聚类

陈封能等著，《数据挖掘导论》,第2版，机械工业出版社，2019

周志华著，《机器学习》，清华大学出版社，2016，P197

李航著，《统计学习方法》,第2版，清华大学出版社，2019

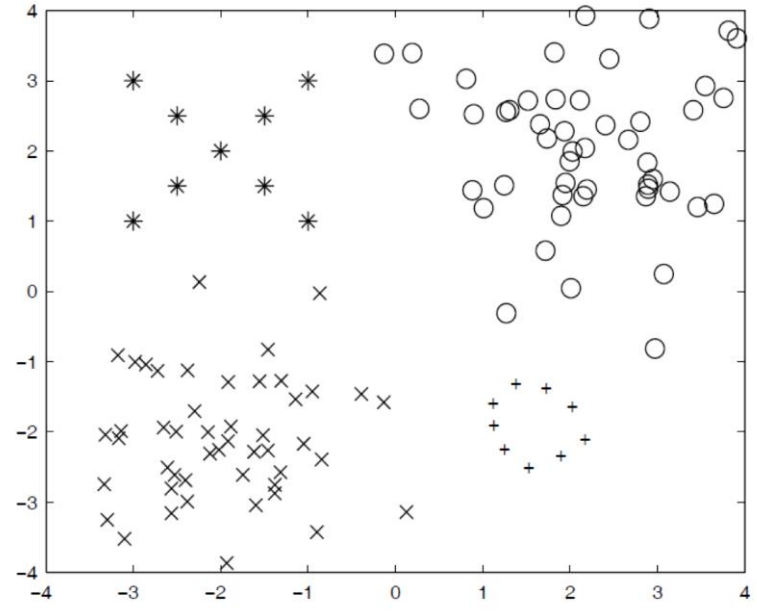
无监督学习简介

- 现实生活中常常会有这样的问题:
  - 人们很容易就获得大量未标记的样本
  - 缺乏足够的先验知识，因此难以人工标注类别，或类别标注的成本太高
- 很自然地，我们希望计算机能代我们（部分）完成这些工作，或至少提供一些帮助将一组数据依照内在相似性划分为多个类别，使类别内的数据相似度较大而类别间的数据相似度较小。
- **无监督学习**——根据没有标记的样本，学习数据中的信息
- 聚类分析假设数据的特征允许我们可以识别不同的类别，但事先并不知道数据由几个类构成，因而是一种无监督的学习。

- 例如商店希望刻画顾客群的特征，区分不同的客户类，挖掘有价值的客户，以制定不同的关系管理方式，提高客户对商业活动的响应率

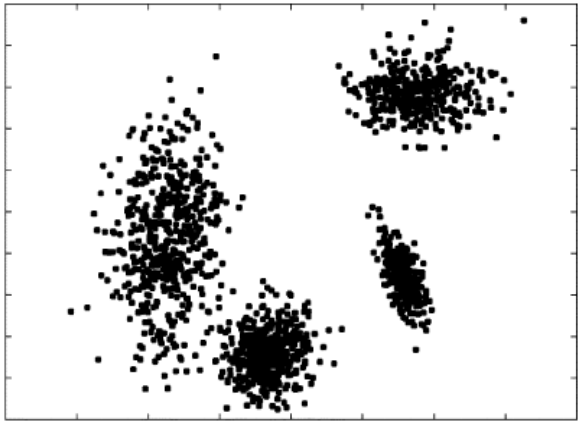
什么是类?

- 至今还没有普遍接受的定义：哪些特征决定了一个类。因此，不同的聚类方法可能得到不同的聚类结果。
- 直观上：一个类是一组个体（对象、点等），这些个体离这个类的中心个体比较“近”（在合适的度量下）；不同类的成员之间的距离“比较远”。
- 必须注意：“类”可能仅仅是一个聚类方法的结果
- 一个“类”依赖于如何定义它以及应用背景



聚类与分类 (clustering and classification)

- 分类:
  - 有类别标记信息, 因此是一种监督学习
  - 根据训练样本获得分类器，然后把每个数据归结到某个已知的类，进而也可以预测未来数据的归类。
- – 分类具有广泛的应用，例如医疗诊断、信用卡的信用分级、图像模式识别。
- 聚类:
  - 无类别标记, 因此是一种无监督学习
  - 无类别标记样本，根据信息相似度原则进行聚类，通过聚类，人们能够识别密集的和稀疏的区域，因而发现全局的分布模式，以及数据属性之间的关系

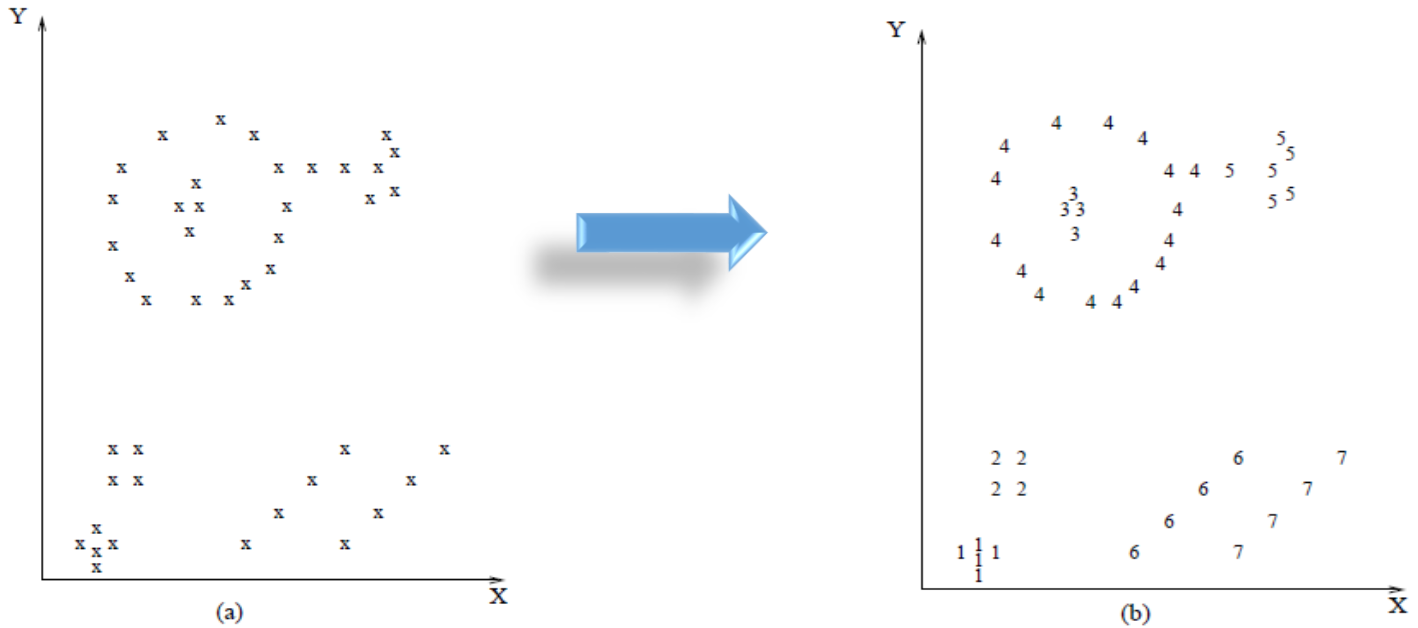


聚类 (clustering)

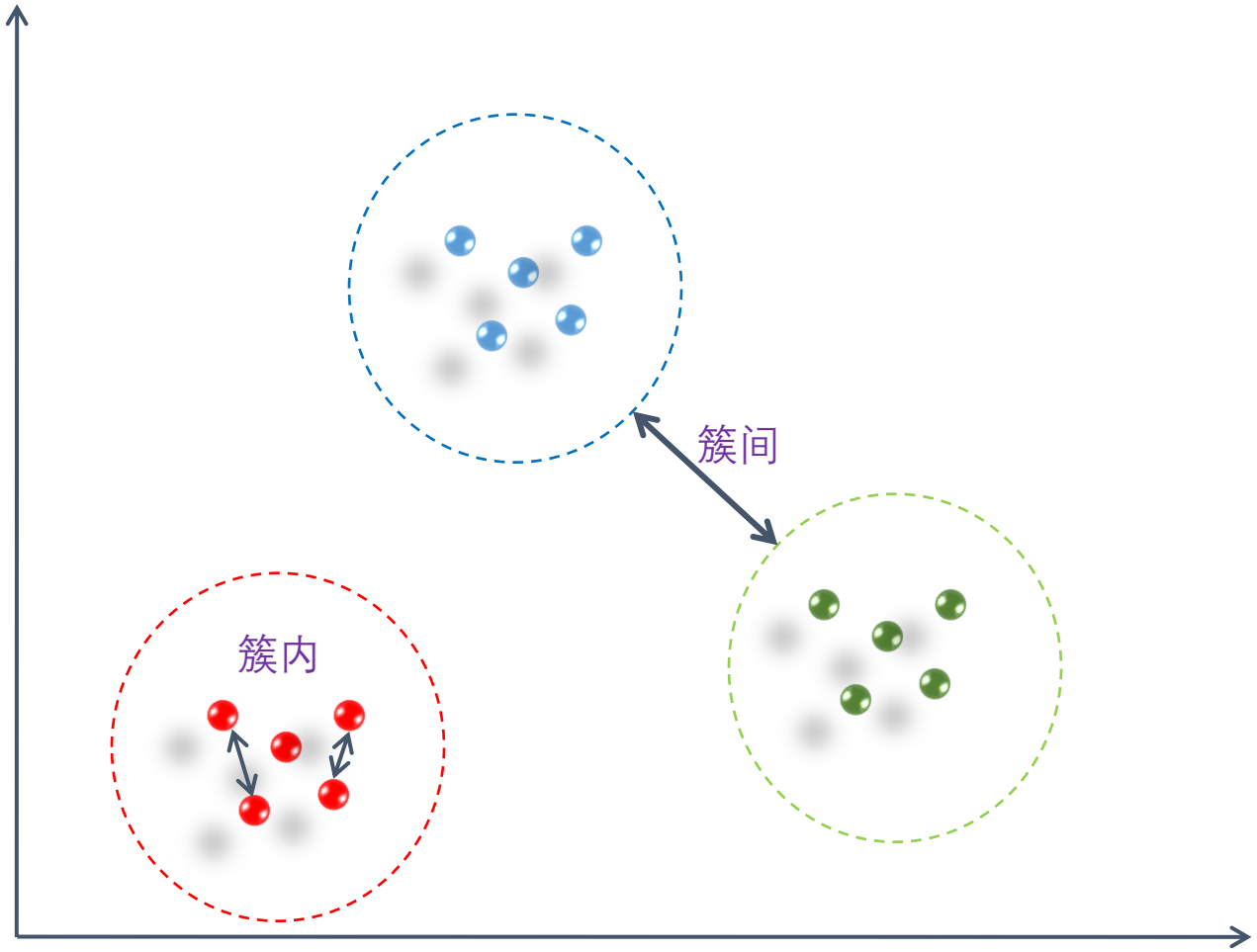
- 聚类是一个难以被严格定义的问题，因为“自然分组”本身就很抽象，且可能因人而异
- 所以，必须首先由人来对问题进行定义。具体来说，需要回答两个问题：
  - **怎样度量样本之间的相似性**？（相似度度量是什么）
  - **怎样衡量某一种分组的好坏**？（目标函数是什么）
- 即使有了明确的定义，要找到“最优分组”也是NP-hard的
  - 例如将100个样本聚集为5类需要考虑超过1067种可能的划分（5100/5！）

什么是聚类分析?

- 聚类分析: 把数据对象划分成子集的过程。 每个子集为一个簇（cluster）
- 求对象组（簇）
  - 同一簇的样本尽可能彼此相似
  - 不同簇的样本尽可能不同.
- 无监督学习（Unsupervised Learning）
  - 没有标记信息（No labels）
  - 数据驱动（Data driven）

