

点击编辑主字幕样式 永大

大数据分析|何铁科
[https:// hetieke.cn](https://hetieke.cn)



南京大学

南京大学

高维数据

给定一个数据点云，我们想要了解它的结构



问题定义

“给定一组点，以及点与点之间的距离的概念，将点分组到一些簇中，因此

§ 一个集群的成员彼此接近或相似 § 不同集群的成员是不相似的

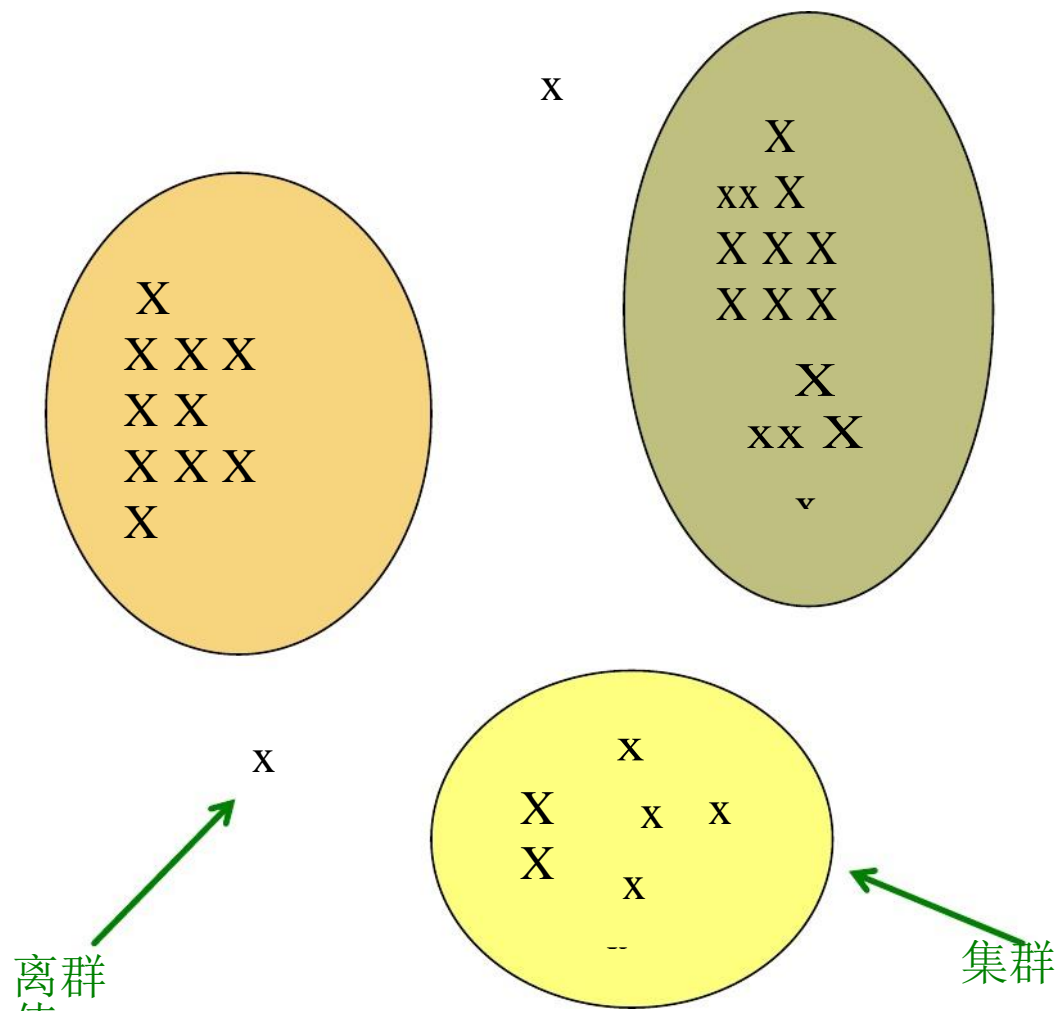
通常

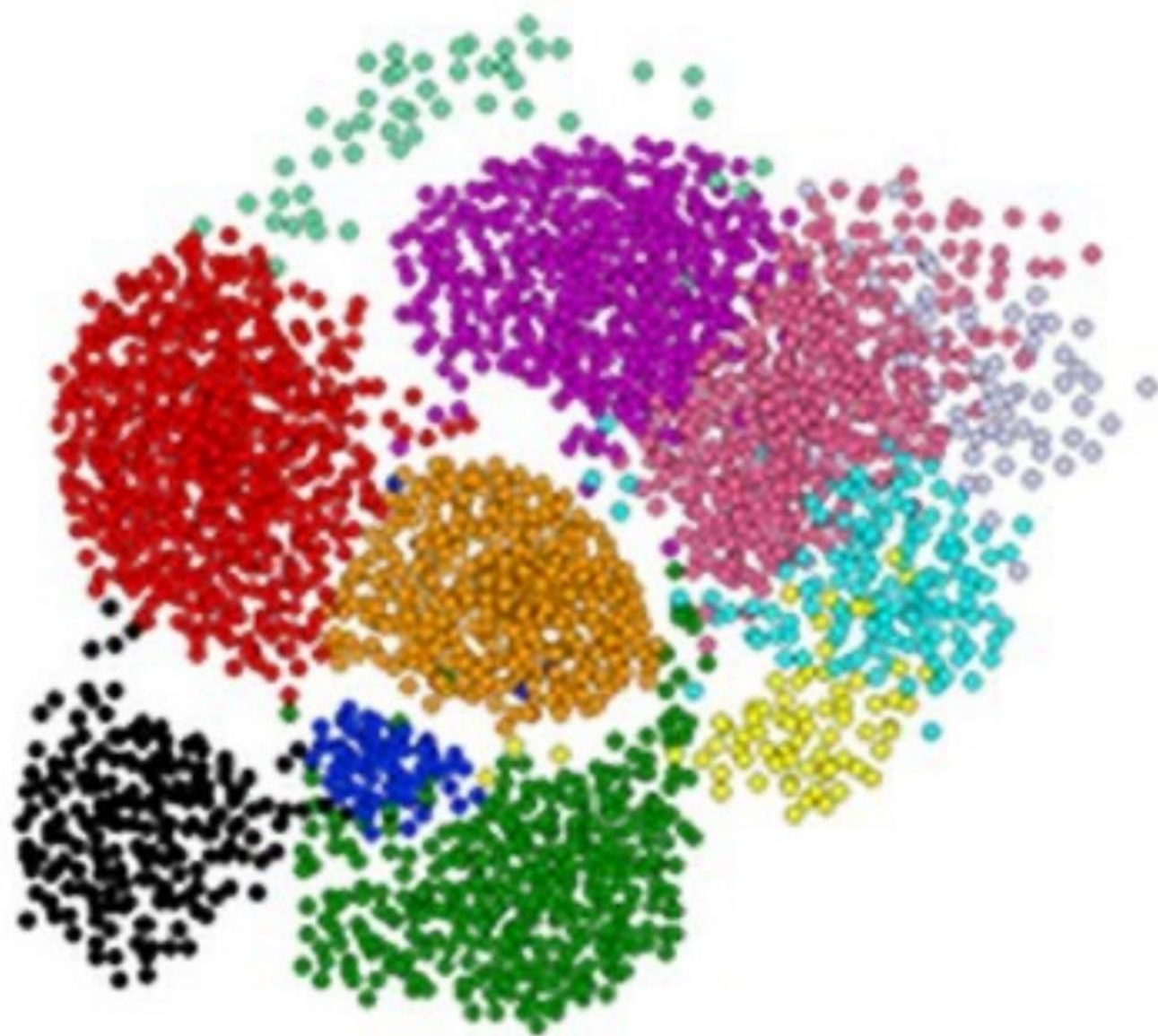
§ 点处于高维空间

§ 相似性是用距离测度来定义的

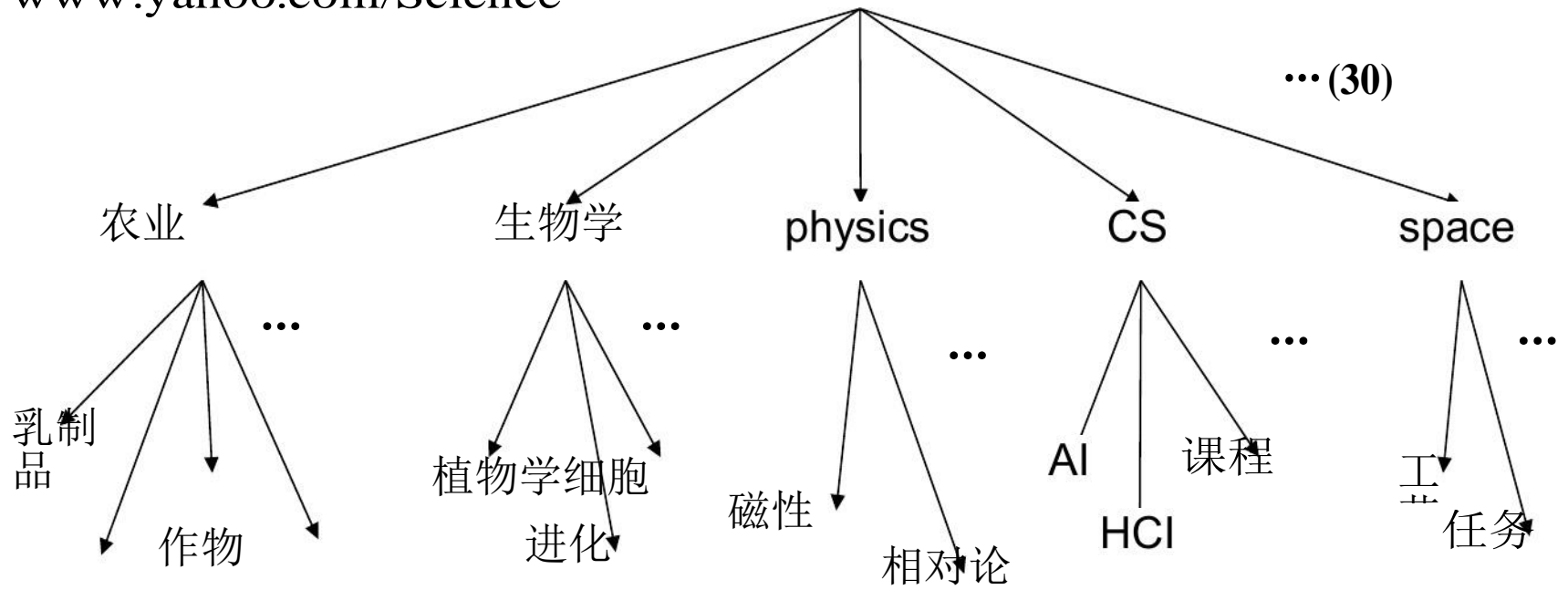
§ 欧几里得，余弦，杰卡德，编辑距离，...

