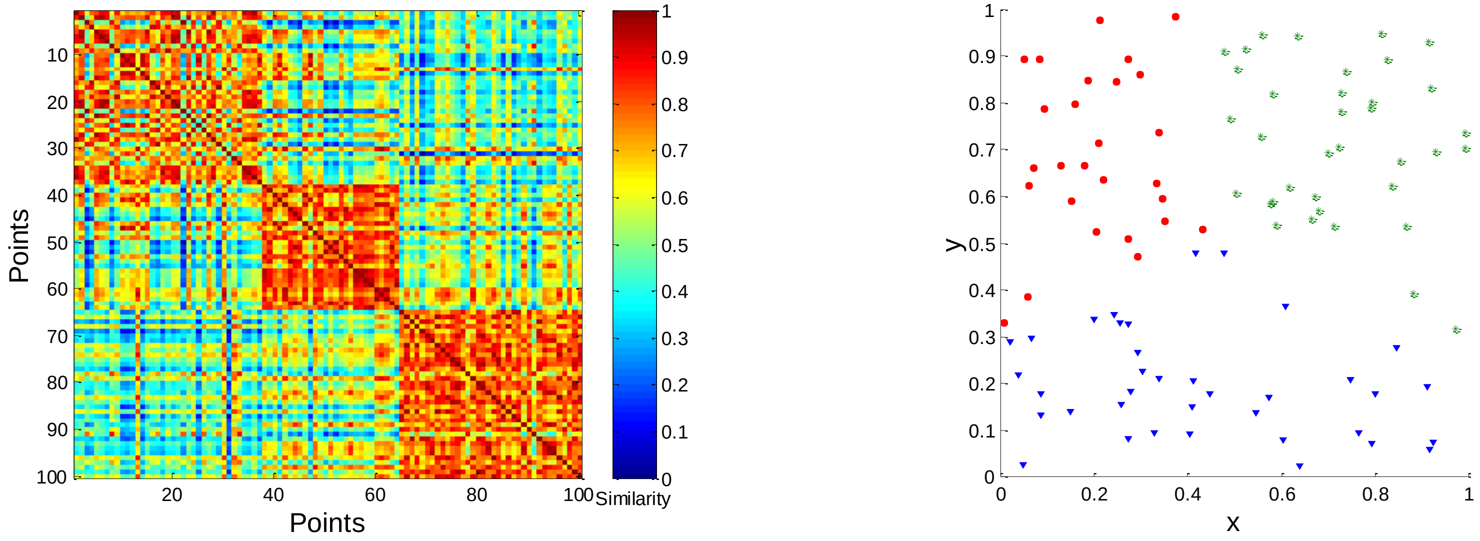




DBSCAN

Using Similarity Matrix for Cluster Validation

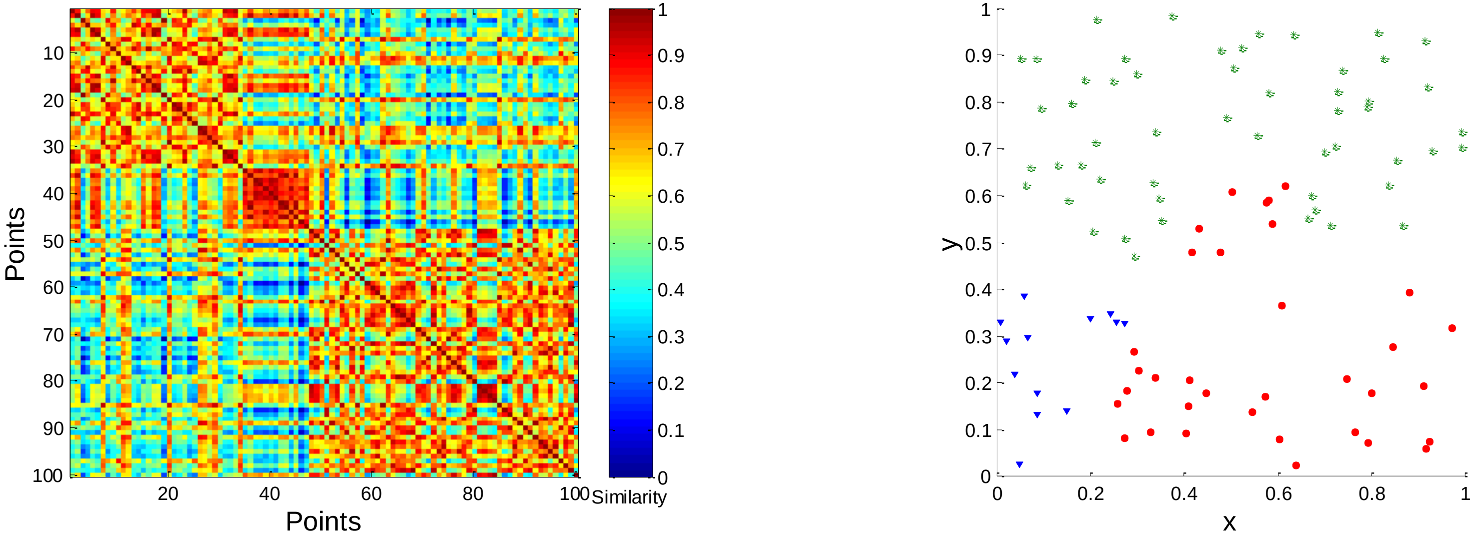
Clusters in random data are not so crisp