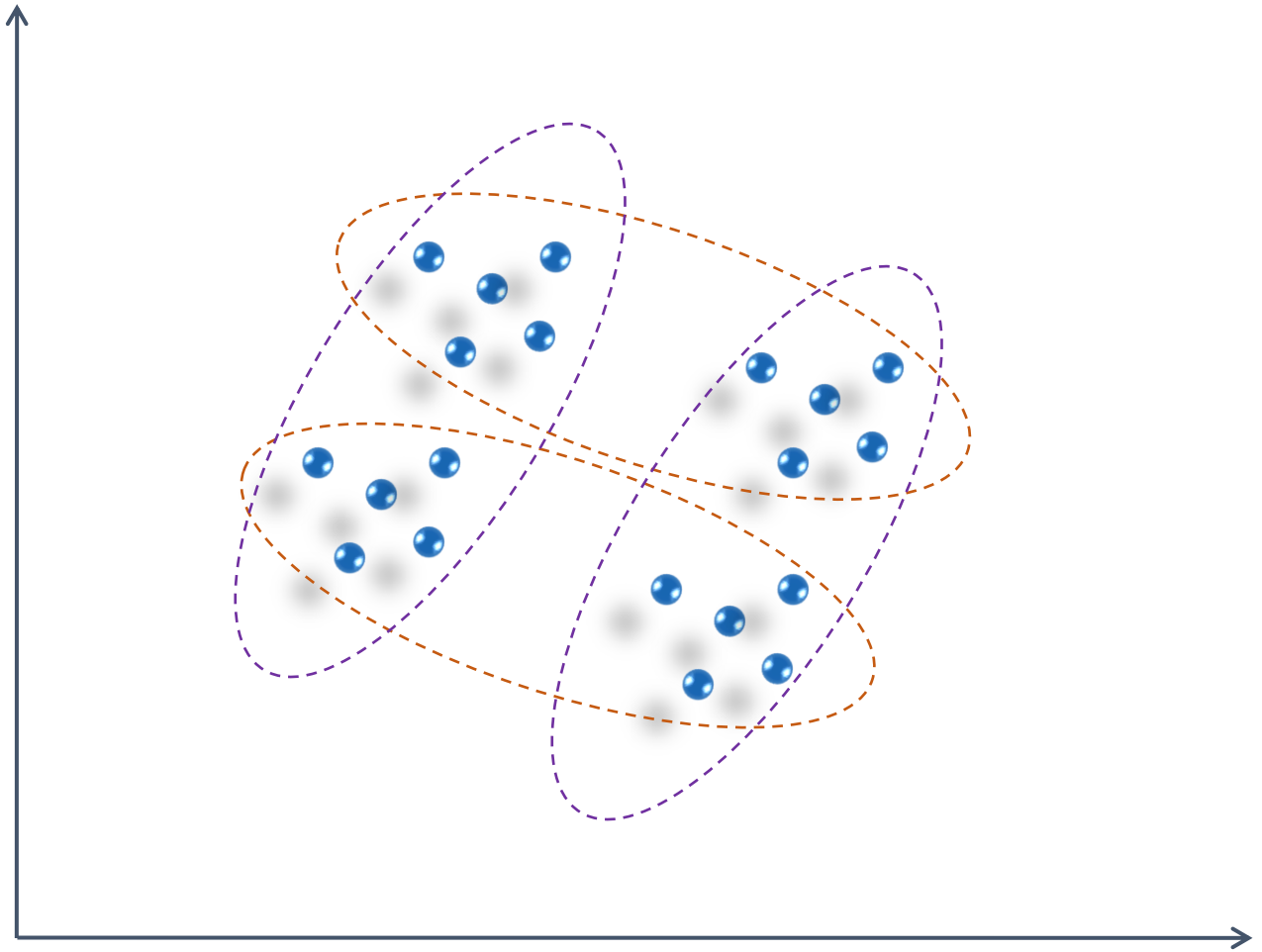


簇（Clusters）：簇间和簇内



我们期望聚类后不同簇间的样本距离要越远越好，同一簇内的样本距离越近越好

簇（Clusters）



聚类为无监督的，没有严格意义的聚类对与错，希望聚类后同一类样本尽量接近

相异性 (距离) 与相似性度量

- 聚类就是发现数据中具有“相似性”(similarity) 的个体
- 选择合适的“相似性”度量是进行聚类的关键, 相似性度量函数 s(·, ·) 一般满足

$$
\begin{aligned}
&1.\ 0 \le s(\mathbf{x}, \mathbf{y}) \le 1 \\
&2.\ s(\mathbf{x}, \mathbf{x}) = 1 \\
&3.\ s(\mathbf{x}, \mathbf{y}) = s(\mathbf{y}, \mathbf{x})
\end{aligned}
$$

- 也可以使用相异性(dissimilarity) 来度量数据之间的接近程度。下面我们以相异性为例. 相异性度量和相似性度量之间一般可以相互转换。
- 相异性度量多为某种“距离” 度量
- 样本点之间的相异性(距离) 函数 $d(\cdot, \cdot)$ 一般满足

$$
\begin{aligned}
&1.\ d(\mathbf{x}, \mathbf{y}) \ge 0, \text{等号成立当且仅当 } \mathbf{x} = \mathbf{y} \\
&2.\ d(\mathbf{x}, \mathbf{x}) = 0 \\
&3.\ d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x}) \\
&4.\ d(\mathbf{x}, \mathbf{y}) \le d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y})
\end{aligned}
$$

- 相似度的度量并无统一的标准，实际中常用欧式距离，也可根据实际问题自己定义

聚类方法

- 划分方法 (Partitioning Methods)
  - **K-均值**（K-Means）
  - 顺序领导者方法（Sequential Leader Methods）
  - **基于密度的方法**（ Density Based Methods)
  - 基于模型的方法（Model Based Methods）(GMM)

- **层次方法**（Hierarchical Methods）

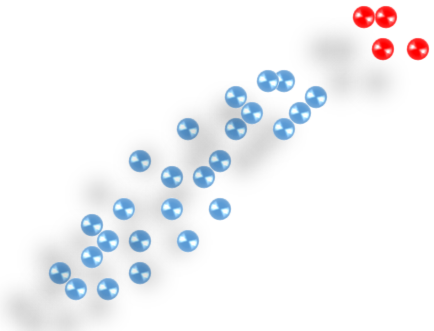
K-means 聚类质量评价的目标函数

- **误差平方和** (Sum of the squared error, SSE)

$$\mathrm{SSE} = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - c_i\|_2^2, \quad c_i = \frac{1}{n_i} \sum_{x \in C_i} x$$

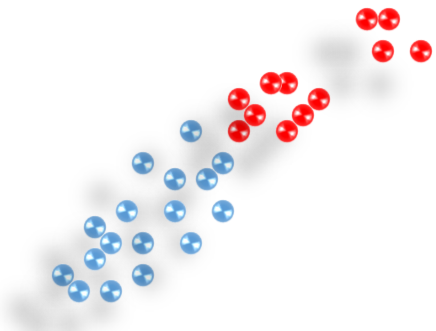
其中 K 是总的簇的个数。 C_i 表示第 i 个簇，c_i 表示第 i 个簇的质心(均值)，x 表示第 i 个簇的任一样本，第 i 个簇 C_i 的样本数为 n_i 。

- 误差平方和准则通过计算每个样本和它到其类均值的距离平方，最后再求和得到最后的数值
- 样本集给定情况下，SSE的值取决于 K 个聚合中心
- SSE 刻画了簇内样本围绕簇均值的紧密程度，其值越小，则簇内样本相似度越高
- 适用于各类样本比较密集 (球状分布) 且样本数目悬殊不大的样本分布



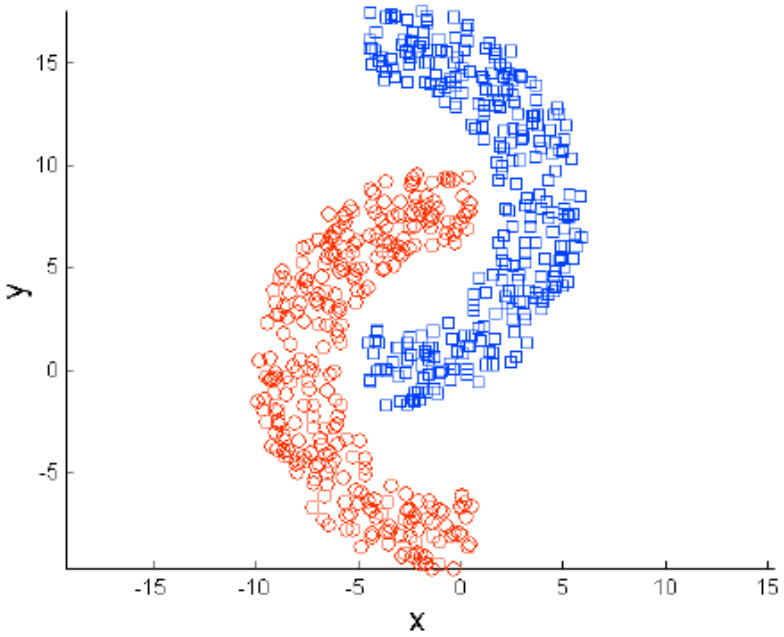
聚成两类合乎人的直观

VS.

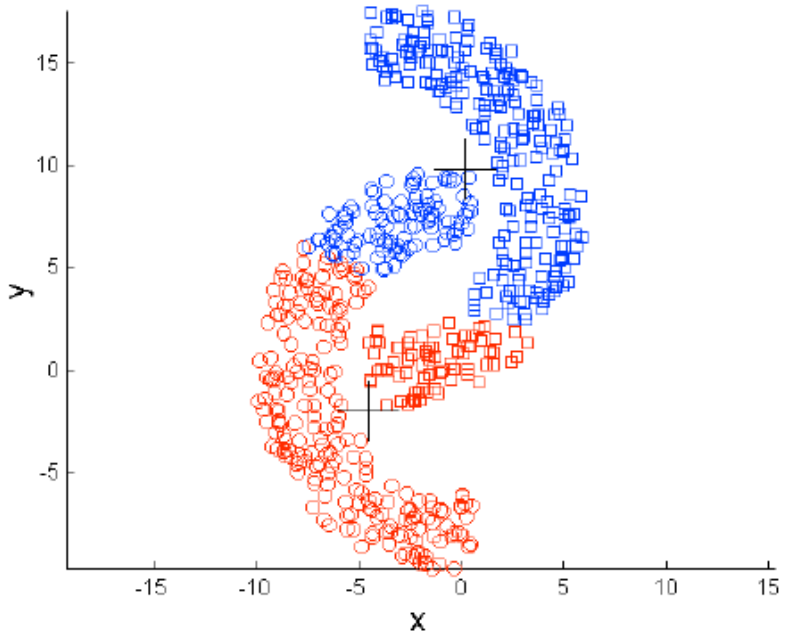


聚成两类误差平方和更小

K-means 聚类质量评价的目标函数



聚成两类合乎人的直观



聚成两类误差平方和更小

K-means 算法的流程

1. 输入训练数据和聚类数目K;
2. 执行下面二者之一:
   - 随机将数据分为K个类C_1，……，C_K，计算每个类的中心 $c_i,\ i = 1, \dots, K$;
   - 指定K个类的中心 c_i,，$i = 1, \dots, K$，将所有数据点划分到离其最近的类中心所在的类
3. 计算每个数据点到其所属类的中心的平方距离

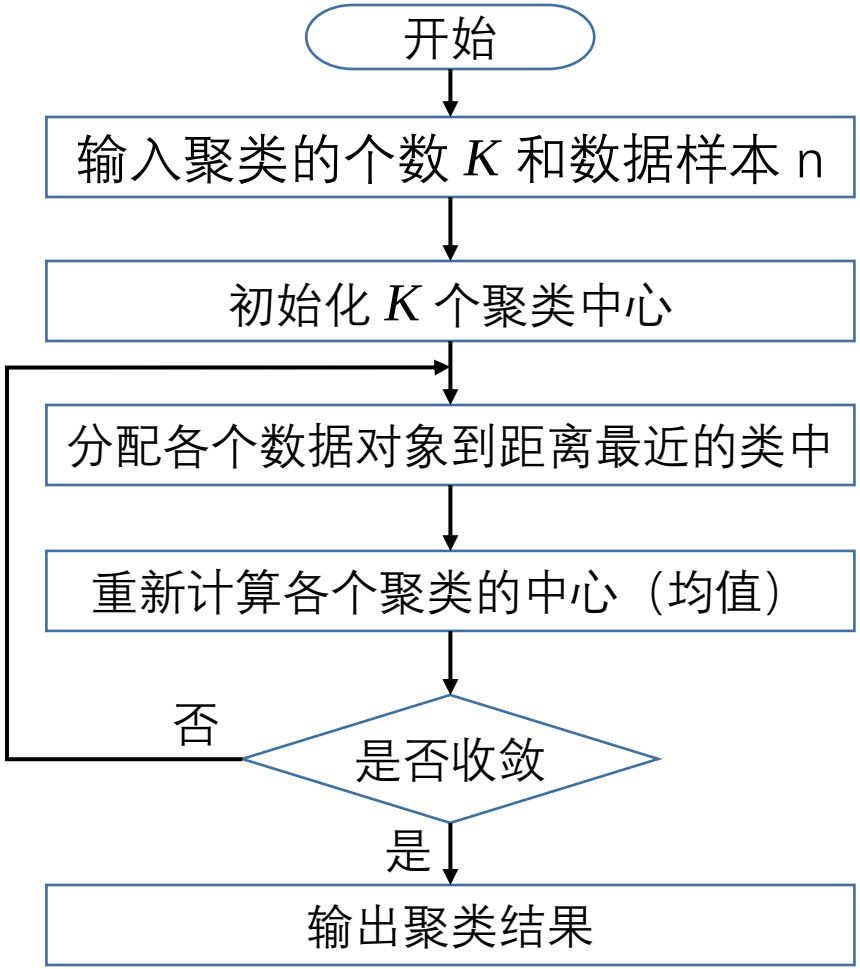
$$\mathrm{SSE} = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - c_i\|_2^2, \quad c_i = \frac{1}{n_i} \sum_{x \in C_i} x$$

4. 重新将每个数据点划分到离其最近的类中心所在的类, 使得 SSE 减少. 完成后重新计算各类的中心 $c_i,\ i = 1, \dots, K$;
5. 重复 3 和 4，直到没有样本点需要调整（即SSE不能再减少）;

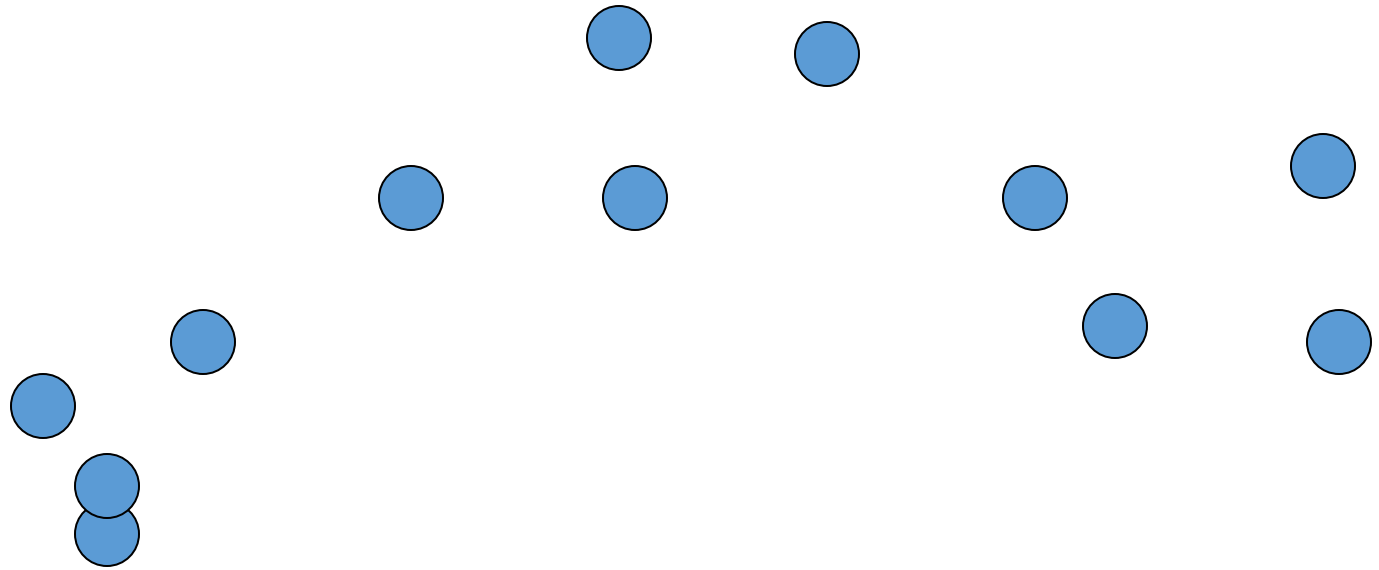
- Reference:

J. MacQueen (1967): "Some Methods for Classification and Analysis of Multivariate Observations", *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, vol.1, pp. 281-297, 1967.