类簇和离群点





www.yahoo.com/Science



难点与挑战

“二维聚类看起来很容易” “聚类少量数据看起来很容易” “在大多数情况下，外表是**不会**骗人的。

许多应用不是只涉及 2 个维度，而是 10 个或 10 000 个维度

‘**高维空间看起来不一样**：几乎所有点对之间的距离都差不多

星系问题

‘20 亿个“天体”的目录
通过 7 个维度(频带)
的辐射来表示
天体’ 问题:聚集成相似的天体
， 例如星系、附近的恒星、类星体等
’ 数字巡天



音乐CD(一)

‘直观上:音乐分为几个类别, 顾客喜欢几个类别 § 但类别到底是什么?

‘代表一组顾客购买的 CD:

相似的 cd 有相似的客户群, 反之亦然

音乐CD(二)

所有 cd 的空间:

想想每个顾客都有一个昏暗的空间

§ 一个维度的值可能只有 0 或 1 § 一张 CD 是这个空间中的一个点 (x_1, x_2, \dots, x_k) , 其中 $x_i = 1$ iff the i^{th} customer buying the CD

“对亚马逊来说, 维度是数千万

! 任务:找到一组相似的 cd

文档

寻找主题:

“用向量(x_1, x, \dots, x_{k2})表示一个文档, 其中 $x_i = 1$ iff 第 i^{th} 个单词(以某种顺序)出现在文档中 §
实际上 k 是无限的并不重要;也就是说, 我们不限制单词的集合

“具有相似单词集文档可能是关于相同的主题

距离度量a

“理想:语义相似。 ‘实用:术语统计相似度 § 我们将使用余弦相似度。 § 文档作为向量。

§ 对于许多算法来说，更容易根据文档之间的 *距离*(而不是相似度)来思考。

§ 我们将主要讨论欧几里得距离，但实际实现使用余弦相似度

距离度量b

- “在 cd 中，我们可以选择将文档看作词集或片集：
 - § 集作为向量:通过余弦距离度量相似性
 - 集合作为集合:通过 Jaccard 距离度量相似性
 - § 集作为点:通过欧几里得距离度量相似性

硬聚类vs.软聚类

- | 硬聚类:每个文档正好属于一个聚类

 - § 比较常见, 也比较容易做

软聚类:一个文档可以属于多个聚类。

- | § 对于创建可浏览的层次结构等应用程序更有意义

 - § 你可能想把一双运动鞋分为两类:(i)运动服装和(ii)鞋子

 - § 你只能用一种软聚类方法来做到这一点。

什么是集群?

也叫无监督学习，有时统计学家称之为分类，心理学家称之为分类，营销人员称之为细分

- 根据数据的类别进行组织
 - 高类内相似性
 - 类间相似度低
- 直接从数据中找到类标签和类的数量(与分类相反)。
- 更非正式地，在对象之间找到自然的分组。

集群应用程序示例

营销:帮助营销人员在他们的客户群中发现不同的群体，然后利用这些知识来制定有针对性的营销方案

‘土地使用:在地球观测数据库中识别类似土地使用的区域

保险业:确定平均理赔费用较高的车险保单持有人群体

•城市规划:根据房屋类型、价值和地理位置确定房屋群

地震学:观测到的地震震中应该聚集在大陆断层附近

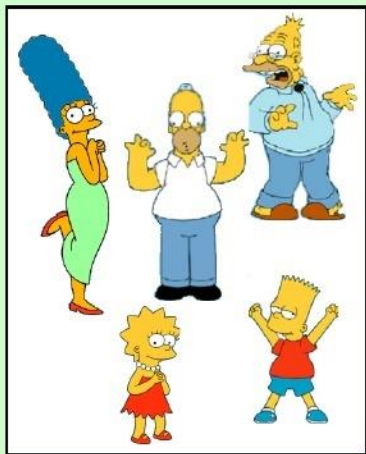
这些对象之间的自然分组是什么？



这些对象之间的自然分组是什么？



聚类是主观的



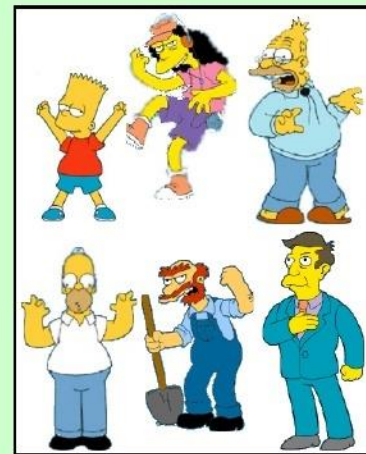
辛普森的家人



学校的员工



女性



男性

什么是相似性？

相似:相似的性质或状态;相似;相似之处;As, 特征的相似性。韦氏词典



相似性很难
定义，但是……
“我们一看到它
就知道”

相似的真正含义
是一个
哲学
问题。我们将
采取更加
务实
的做法。

定义距离测度

定义: 让 O_1 和 O_2 从两个对象

可能物体的宇宙。和之间 O_1 的距离(不相似度)是 O_2 一个实数，用 $D(O, O_1)_{O_2}$ 表示。



0.23

彼得彼得



3



342.7

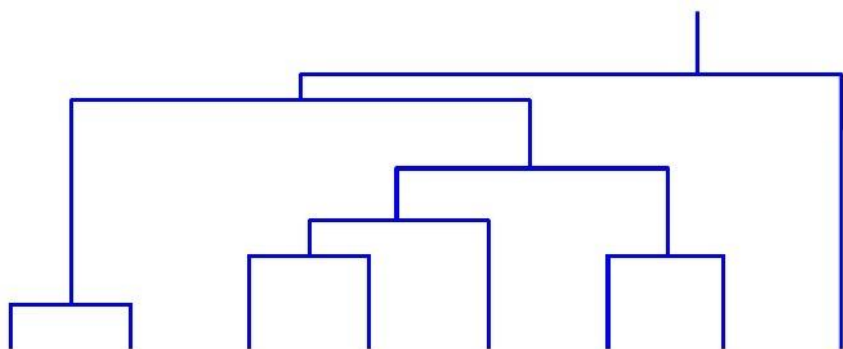
编辑距离示例

可以将任意字符串 Q 转换为字符串 C ，只需使用替换、插入和删除。

假设这些操作符中的每一个都有与之相关的代价。

两个字符串之间的相似度可以定义为从 Q 到 C 的最便宜变换的代价。

注意，现在我们已经忽略了如何找到这个最便宜的转换的问题



“彼得”和“彼得”这两个名字有多相似？

假设如下代价函数

替换
1

一个
单位
一个

$D(\text{彼得}, \text{彼得})$

彼得



坑

代换(i 换 e)