K-means

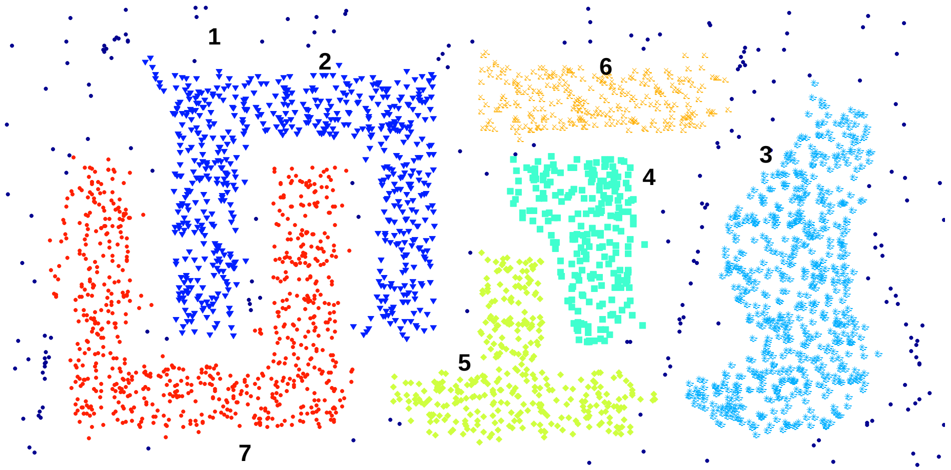
Using Similarity Matrix for Cluster Validation

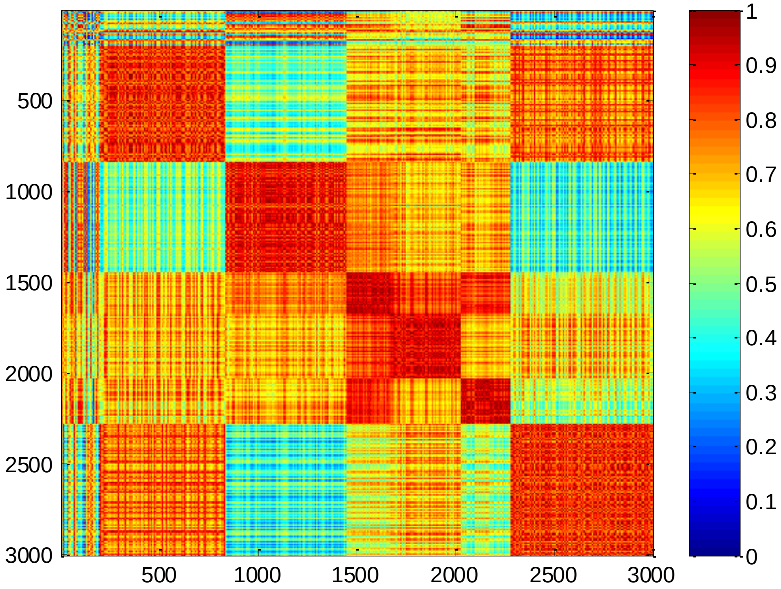
Clusters in random data are not so crisp