K-means算法的工作流程

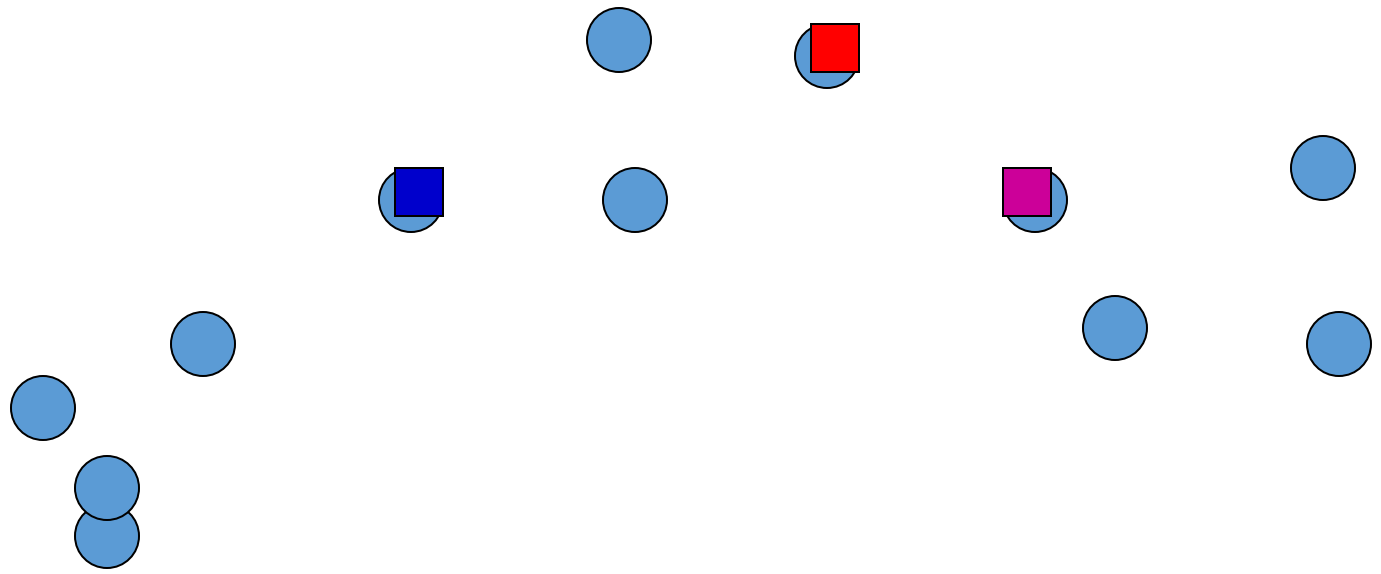


K-means: 一个例子

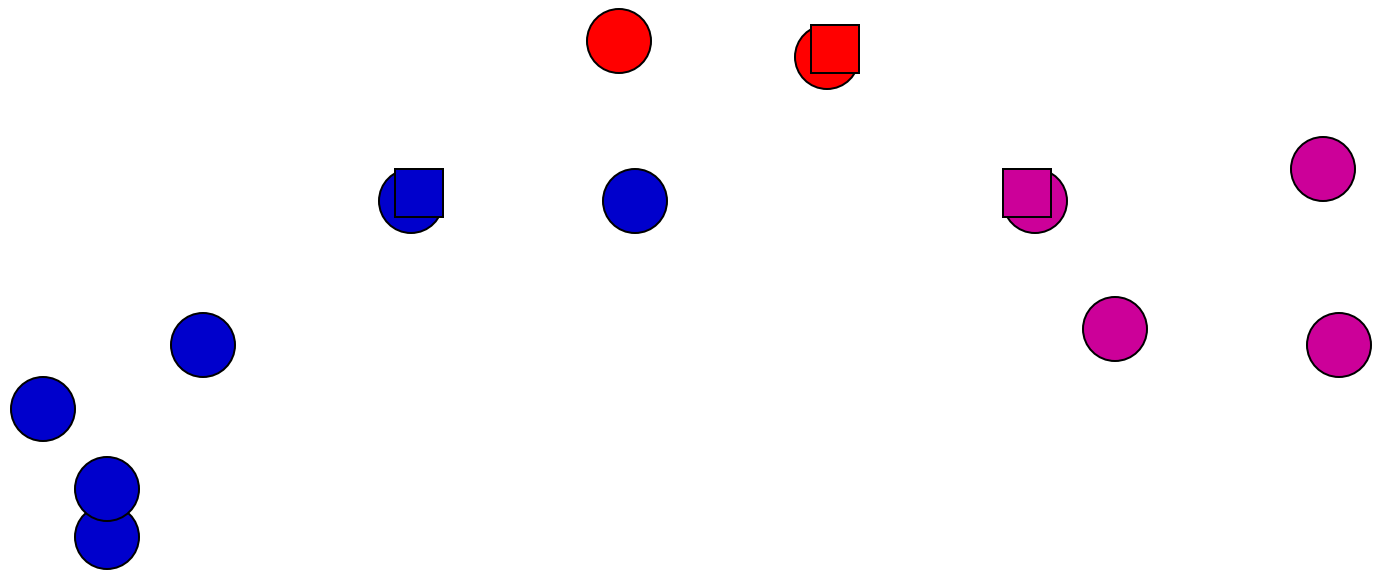


K-means: 随机初始化中心点

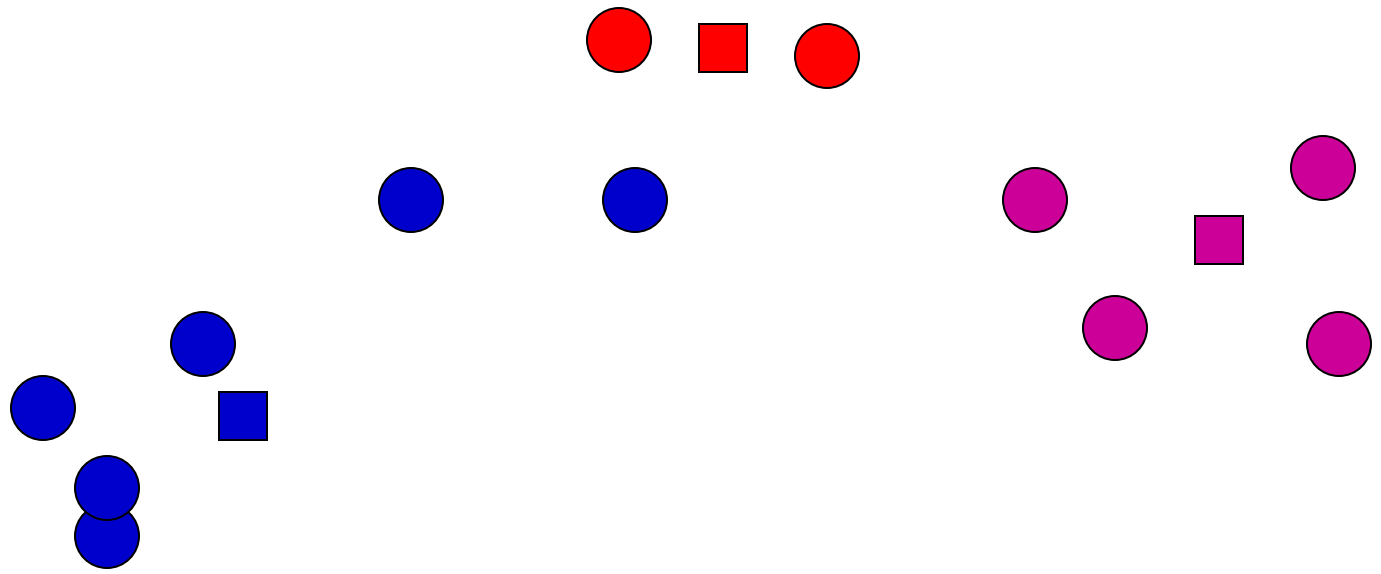
假设聚成3类 (聚类前指定K=3)，　用方形表示聚类中心



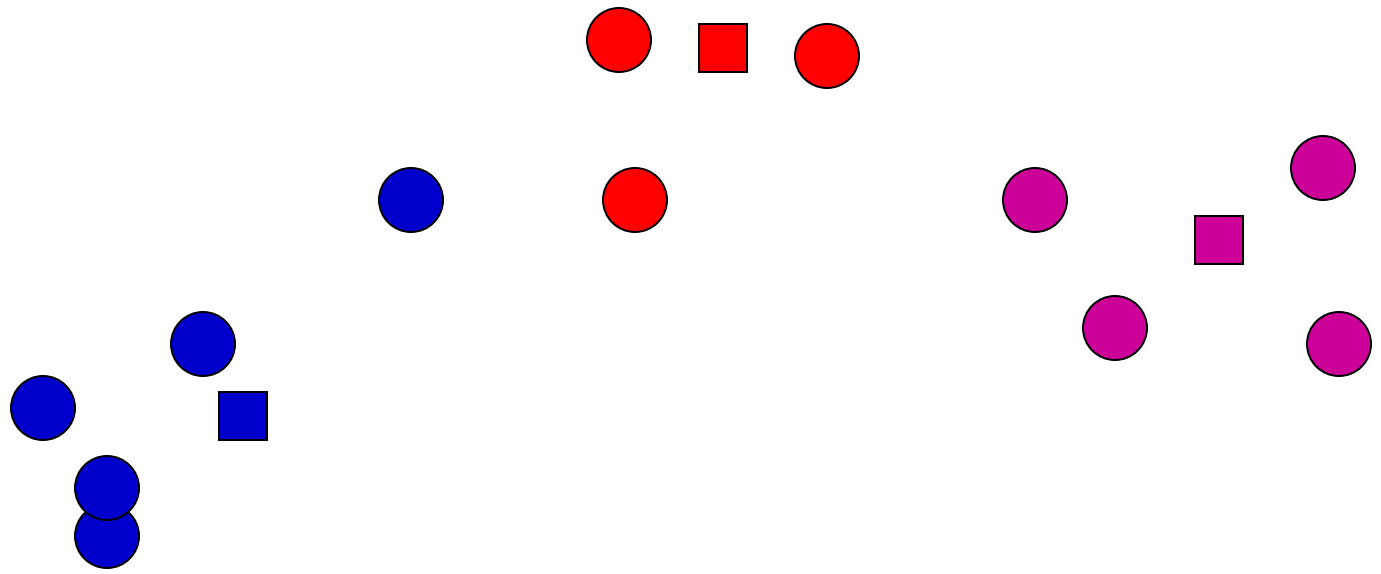
K-means: 将点分配到最近的中心



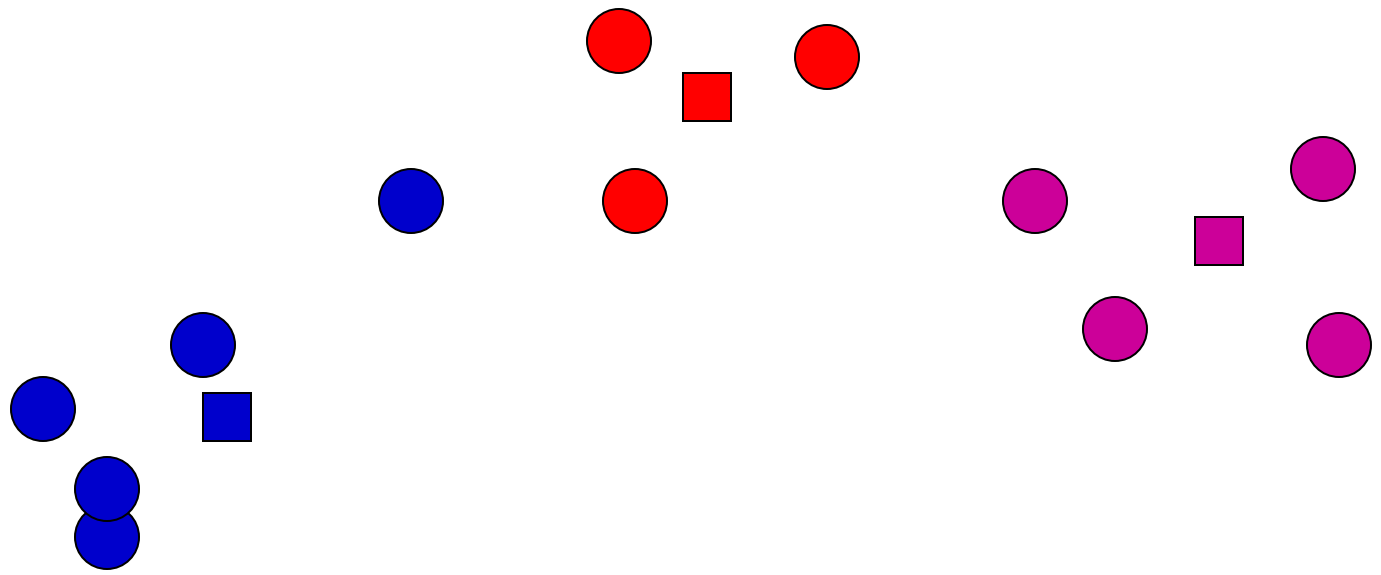
K-means: 重新调整中心



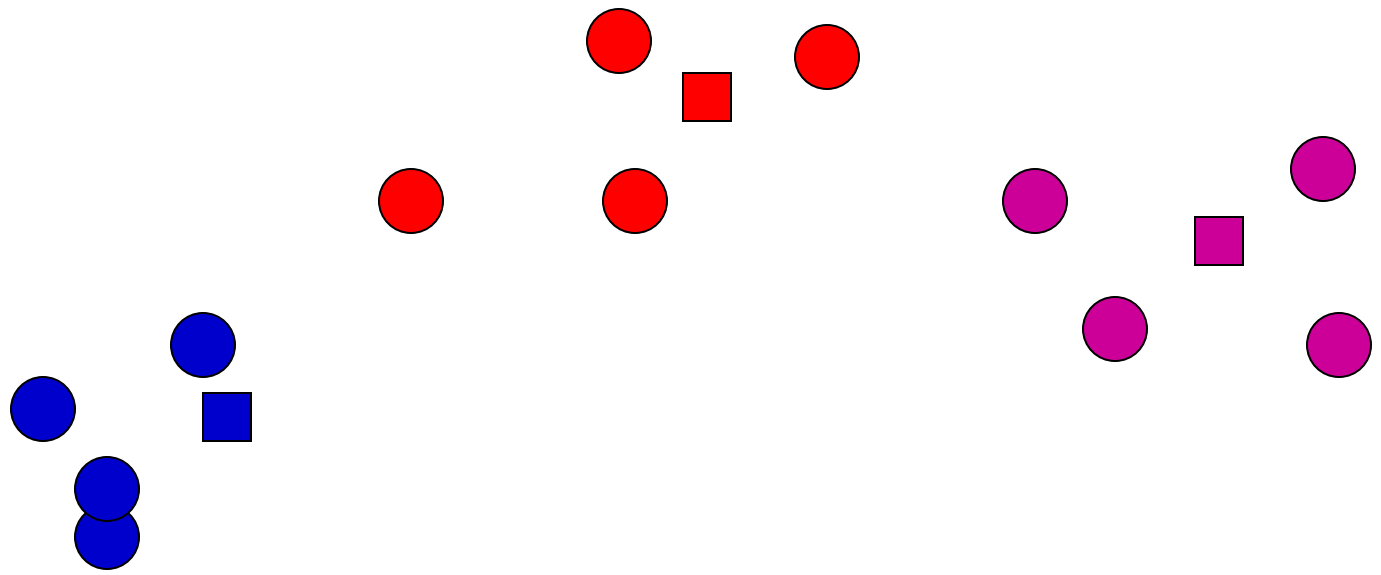
K-means: 将点分配到最近的中心



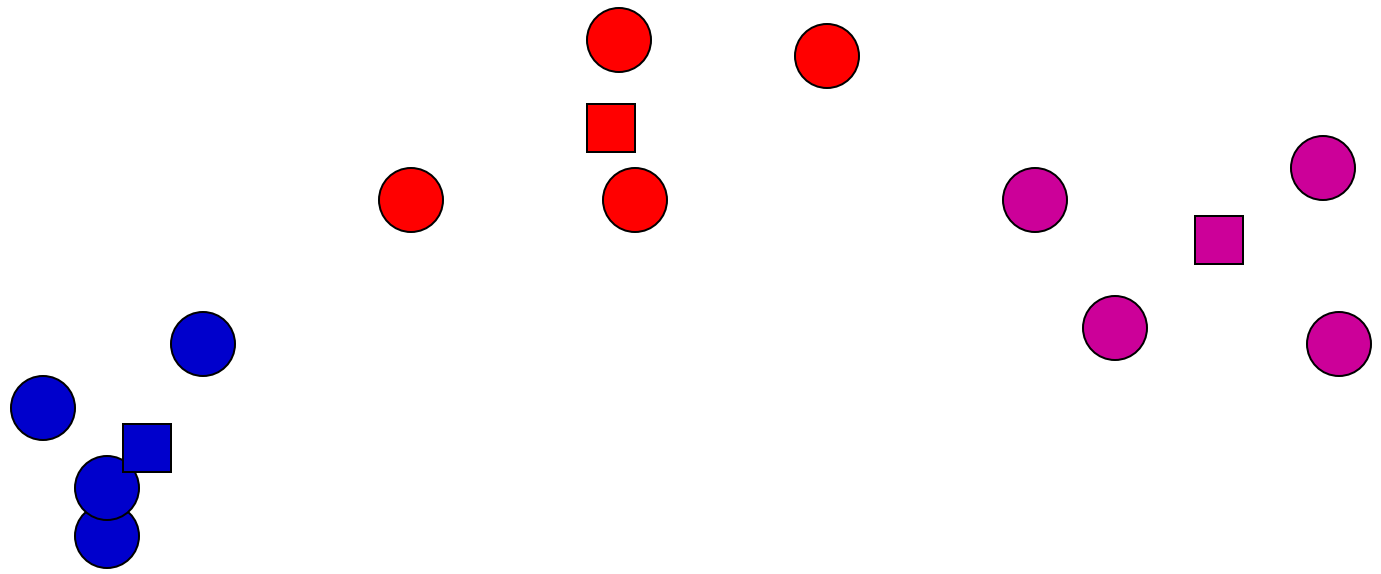
K-means: 重新调整中心



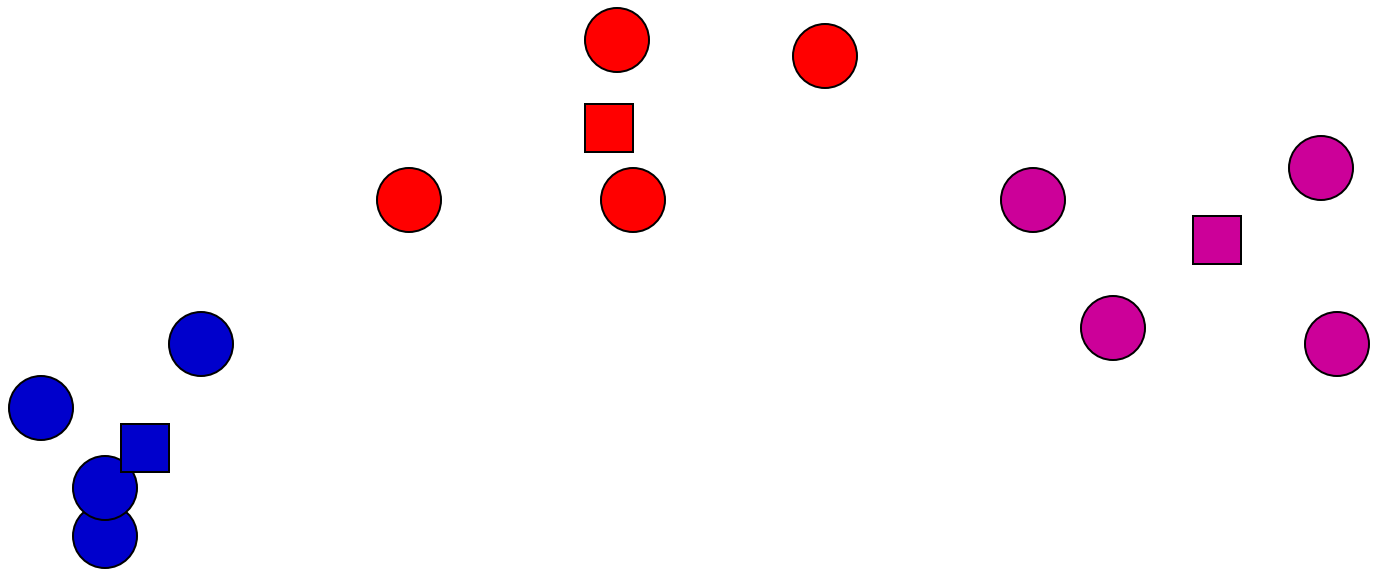
K-means: 将点分配到最近的中心