皮特

插入(o)

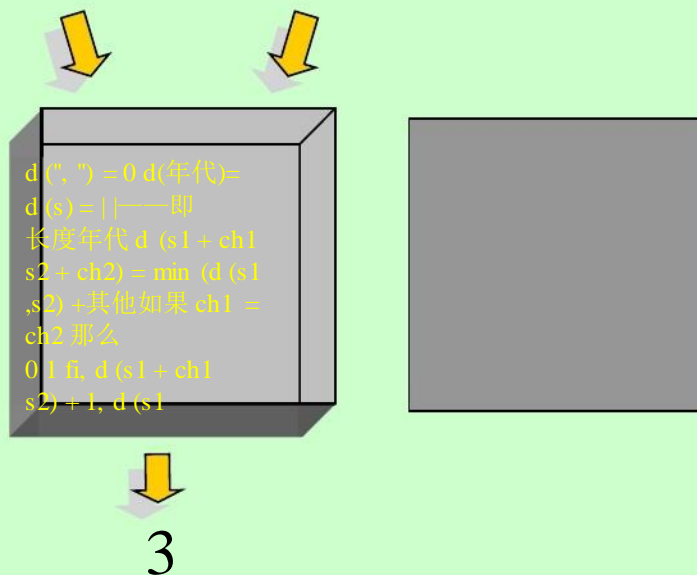


彼得

删除(e)

彼得

彼得



当我们窥视其中一个黑盒子的内部

时，我们看到了一些关于两个变量的函数。这些函数可能非常简单，也可能非常复杂。

在这两种情况下，我们很自然地会问

距离测量应该具有什么属性性质？

- $d(a, b) = d(b, a)$

对称

- $D(A, A) = 0$

- $D(A, B) = 0 \iff A = B$

自相似正(分离)的恒定性

- $D(A, B) \leq D(A, C) + D(B, C)$

三角不等式

理想距离测量属性背后的直觉

$$D(A,B) = D(B,A)$$

对称

否则，您可以声明“Alex 长得像 Bob，但 Bob 一点也不像 Alex。”

$$D(A,A) = 0$$

自相似性恒常性

否则你可以说“亚历克斯看起来比 Bob 更像 Bob。”

$$D(A,B) = 0 \text{ IIf } A=B$$

积极性(分离)

否则，你的世界里有不同的物体，但你无法区分。

$$D(A,B) \leq D(A,C) + D(B,C) \text{ 三角不等式}$$

否则你可以说“亚历克斯很像鲍勃，亚历克斯很像卡尔，但鲍勃很不像卡尔。”

聚类算法的理想性质

- 可扩展性(时间和空间方面)
- 处理不同数据类型的能力
- 确定输入参数对领域知识的最低要求
- 能够处理噪声和异常值
- 对输入记录的顺序不敏感
- 合并用户指定的约束
- 可解释性和可用性

将数据划分为相似对象组。

聚类方法论

分层的方法

- § 凝聚算法

- § 分裂式算法

分区方法

- § 重定位算法

- § 概率聚类

- § K-medoids 方法

- § k 均值方法

- § 基于密度的算法

 - § 基于密度的连接聚类

 - § 密度函数聚类

基于网格的方法

基于符号数据共现的约束聚类方法

机器学习中使用的聚类算法 § 梯度下降和人工神经网络

- § 进化方法

可扩展的聚类算法

用于高维数据的算法

- § 子空间聚类

贯穿始终的一些话题

算法可以处理的属性类型

可扩展性到大型数据集

能够处理高维数据

能够发现不规则形状的簇

处理离群值

时间复杂度(当没有混淆时，我们使用术语复杂度)数据顺序依赖

标注或赋值(硬的或严格的 vs. 软的模糊)

对先验知识和用户定义参数的依赖结果的可解释性

划分方法

分区法:构造一个数据库的分区 D

N 个对象组成一组 k 个簇

给定 k , 找到 k 个簇的一个划分, 以优化所选的划分准则

§ 全局最优:穷举枚举所有分区 § 启发式方法: k -means 和 k -medoids 算法 § k -means(macquarie, 1967):每个簇都由簇的中心表示

§ k -中心点(k -medoids)或 PAM(围绕中心点的划分)(Kaufman & Rousseeuw, 1987):每个簇由簇中的一个对象表示

启发式

“启发式方法用于快速得出一个希望接近最佳可能答案的解决方案，或‘最优解决方案’。启发式方法是一种“经验法则”，一种有根据的猜测，一种直觉的判断或简单的常识。

“启发式是解决问题的一般方法。
Heuristics 作为名词是启发式方法的另一个名称。

启发式算法

┆ 仿动物类的算法：

§ 粒子群优化 § 蚂蚁优化
§ 鱼群算法 § 蜂群
算法等 ┆ 仿植物类的算
法： § 向光性算法 §
杂草优化算法 ┆ 仿人类
的算法： § 和声搜索
算法

k - means

假定文档是实值向量。‘基于簇中点的^o质心(又名重心或平均值)聚类, c :

$$\mu(c) = \frac{1}{n_c} \sum_{x \in c} x$$

$$|c|_x = \hat{1}_c$$

‘将实例重新分配到簇是基于到当前簇质心的距离。§ (或者一个人可以等价地用相似度来表达)

k - means算法(年代)