Complete Link

Using Similarity Matrix for Cluster Validation





DBSCAN

Internal Measures: SSE（自学）

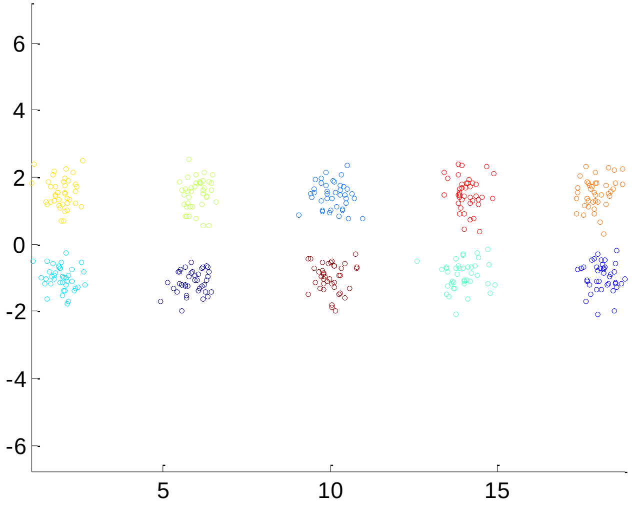
Clusters in more complicated figures aren’t well separated

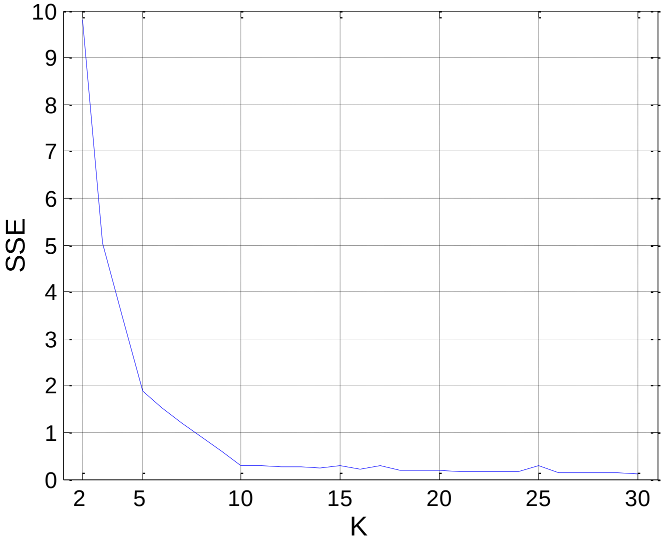
Internal Index: Used to measure the goodness of a clustering structure without respect to external information