K-means: 重新调整中心



K-means: 将点分配到最近的中心



簇中心不再变化：聚类完成

关于 K-means 的总结

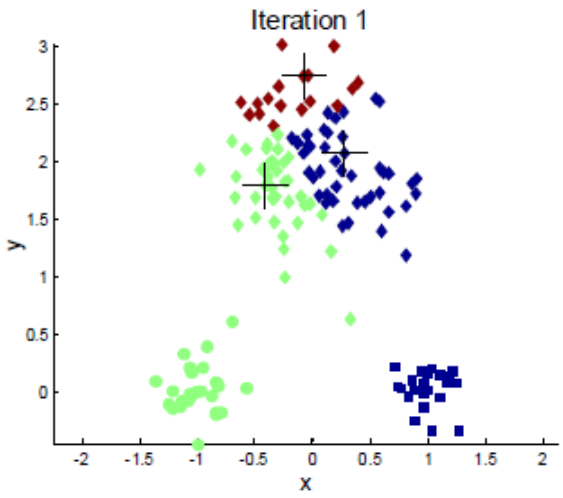
优点

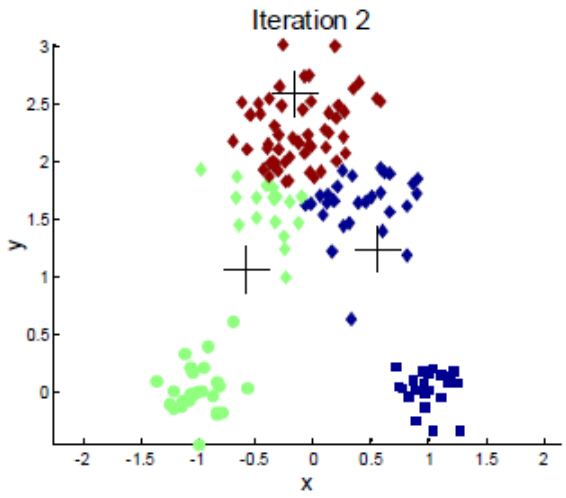
- 简单，适用于常规不相交的簇。
- 收敛相对较快。
- 相对有效和可扩展。时间复杂度 O(I×K×n×m)
  - I：收敛所需迭代次数; K：中心数; n：数据点数, m：类别数
- 假设数据是呈球形分布。实际任务中很少有这种情况。

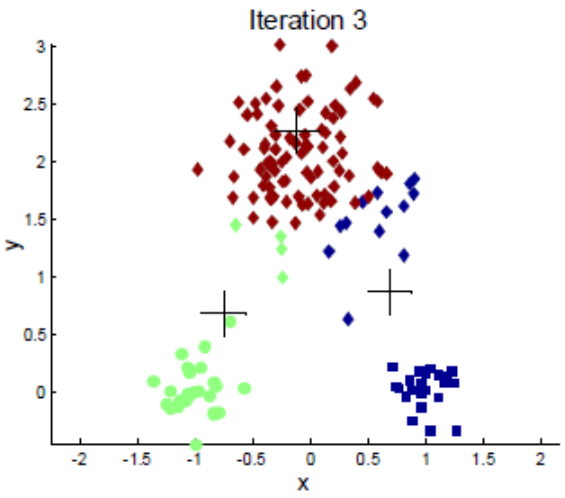
缺点

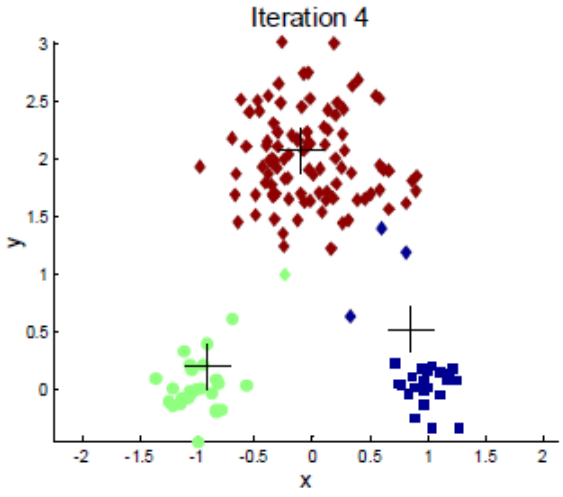
- 需要提前指定 K 的值。
  - 很难确定，领域知识可能会有所帮助。
- 可能会收敛到局部最优点。
  - 在实践中，需要尝试不同的初始中心点。
- 可能对噪声数据和异常值敏感。
  - 因为簇的中心是取平均，因此聚类簇很远的地方的噪声会导致簇的中心点偏移
- 不适合非凸不规则形状的簇, 普遍对球形分布样本聚类较好