假定欧氏空间/距离

、从选择 k 开始, k 是簇的数量

、通过每个簇选择一个点来初始化簇

§ 示例:随机选择一个点, 然后 $k-1$ 个其他点, 每个点都与前面的点尽可能远

填充集群

✕ 1) 对于每个点，将其放置在距离其当前质心最近的簇中

± 2) 所有点分配完毕后，更新 k 个簇的质心位置

′ 3) 重新分配所有点到它们最近的质心 § 有时会在簇之间移动点

′ 重复第 2 和第 3 步，直到收敛

§ 收敛: 点不在簇之间移动，质心稳定

终止条件

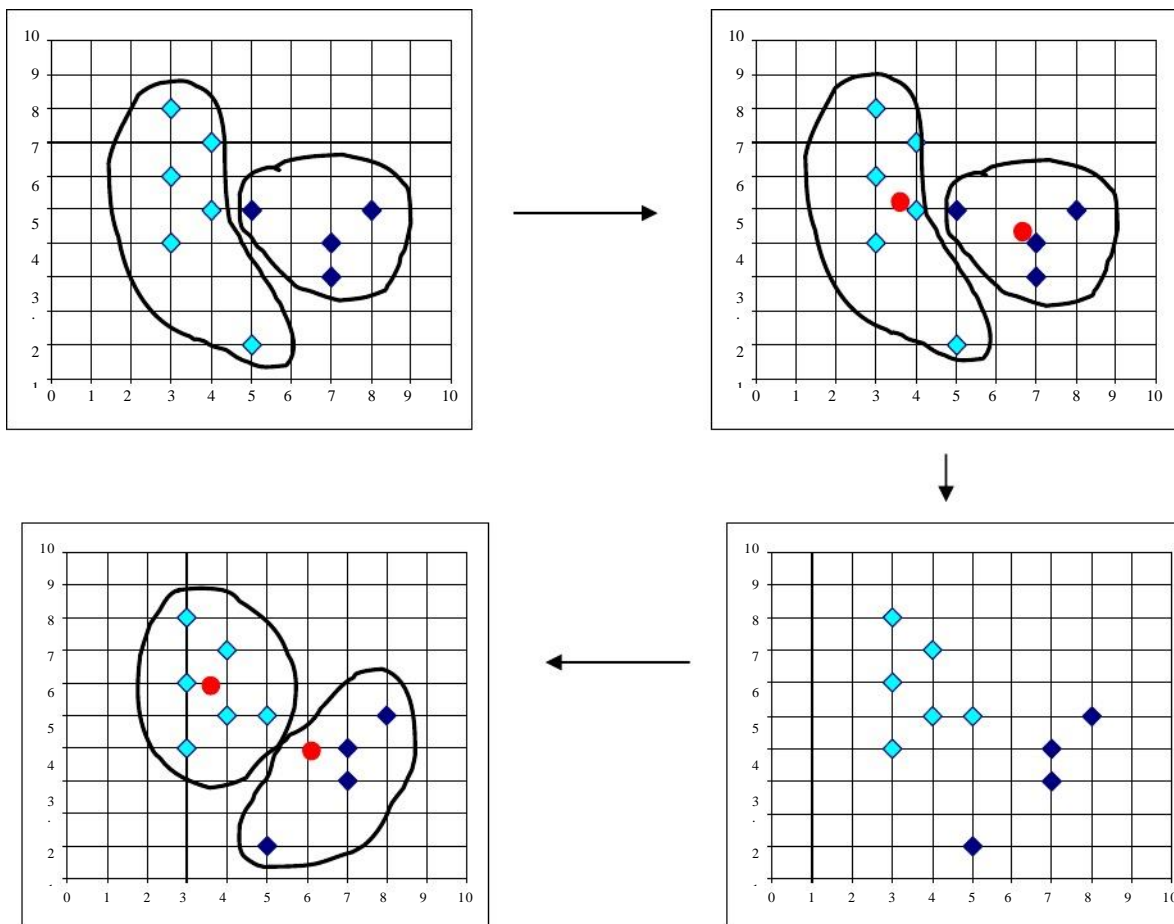
“几种可能性，例如 § 固定的迭代次数。 § 文档分区不变。 § 质心位置不变。