- SSE

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

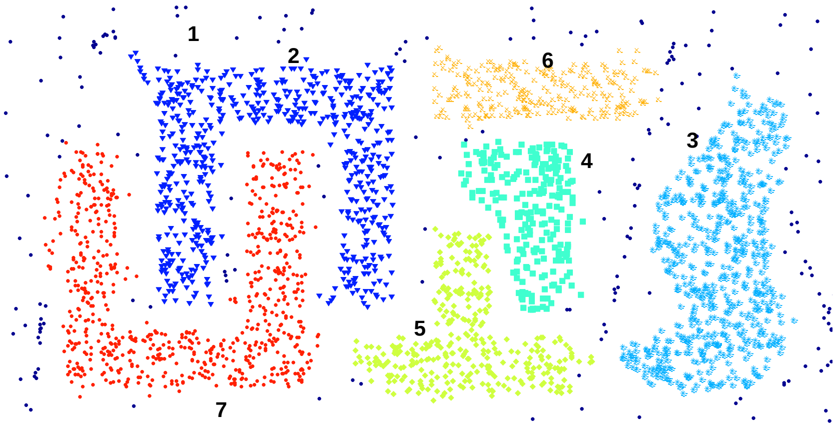
Can also be used to estimate the number of clusters

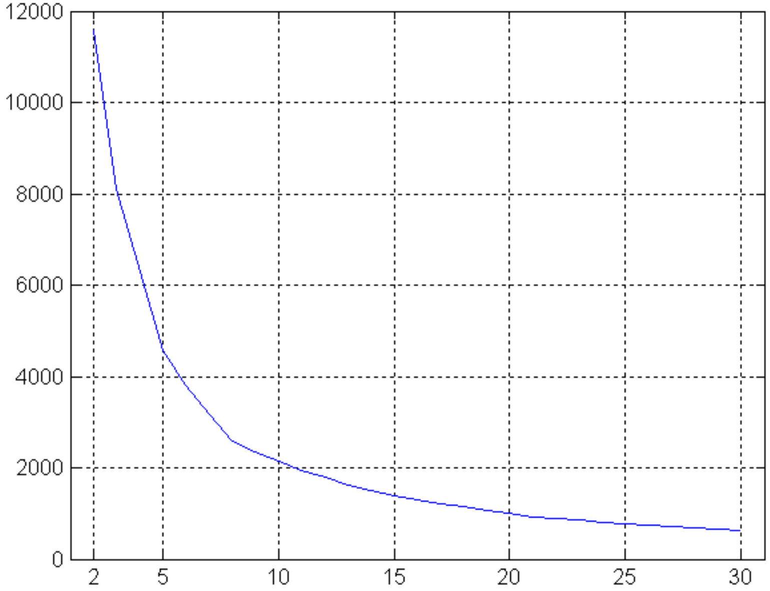




Internal Measures: SSE （自学）

SSE curve for a more complicated data set





SSE of clusters found using K-means

Framework for Cluster Validity （自学）

Need a framework to interpret any measure.

- For example, if our measure of evaluation has the value, 10, is that good, fair, or poor?

Statistics provide a framework for cluster validity

- The more “atypical” a clustering result is, the more likely it represents valid structure in the data
- Can compare the values of an index that result from random data or clusterings to those of a clustering result.
  - If the value of the index is unlikely, then the cluster results are valid
- These approaches are more complicated and harder to understand.

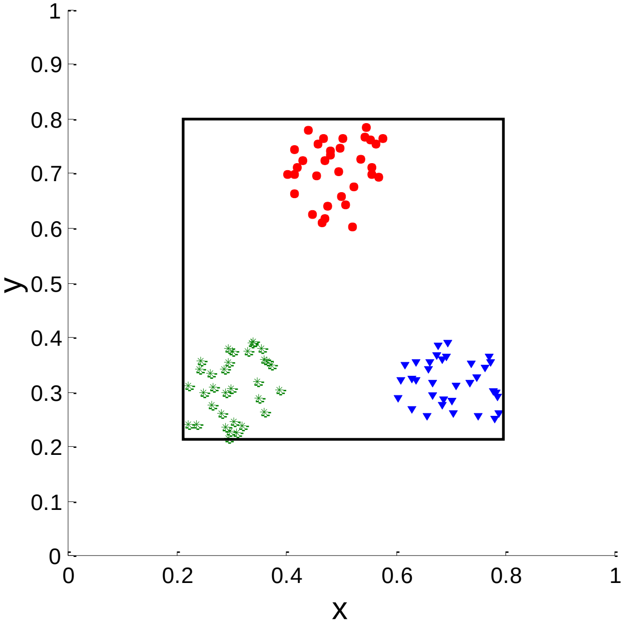
For comparing the results of two different sets of cluster analyses, a framework is less necessary.