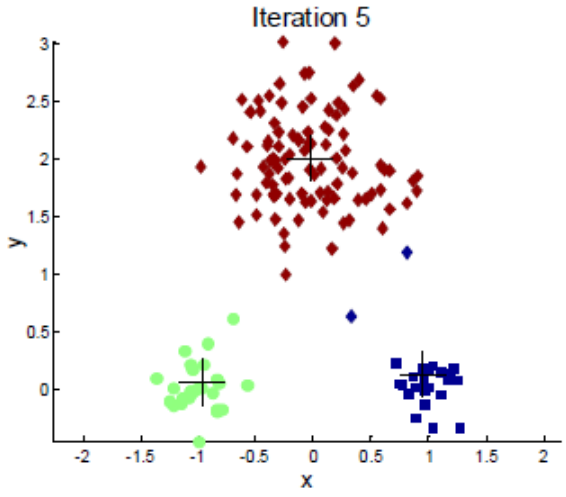
初始中心点对k-means聚类的影响（初始中心点选择较好的例子）

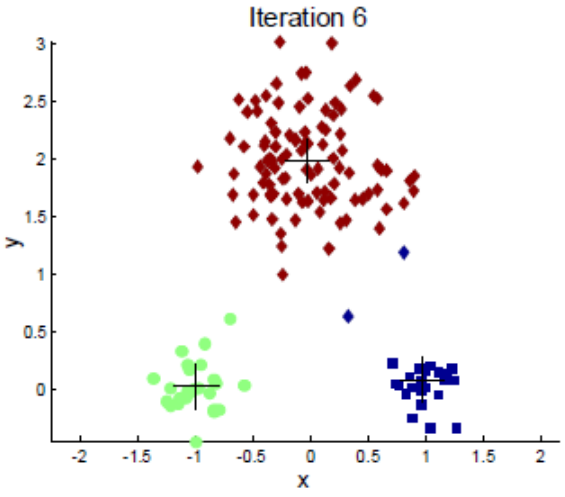




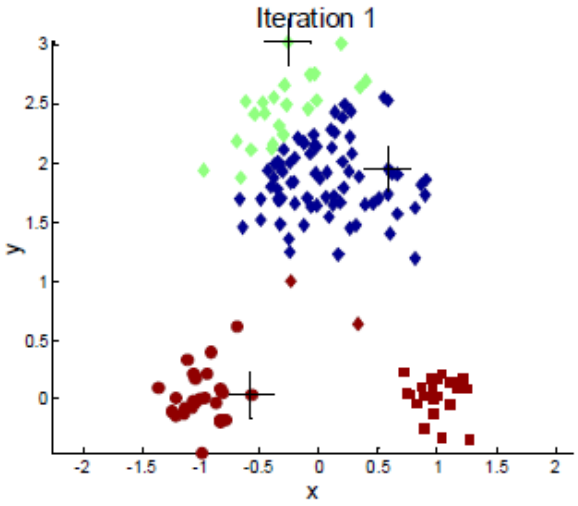


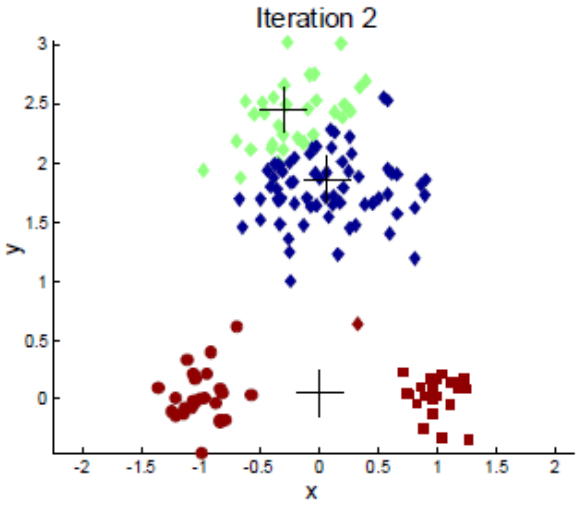


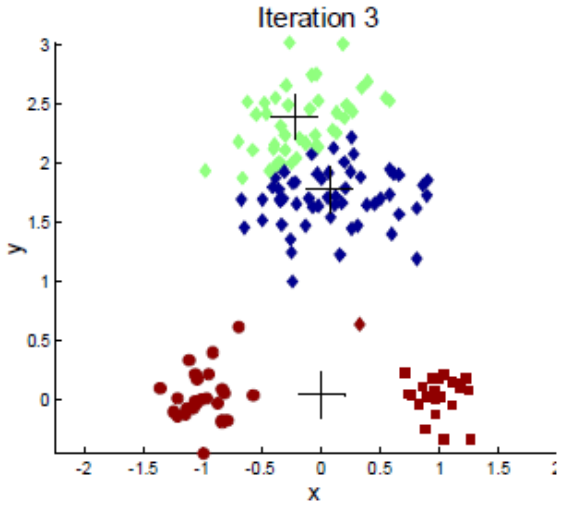


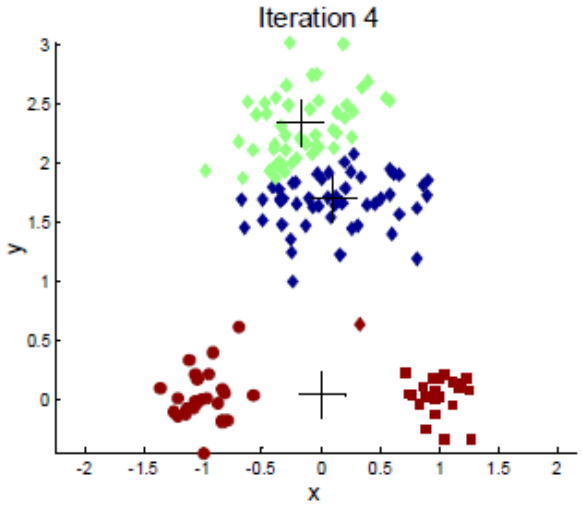


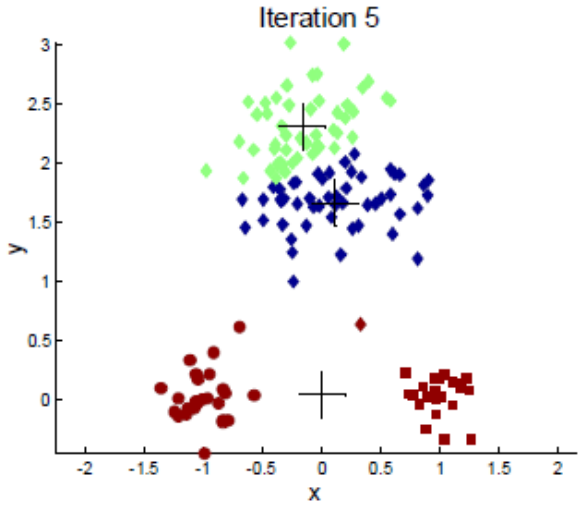
初始中心点对k-means聚类的影响（初始中心点选择较差的例子）











在实际中，可能需要尝试不同的初始中心点。挑选最好的聚类结果

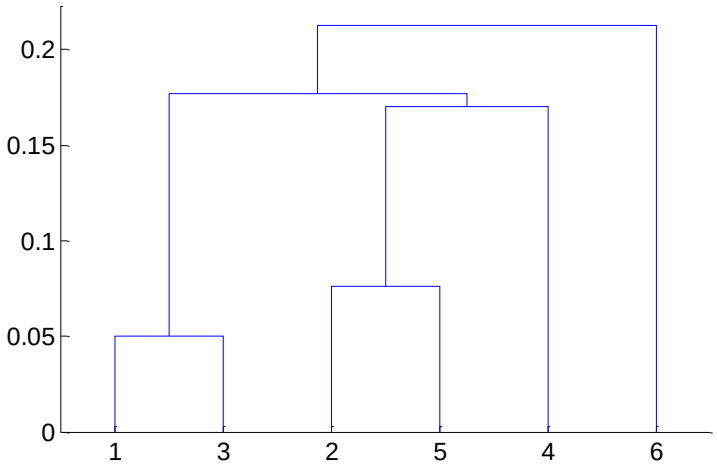
K-means 聚类例子

- 用K-means算法将右表中的8个点聚为三个簇，假设第一次迭代选择序号1、序号4和序号7当作初始点，请给出第一次执行后的三个聚类中心以及最后的三个簇
- 参考答案：最后三个簇(1,4,8)、(3,5,6)、(2,7)

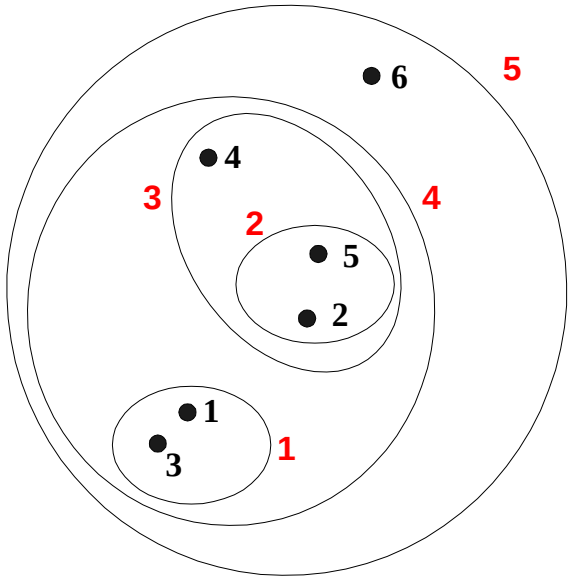
序号	属性1	属性2
1	2	10
2	2	5
3	8	4
4	5	8
5	7	5
6	6	4
7	1	2
8	4	9

层次聚类（Hierarchical Clustering）

- 产生层次聚类的两种基本方法:
  - **凝聚的（自下而上）**。从点作为个体簇开始, 每一步合并**两个最接近**的簇。这需要定义簇的邻近性概念。
  - **分裂的（自上而下）**。从包含所有点的某个簇开始, 每一步分裂一个簇, 直到仅剩下单点簇。在这种情况下, 需要确定每一步分裂哪个簇, 以及如何分裂。
  - 一般而言，凝聚式聚类更为常见
- 层次聚类的图显示有两种表示方式:
  - 树状图(dendrogram)。
  - 嵌套簇图(nested cluster diagram)。



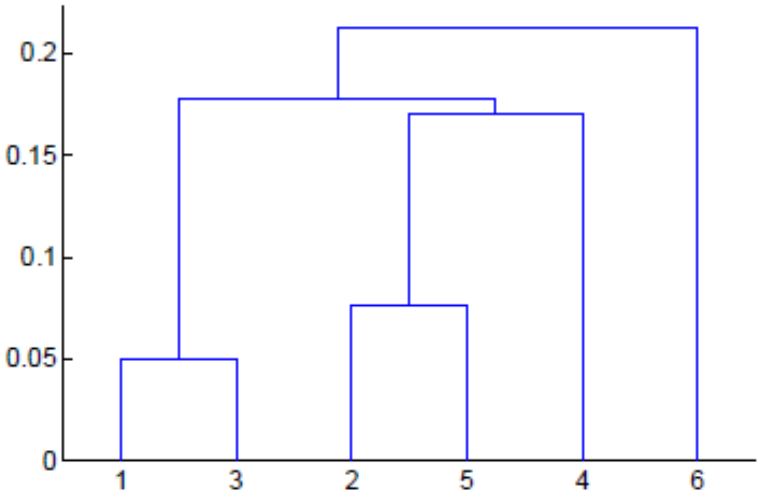
树状图



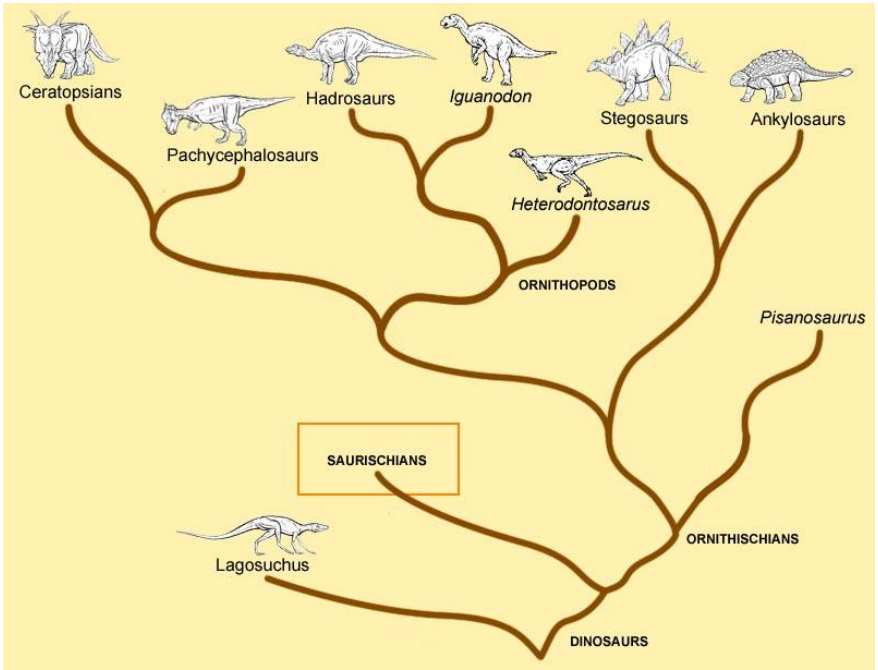
嵌套簇图

层次聚类（Hierarchical Clustering）

- 假设类别之间存在层次结构，将样本聚到层次化的类中，生成一组嵌套的树状簇。
- 可以通过树状图进行可视化
  - 通过切割想要的簇数的聚类。
  - 无需提前指定 K。
  - 可能对应于有意义的分类法。



树状图



“一颗倒立的树”

凝聚方法（Agglomerative Methods）(层次聚类方法之一)

- 自下而上的方法: 从个体点作为簇开始, 相继合并两个最接近的簇, 直到只剩下一个簇

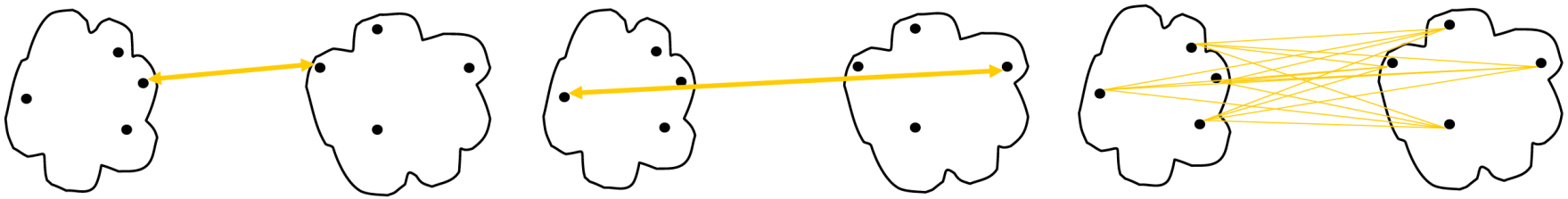
- 将每个数据点看作一个簇;
- 计算邻近度矩阵;
- 合并最近的一对簇;
- 重复直到只剩下一个簇;

- 如何计算簇之间的**邻近度**?