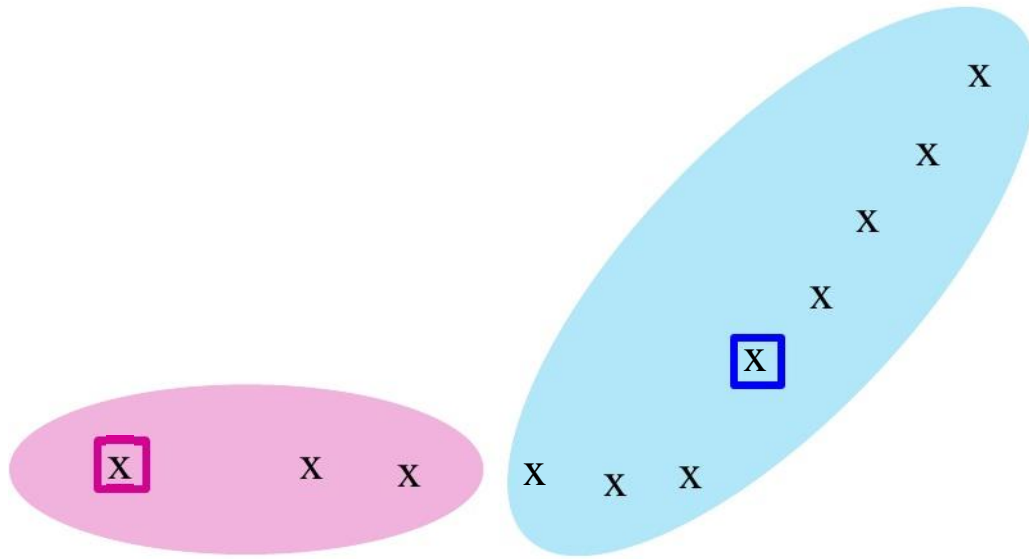
这是否意味着集群中的文档是不变的？

k-均值聚类示例

例子



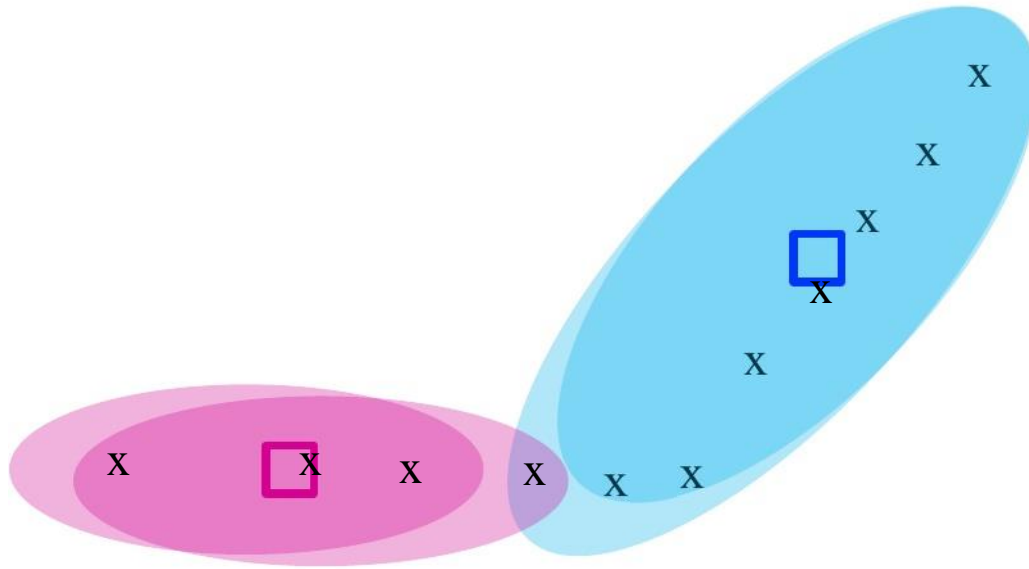
示例:分配集群



X, 数据点,
— 质心

第一轮之后的簇

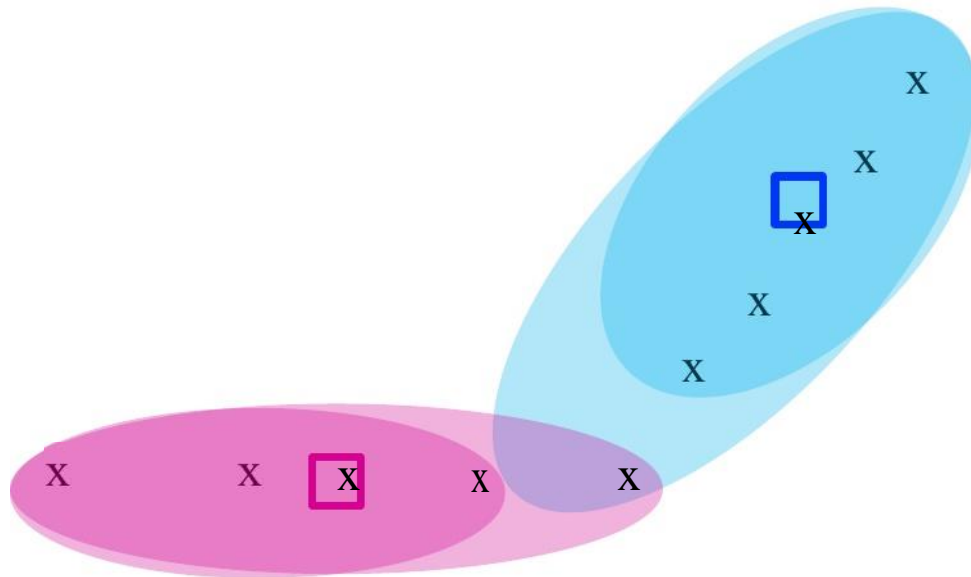