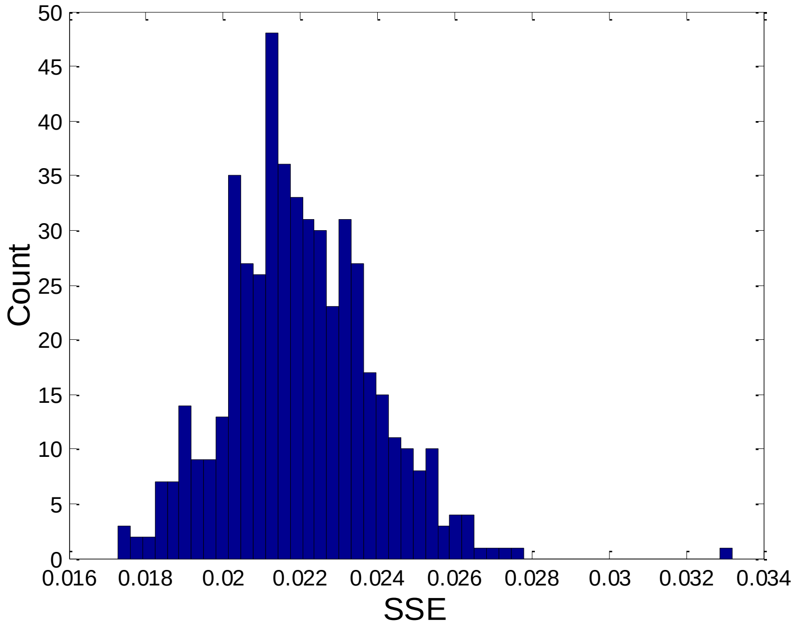
- However, there is the question of whether the difference between two index values is significant

Statistical Framework for SSE （自学）

Example

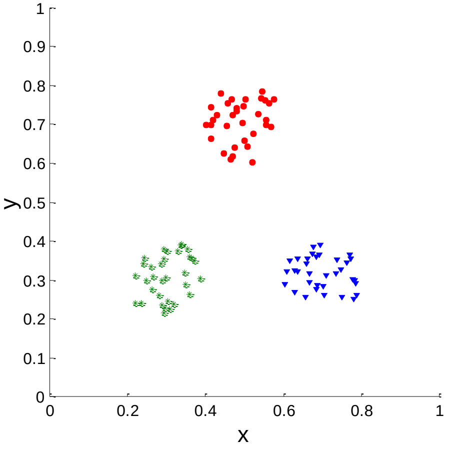
- Compare SSE of 0.005 against three clusters in random data
- Histogram shows SSE of three clusters in 500 sets of random data points of size 100 distributed over the range 0.2 – 0.8 for x and y values