邻近度矩阵：用于存储两两簇之间的邻近度

凝聚方法-邻近性度量

- 簇之间的邻近性通常用特定的簇类型定义，主要有三种定义方式:
- **单链 (single link或MIN)**。MIN定义簇的邻近度为**不同簇的两个最近的点之间的邻近度**, 或者说不同的结点子集中两个节点之间的最短边。
- **全链 (complete link或MAX)**。MAX取**不同簇中两个最远点之间的邻近度**作为簇的邻近度, 或者说不同结点子集中两个结点之间的最长边。
- **组平均 (group average)**。定义簇邻近度为取自**不同簇的所有点对邻近度的平均值**(平均边长)。
- 如果两个点之间的邻近度度量是**距离(相异度)**，则MIN和MAX两个名字有提示作用，即**值越小表示点越接近(单链“小中取小”，全链“大中取小”)**。
- 如果两个点之间的邻近度度量是**相似度**，则**值越大表示点越接近(单链“大中取大”，全链“小中取大”)** 。



单链(MIN)　　全链(MIN)　　组平均

单链和全链层次聚类步骤

□基于距离(相异度)的层次聚类

- 1. 单链层次聚类步骤（小中取小）：

  ①计算不同簇的两个最近的点之间的距离,再找出距离最小的两个簇，第一个合并;

  ② 按照“小中取小”的原则依次合并剩余点，直至合并完所有点。

- 2. 全链层次聚类步骤（大中取小）：

  ① 计算不同簇的两个最远的点之间的距离,再找出距离最小的两个簇，第一个合并;

  ② 按照“大中取小”的原则依次合并剩余的点，直至所有点合并完成。

□基于相似度矩阵的层次聚类

- 1. 单链层次聚类步骤（大中取大）：

  ① 找出所有点相似度最大的两个点，第一个合并;

  ② 按照“大中取大”的原则进行合并剩余的点，直至所有点合并完成为止。

- 2. 全链层次聚类步骤（小中取大）：

  ① 找出所有点相似度最大的两个点，第一个合并;

  ② 按照“小中取大”的原则进行合并剩余的点，直至所有点合并完成为止。

基于单链的聚类例子（小中取小）

邻近度矩阵（两两簇间最近的点之间的距离）

	BA	FI	MI	NA	RM	TO
BA	0	662	877	255	412	996
FI	662	0	295	468	268	400
MI	877	295	0	754	564	138
NA	255	468	754	0	219	869
RM	412	268	564	219	0	669
TO	996	400	138	869	669	0

单链

BA:巴里，FI:佛罗伦萨，MI:米兰，NA:那不勒斯，RM:罗马，TO:都灵

基于单链的聚类例子（小中取小）

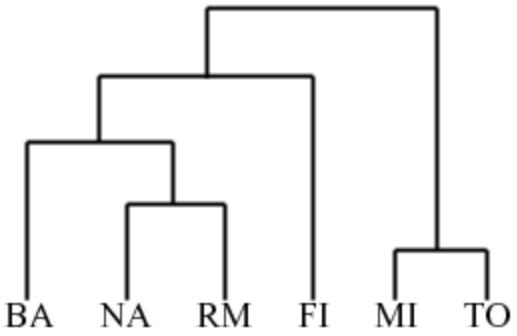
	BA	FI	{MI,TO}	NA	RM
BA	0	662	877	255	412
FI	662	0	295	468	268
{MI,TO}	877	295	0	754	564
NA	255	468	754	0	219
RM	412	268	564	219	0

	BA	FI	{MI,TO}	{NA,RM}
BA	0	662	877	255
FI	662	0	295	268
{MI,TO}	877	295	0	564
{NA,RM}	255	268	564	0

基于单链的聚类例子（小中取小）

	BA/NA/RM	FI	MI/TO
BA/NA/RM	0	268	564
FI	268	0	295
MI/TO	564	295	0

	BA/FI/NA/RM	MI/TO
BA/FI/NA/RM	0	295
MI/TO	295	0



利用单链和全链对表7.4中6个点进行层次聚类

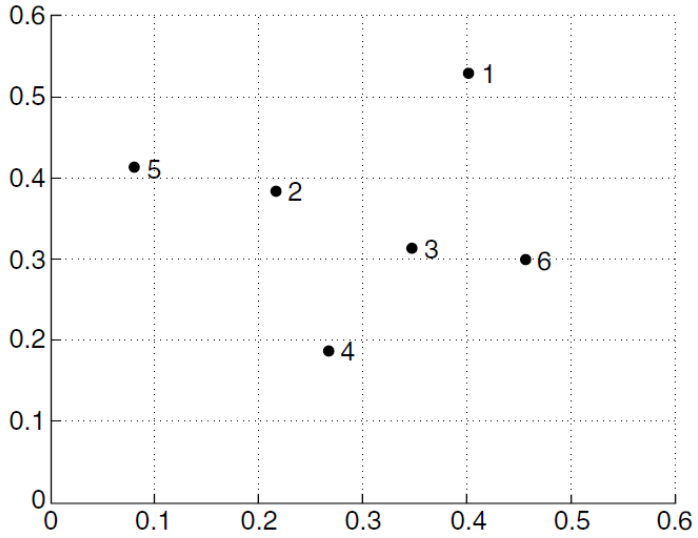


图7.15 6个二维点的集合

图7.3 6个点的xy坐标

Point	x Coordinate	y Coordinate
p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

表7.4 **6个点的欧几里得距离邻近度矩阵**

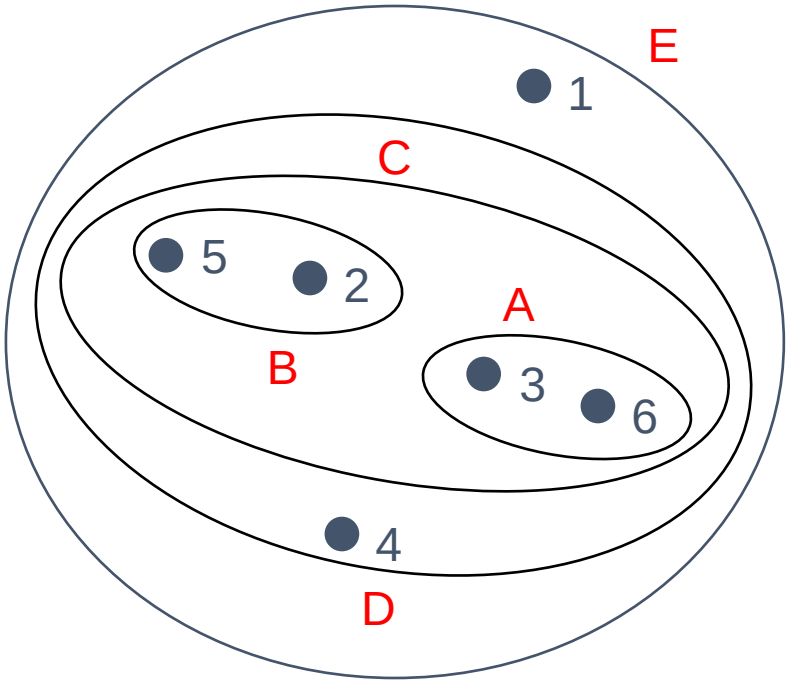
0.1483

0.1513

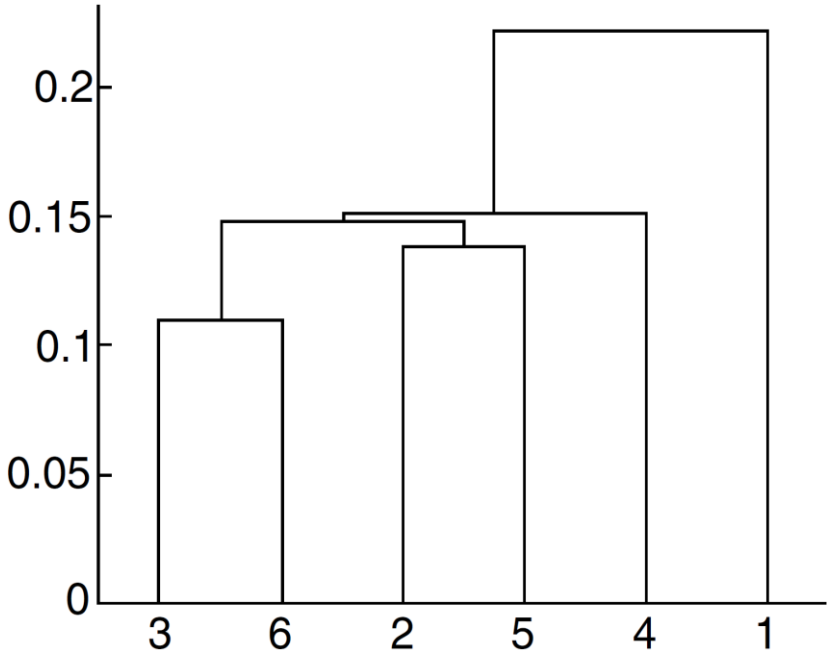
	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

表7.4见《数据挖掘导论》(第2版) P323-P325

单链 (Min) 聚类



单链聚类



单链树状图

具体计算步骤参考教材P324的例7.4

表7.4 **6个点的距离邻近度矩阵**

	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

单链 (Min) 层次聚类计算流程例子

- (1)首先用6个样本构建6个单点类，$G_i = \{p_i\}, i = 1,2 \ldots, 6$, 样本之间距离变为类之间距离，6个类之间的距离矩阵亦为D。
- (2)由矩阵D可看出, $D_{36} = D_{63} = 0.11$为最小，把 G_3和G_6合并为一个新类，记作 $G_7 = \{p_3, p_6\}$
- (3)计算G_7与G_1, G_2, G_4, G_5 之间的最短距离, 有