示例:分配集群



X, 数据点,
— 质心

2 轮后的簇

示例:分配集群



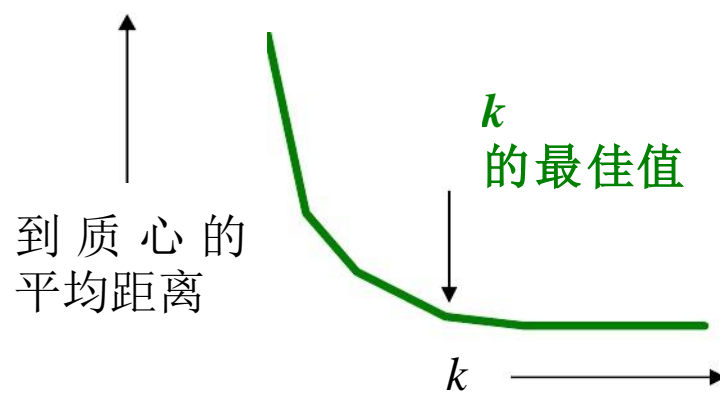
X, 数据点,
— 质心

最后的簇

正确计算k

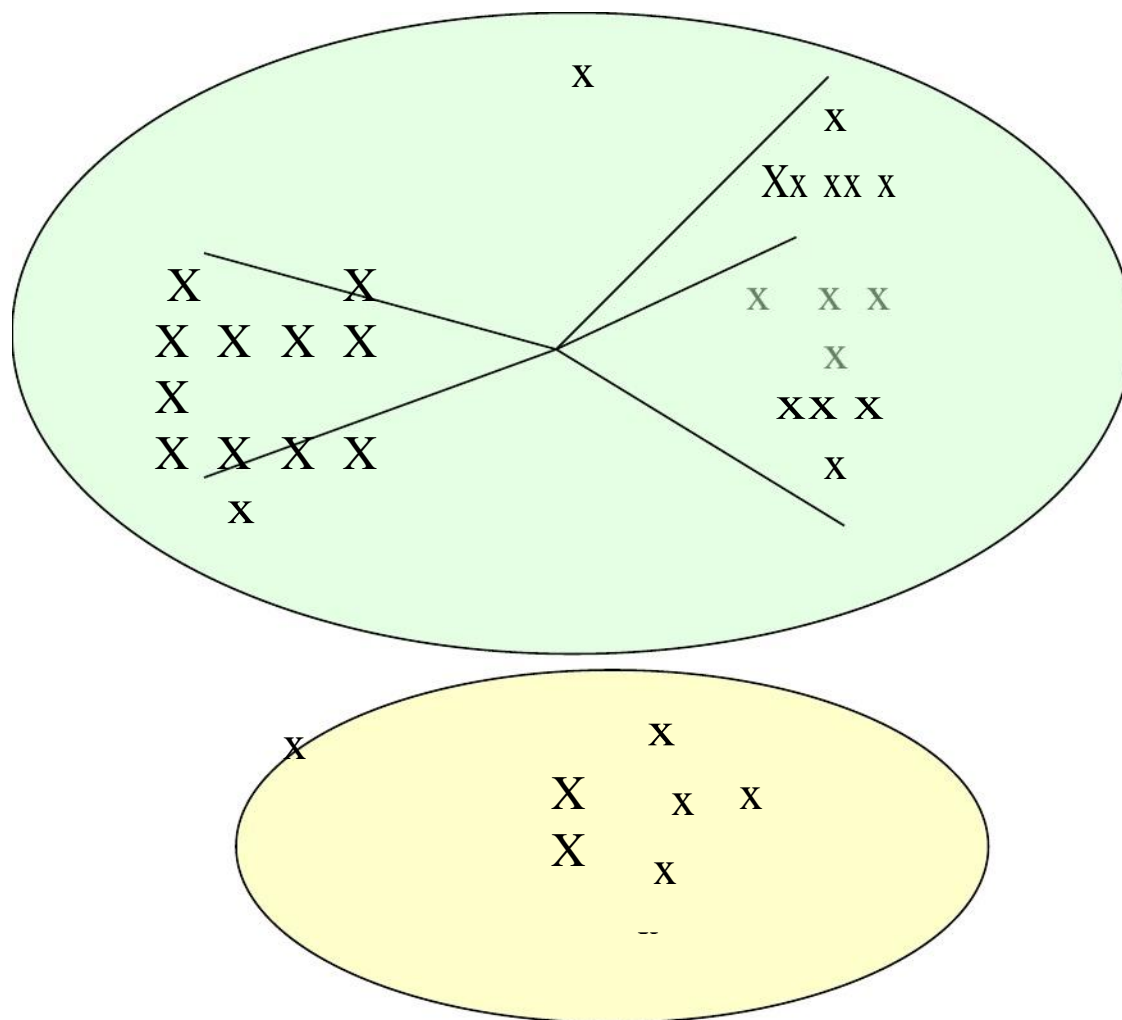
如何选择 k ?