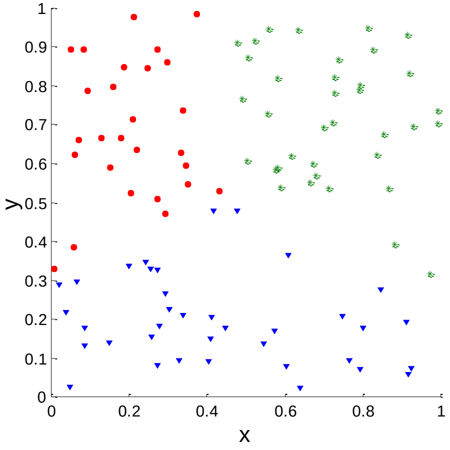


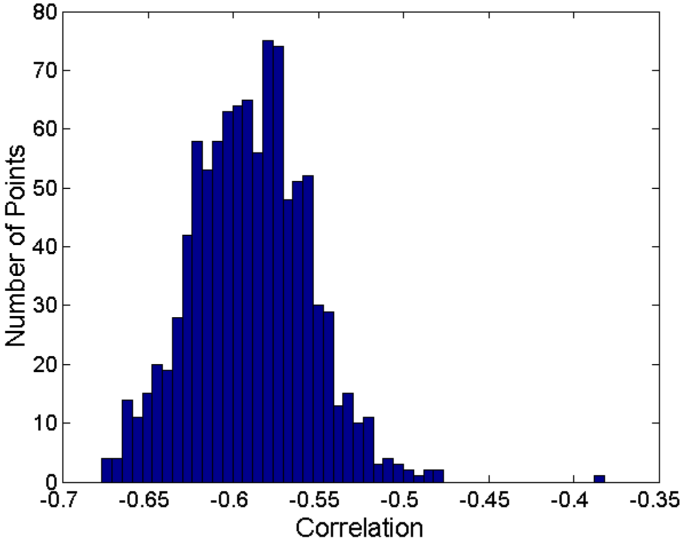


Statistical Framework for Correlation （自学）

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







Corr = -0.9235

Corr = -0.5810

Internal Measures: Cohesion and Separation （自学）

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

- Example: SSE

Cluster Separation: 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)

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

- Separation is measured by the between cluster sum of squares

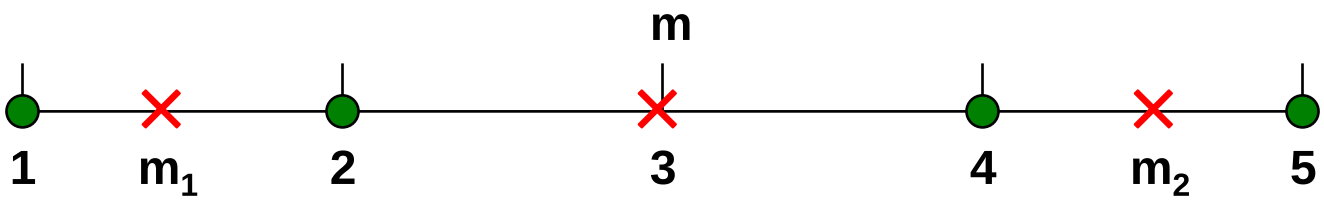
$$BSS = \sum_{i} |C_i| (m - m_i)^2$$

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

Internal Measures: Cohesion and Separation （自学）

Example: SSE

- BSS + WSS = constant



K=1 cluster:

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

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

$$Total = 10 + 0 = 10$$

K=2 clusters:

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

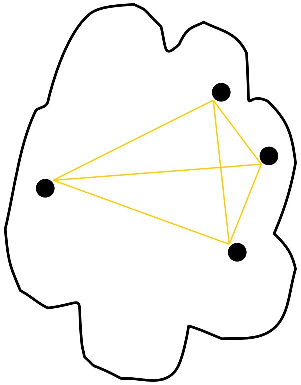
$$BSS = 2 \times (3-1.5)^2 + 2 \times (4.5-3)^2 = 9$$