  $D_{71} = 0.22$，$D_{72} = 0.15$，$D_{74} = 0.15$, $D_{75} = 0.28$, 同时计算

  $D_{12} = 0.24$，$D_{14} = 0.37$，$D_{24} = 0.20$，$D_{25} = 0.14$, $D_{45} = 0.29$,

  由计算可知 D_{25} 最小, 故 G_2与G_5合并成一个新类, 记作 $G_8 = \{p_2, p_5\}$

- 余下四个簇 G_1, G_4, G_7, G_8。计算

  $D_{81} = 0.24$，$D_{84} = 0.20$，$D_{87} = 0.1483 \approx 0.15$，$D_{75} = 0.28$,

  $D_{14} = 0.37$, $D_{17} = 0.22$，$D_{74} = 0.1513 \approx 0.15$,

  由计算可知 D_{87} 最小, 故 G_7与G_8合并成一个新类, 记作 $G_9 = \{p_2, p_3, p_5, p_6\}$

- 余下 G_1, G_4, G_9。计算

  $D_{14} = 0.37, D_{19} = 0.22, D_{49} = 0.15$,

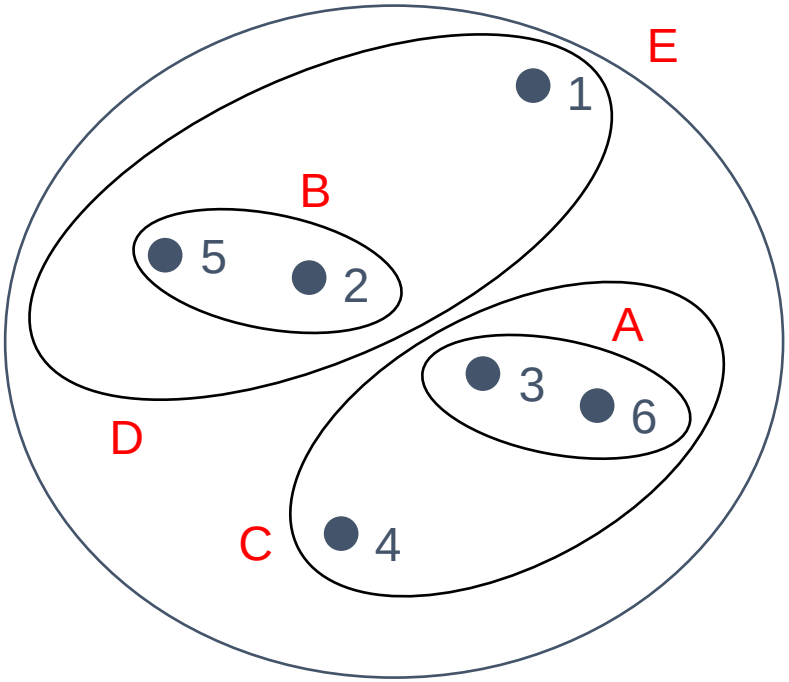
  由计算可知 D_{49} 最小, G_4与G_9合并成一个新类G_{10}

- 最终余下两类 G_1与G_{10}合并成一个新类 G_{11}
- 至此，所有样本都合并到一个大类 G_{11}中

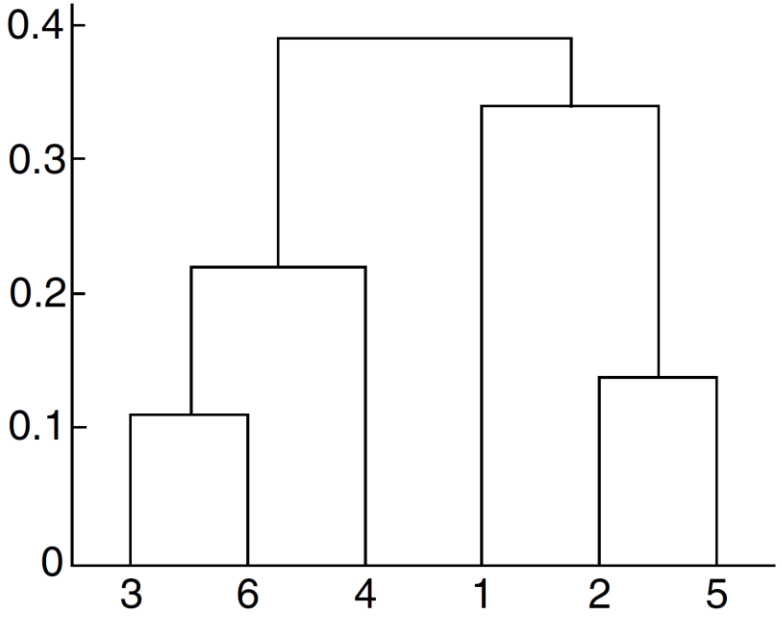
表7.4 **6个点的距离邻近度矩阵**

	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

全链 (Max) 聚类（大中取小）



全链聚类



全链树状图

具体计算步骤参考教材P324的例7.5

表7.4 **6个点的距离邻近度矩阵**

	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

课下作业

- 下表为5个点的相似度矩阵，将这5个点进行单链和全链聚类，并画出层次树状图

5个样本点之间的相似度矩阵					
	p1	p2	p3	p4	P5
p1	1.00	0.30	0.41	0.55	0.35
p2	0.30	1.00	0.64	0.47	0.98
p3	0.41	0.64	1.00	0.44	0.85
p4	0.55	0.47	0.44	1.00	0.89
p5	0.35	0.98	0.85	0.89	1.00

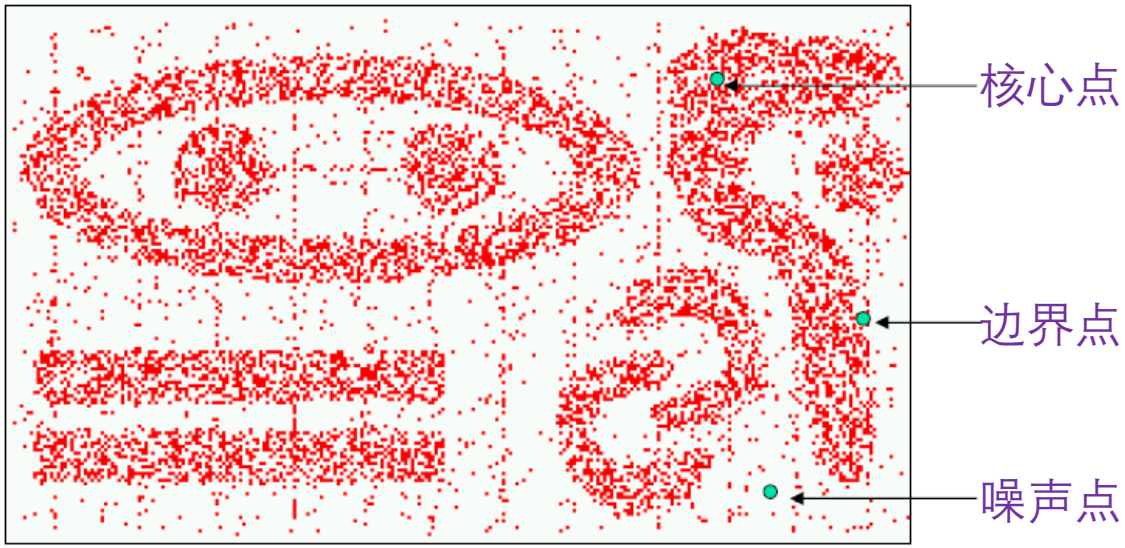
基于密度的聚类方法（Density Based Methods，DBSCAN）

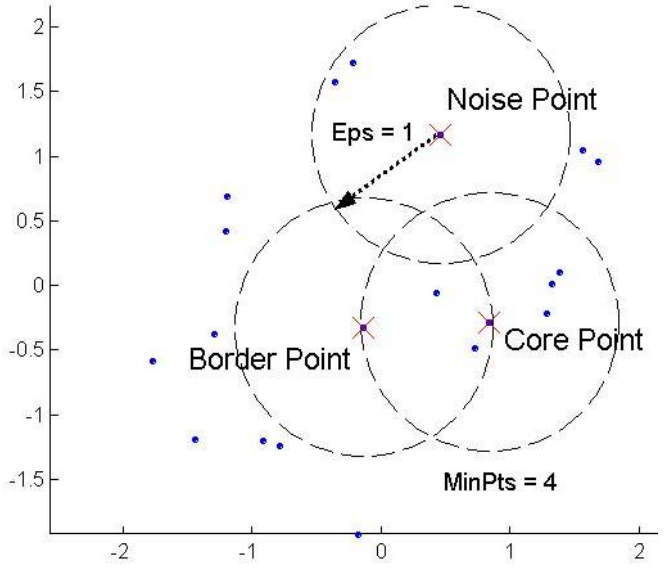
- 基于密度的方法：根据密度完成样本数据的聚类，一般假定**类别**可以通过**样本分布的紧密程度**确定。
- 基本假设：只有达到一定密度，才足以成为一个簇
- 密度：指定样本一定半径的样本数量
- 密度聚类算法从样本密度的角度来考察样本之间的可连接性，并基于可连接样本不断扩展聚类簇以获得最终的聚类结果。
- 典型的基于密度方法-DBSCAN (Density-based Spatial Clustering of Application with Noise): 该算法通过不断生长足够高密度区域来进行聚类; 它能从含有噪声的空间数据库中发现任意形状的聚类。此方法将一个聚类定义为一组“密度连接”的点集。

基于密度的方法（DBSCAN）

基于密度的方法将样本点分类为:

- **核心点** (Core point，**稠密区域内部的点**)。核心点的定义为: 如果该点的给定邻域内的点的个数超过给定的阈值MinPts (MinPts由用户指定), 则这些点为核心点。
- **边界点** (Border point，**稠密区域边缘上的点**)。边界点不是核心点, 但它落在某个核心点的邻域内。边界点可能落在多个核心点的邻域内。
- **噪声点**或背景点 (Noise point，**稀疏区域中的点**)。噪声点是即非核心点也非边界点的任何点。



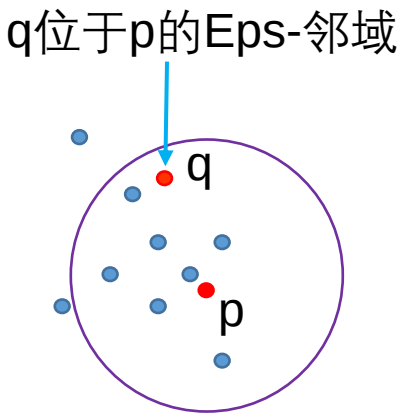


核心点、边界点和噪声点示意图

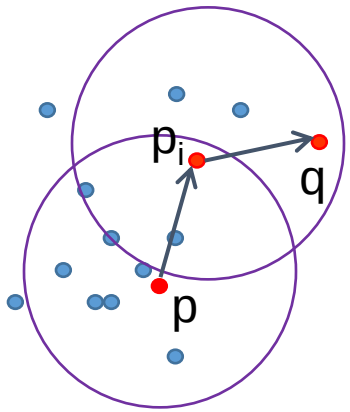
基于密度的方法（DBSCAN）

- DBSCAN算法中有两个重要参数: **邻域半径** Eps 和**阈值** MinPts。前者为定义密度时的邻域半径, 后者为定义核心点时的阈值。
- **DBSCAN中的几个定义**:

- **Eps领域**: 给定样本点p，其半径为 Eps 内的区域称为该样本的 Eps 邻域。
- **核心点:** 如果给定点Eps 邻域内的样本数大于等于MinPts, 则该点为核心点。
- **直接密度可达**: 对于样本集合D, 如果样本点q 在p 的Eps邻域内, 并且p 为核心点, 那么点q 从点p 直接密度可达(又称密度直达)。
- **密度可达**: 对于样本集合D, 给定一串样本点p_1 , p_2 , … p_n , $p=p_1$, $q=p_n$, 假定样本p_i 从 p_{i-1} 直接密度可达, 那么点q 从点p 密度可达。



直接密度可达(密度直达)

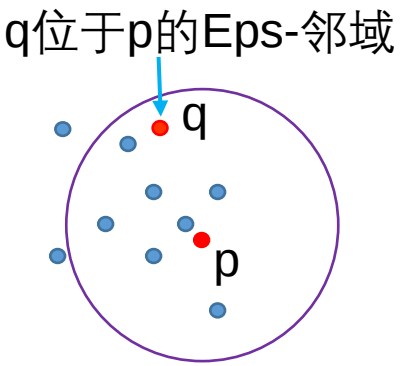


密度可达

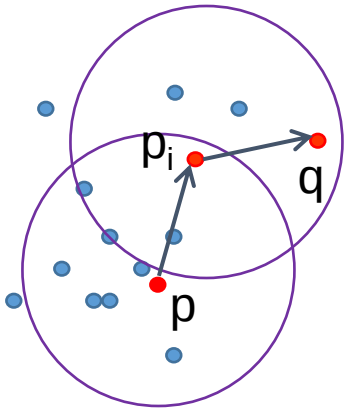
对于p和q，若存在样本序列$p_1,p_2,\ldots,p_n$使得p_i和 p_{i+1}密度直达，且$p=p_1$，$q=p_n$，则称点p和点q密度可达

基于密度的方法（DBSCAN）

密度相连: 对于样本集合D 中的任意一点o, 如果存在点p 到点o 密度可达, 并且点q 到点o 密度可达, 那么点q 到点 p 密度相连。

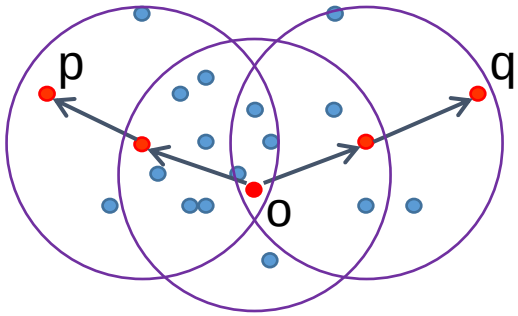


直接密度可达



密度可达

对于p和q，若存在样本序列$p_1,p_2,\ldots,p_n$使得p_i和 p_{i+1}密度直达，且$p=p_1$, $q=p_n$, 则称点p和点q密度可达