$$Total = 1 + 9 = 10$$

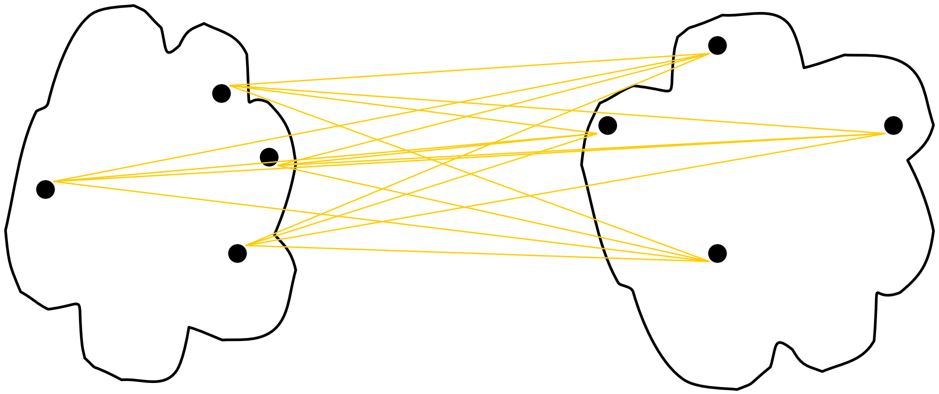
Internal Measures: Cohesion and Separation （自学）

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

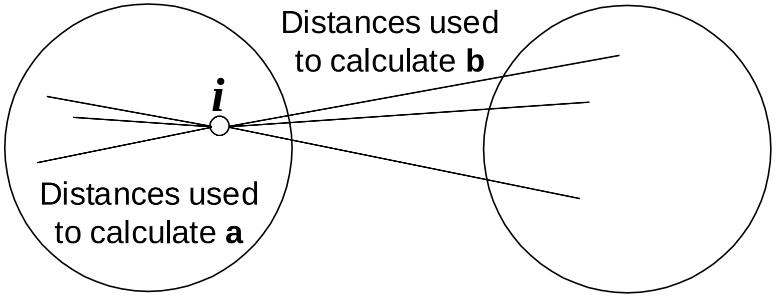
Internal Measures: Silhouette Coefficient （自学）

Silhouette coefficient combines ideas of both cohesion and separation, but for individual points, as well as clusters and clusterings

For an individual point, ***i***

- Calculate ***a*** = average distance of ***i*** to the points in its cluster
- Calculate ***b*** = min (average distance of ***i*** to points in another cluster)
- The silhouette coefficient for a point is then given by

  s = (b – a) / max(a,b)


Distances used to calculate **b**
i
Distances used to calculate **a**

- Typically between 0 and 1.
- The closer to 1 the better.

Can calculate the average silhouette coefficient for a cluster or a clustering

External Measures of Cluster Validity: Entropy and Purity （自学）

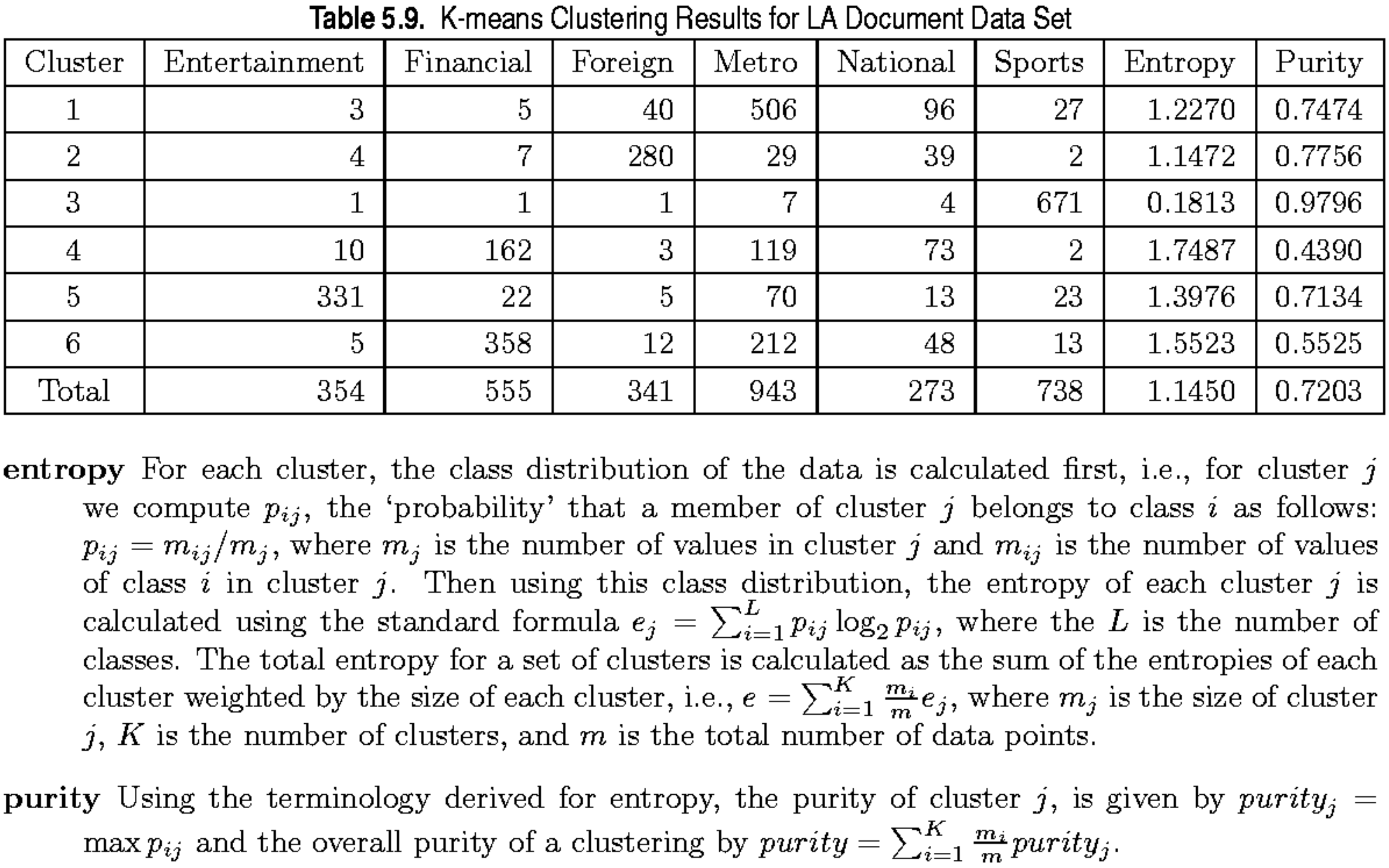


Table 5.9. K-means Clustering Results for LA Document Data Set

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Entropy	Purity
1	3	5	40	506	96	27	1.2270	0.7474
2	4	7	280	29	39	2	1.1472	0.7756
3	1	1	1	7	4	671	0.1813	0.9796
4	10	162	3	119	73	2	1.7487	0.4390
5	331	22	5	70	13	23	1.3976	0.7134
6	5	358	12	212	48	13	1.5523	0.5525
Total	354	555	341	943	273	738	1.1450	0.7203

entropy For each cluster, the class distribution of the data is calculated first, i.e., for cluster j we compute p_{ij}, the 'probability' that a member of cluster j belongs to class i as follows: $p_{ij} = m_{ij}/m_j$, where m_j is the number of values in cluster j and m_{ij} is the number of values of class i in cluster j. Then using this class distribution, the entropy of each cluster j is calculated using the standard formula $e_j = \sum_{i=1}^{L} p_{ij} \log_2 p_{ij}$, where the L is the number of classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e., $e = \sum_{i=1}^{K} \frac{m_i}{m} e_j$, where m_j is the size of cluster j, K is the number of clusters, and m is the total number of data points.

purity Using the terminology derived for entropy, the purity of cluster j, is given by $purity_j = \max p_{ij}$ and the overall purity of a clustering by $purity = \sum_{i=1}^{K} \frac{m_i}{m} purity_j$.

Final Comment on Cluster Validity （自学）

“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.”

Algorithms for Clustering Data, Jain and Dubes

谢谢！

数据挖掘

教师：王东京

学院：计算机学院

邮箱：dongjing.wang@hdu.edu.cn