密度相连

存在点o使得点p和点o密度可达,点q和点o密度可达，则称点p和点q密度相连

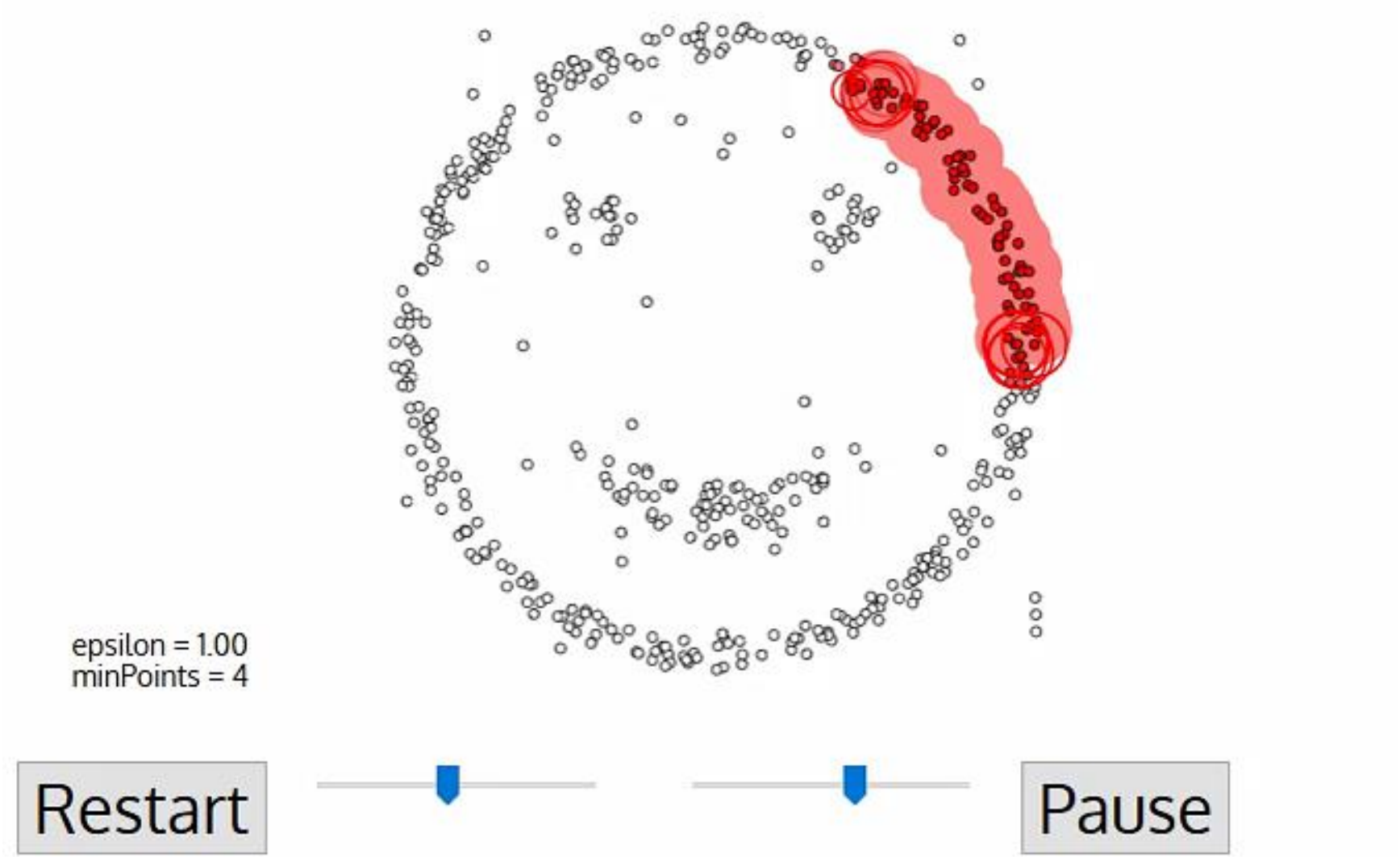
DBSCAN 算法聚类流程

- 输入: 样本集合D，聚类半径Eps, 密度阈值 MinPts
- 输出: 目标类簇集
- 方法:
  - 1. 随机选取未被处理的点p，判断输入点是否为核心点。
  - 2. 找出核心点的 Eps 领域中的所有密度可达点，形成一个新的簇。

    –遍历数据集D, 直到所有输入点都判断完毕;
  - 3. 针对该核心点的 Eps 邻域内所有密度可达点找到最大密度相连的样本点集合, 产生最终的簇结果。
  - 4. 重复执行第2步和第3步，直到数据集D中所有点都为“已处理”状态。

DBSCAN 算法聚类过程演示

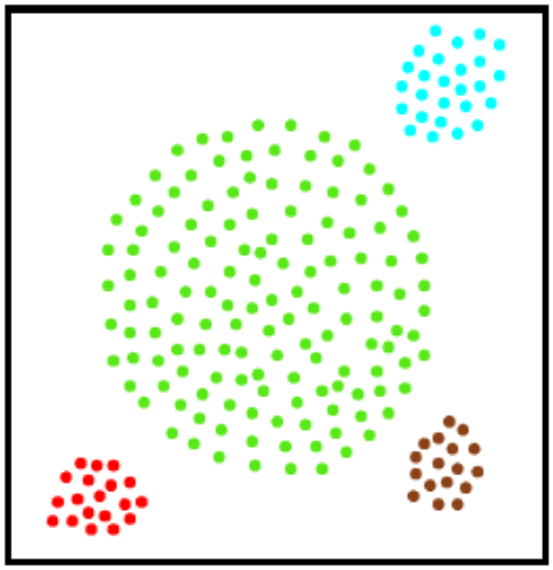
邻域半径Eps=1和阈值MinPts=4



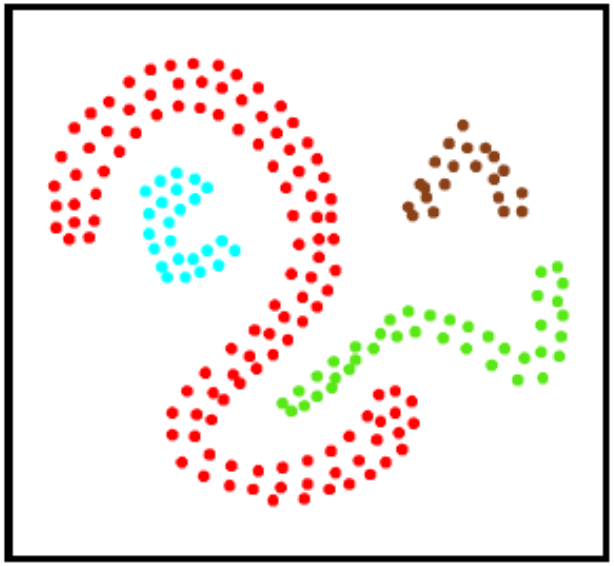
https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

DBSCAN 聚类算法总结

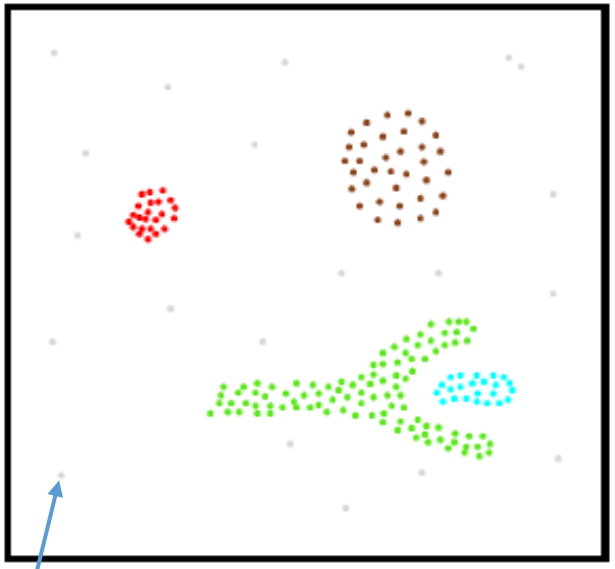
- DBSCAN将“簇”被定义为由密度可达关系导出的最大密度相连的样本集合(从核心点出发)，即为我们最终聚类的一个类别，或者说一个簇;
- 从随机选择的未见的样本点P开始;
- 如果P是核心点，则通过逐渐将密度可达的所有点添加到当前点集来构建簇;
- **噪声点被丢弃**（稀疏的暗色点，未被标记）;
- 产生任意形状的簇;
- 对噪声鲁棒;
- 不需要事先指定类别 K 值， 可根据形状自动确定。



DBSCAN和K-means都可以很好地聚类(球形)



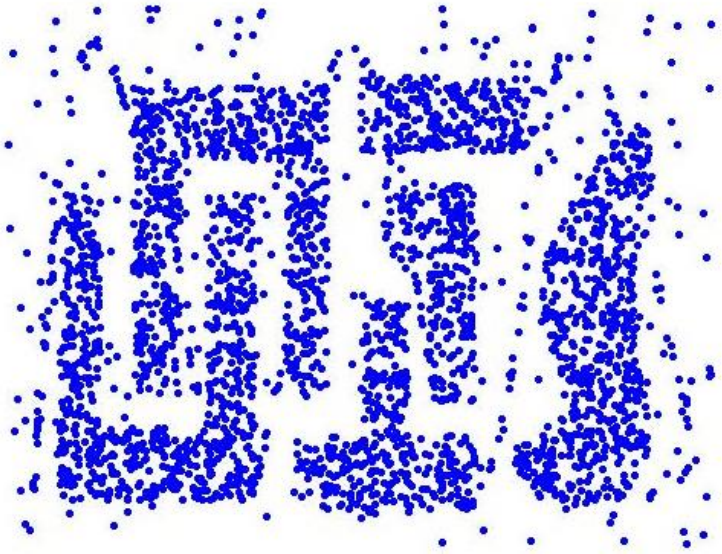
DBSCAN聚类



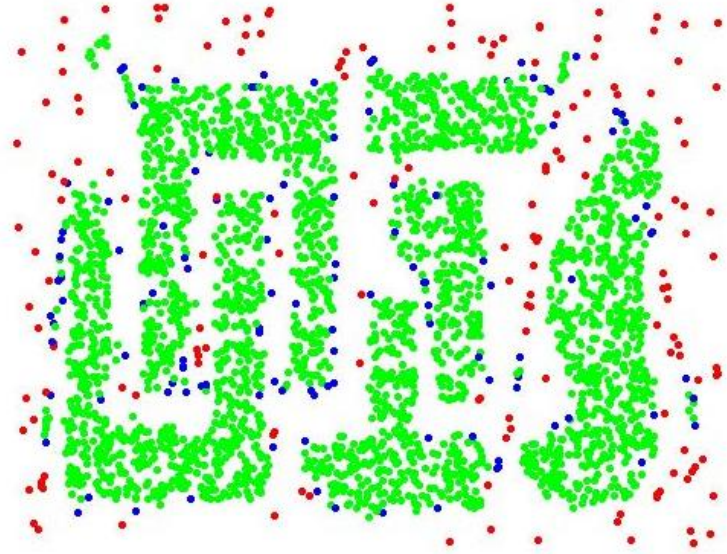
DBSCAN聚类

基于密度的方法（DBSCAN）-例子

- **实例**: 为了解释DBSCAN, 将相对复杂的二维数据集中发现的簇进行聚类。
- 该数据集包含有3000个二维点。该数据的Eps阈值通过对每个点到其第四个最近邻的距离排序绘图, 并识别急剧变化处的值来确定。其中参数为: Eps = 10, Minpts = 4。



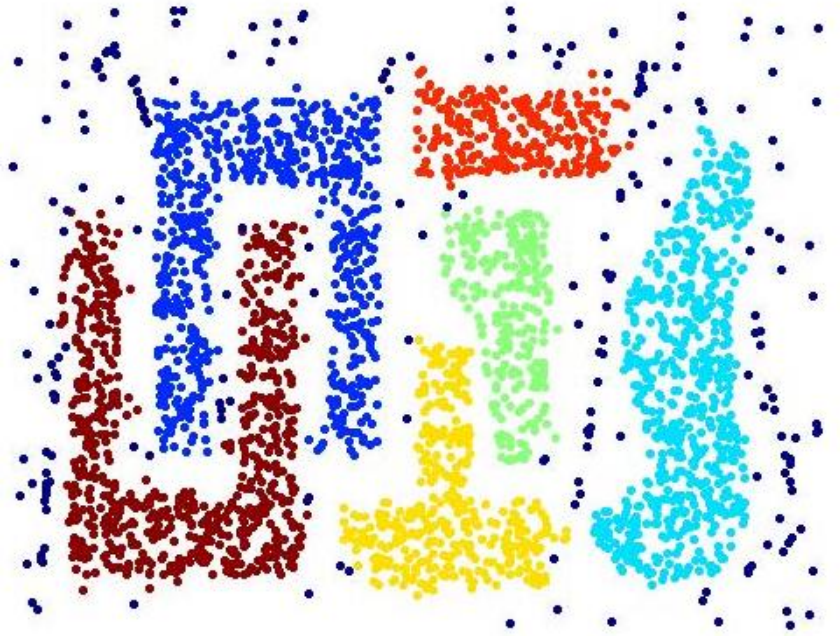
原始样本点



绿色点为核心点, 蓝色点为边界点, 红色点为噪声点

DBSCAN 聚类结果示意图

原始样本点



3000个二维点的DBSCAN聚类结果

周边的噪声除去，内部的数据很好的聚类

DBSCAN 的优缺点

DBSCAN 的主要优点：

- **能发现任意形状的簇**。可对任意形状的稠密数据集进行聚类，相对地，K-means之类的聚类算法一般只适用于凸数据集。
- **对噪点不敏感**。可以在聚类的同时发现异常点，对数据集中的异常点不敏感。

DBSCAN 的主要缺点：

- **对两个参数的设置敏感**，即邻域半径 eps 、阈值 MinPts。调参相对于传统的K-means之类的聚类算法稍复杂，主要需要对邻域半径Eps，邻域样本数阈值MinPts联合调参，不同的参数组合对最后的聚类效果有较大影响。
- **DBSCAN 使用固定的参数识别聚类**。如果样本集的密度不均匀、聚类间距差相差很大时，聚类质量较差，这时用DBSCAN聚类一般不适合。
- **如果数据样本集越大，则收敛时间越长**。如果样本集较大时，聚类收敛时间较长。

DBSCAN 密度聚类

- DBSCAN 算法作者获得 ICDM 2013 Research Contributions Award

[PDF] A density-based algorithm for discovering clusters in large spatial databases with noise

M Ester, HP Kriegel, J Sander, X Xu - kdd, 1996 - cdn.aaai.org

… drawbacks when applied to large spatial databases. In this paper, we presented the clustering algorithm DBSCAN which relies on a density-based notion of clusters. It requires only one …

☆ Save 99 Cite Cited by 32246 Related articles All 64 versions »