‘尝试不同的 k ，观察随着 k 的增加，到质心的平均距离的变化’ average 迅速下降，直到右 k ，然后变化不大



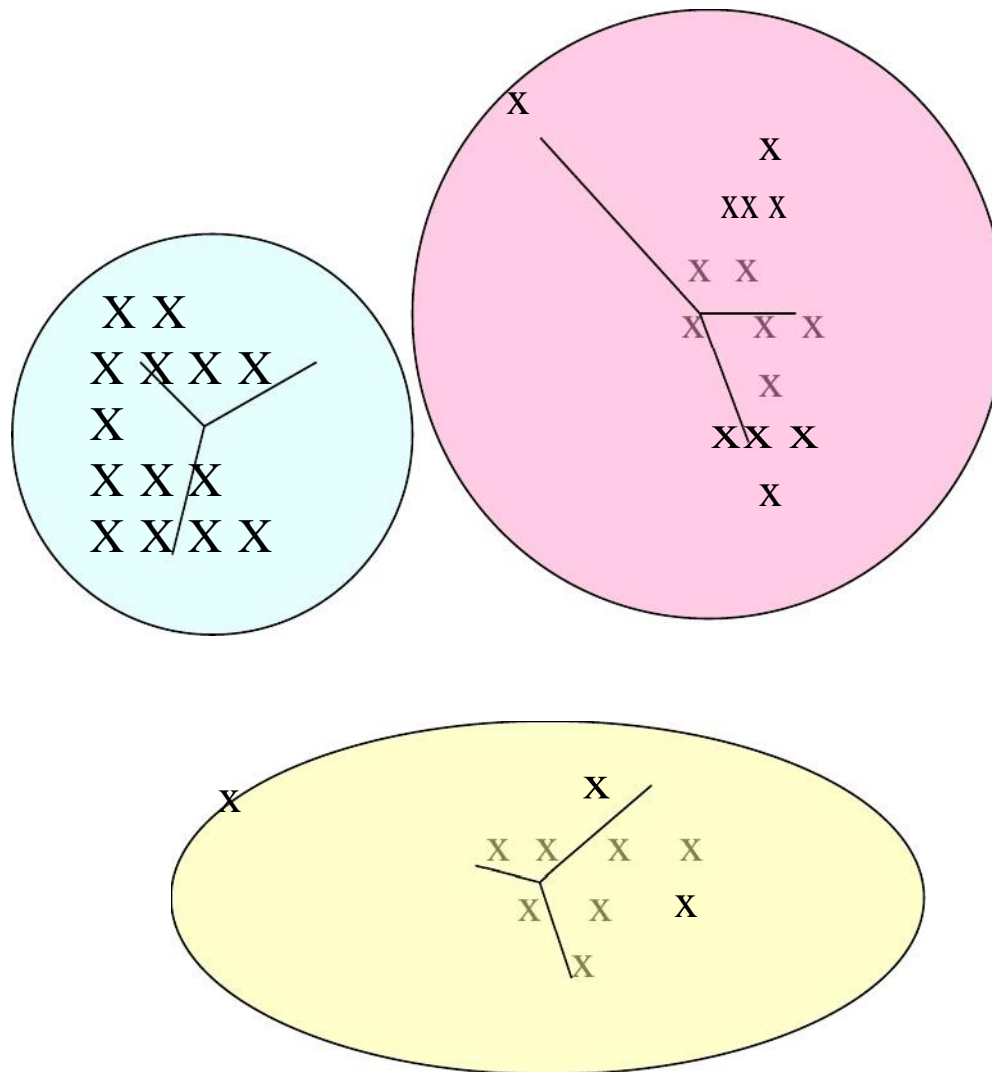
例子:选择k

太 少 ;
距 离
质 心 太



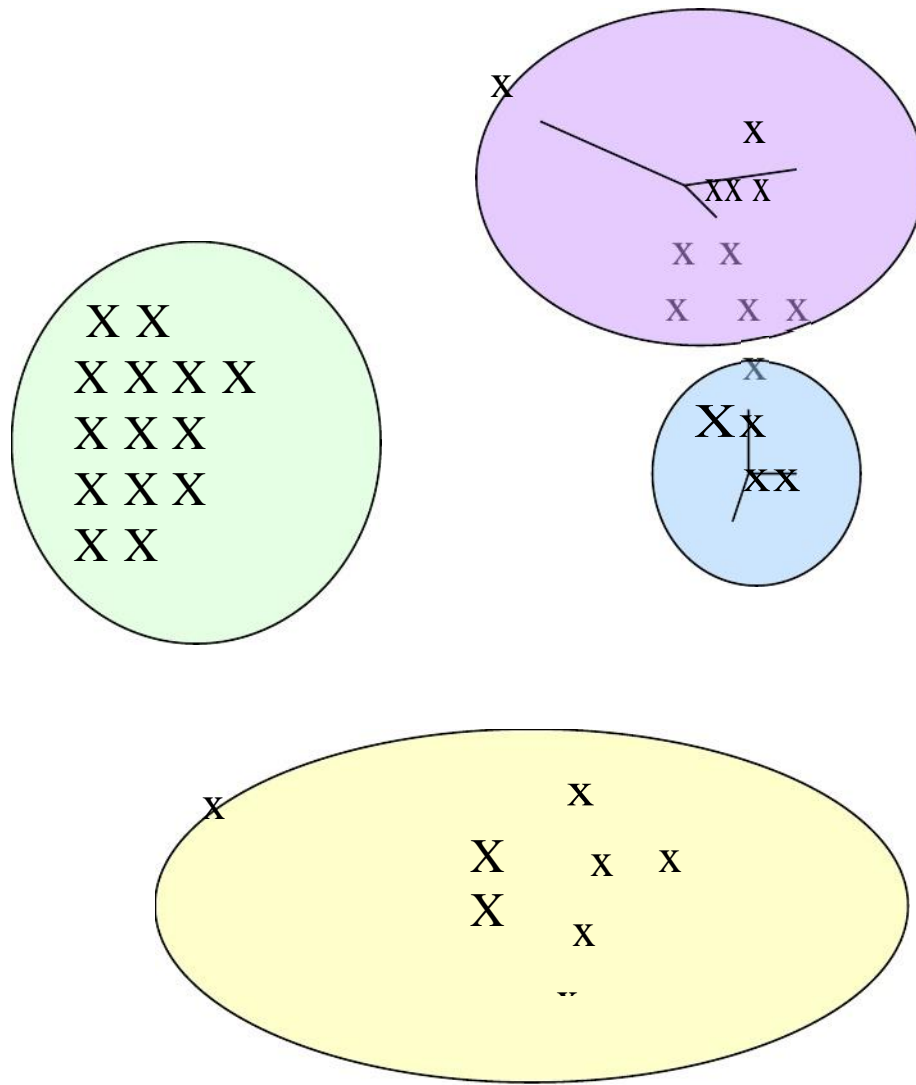
例子:选择k

刚刚好;距离
相当短。



例子:选择k

太多;平均距离几乎没有改善。



k-均值

的优势

§ 相对高效: $O(tkn)$, 其中 n 是#对象, k 是#聚类, t 是#迭代。通常情况下, $k, t \ll n$. § 往往终止于局部最优。使用诸如模拟退火和遗传算法之类的技术可以找到全局最优

弱点

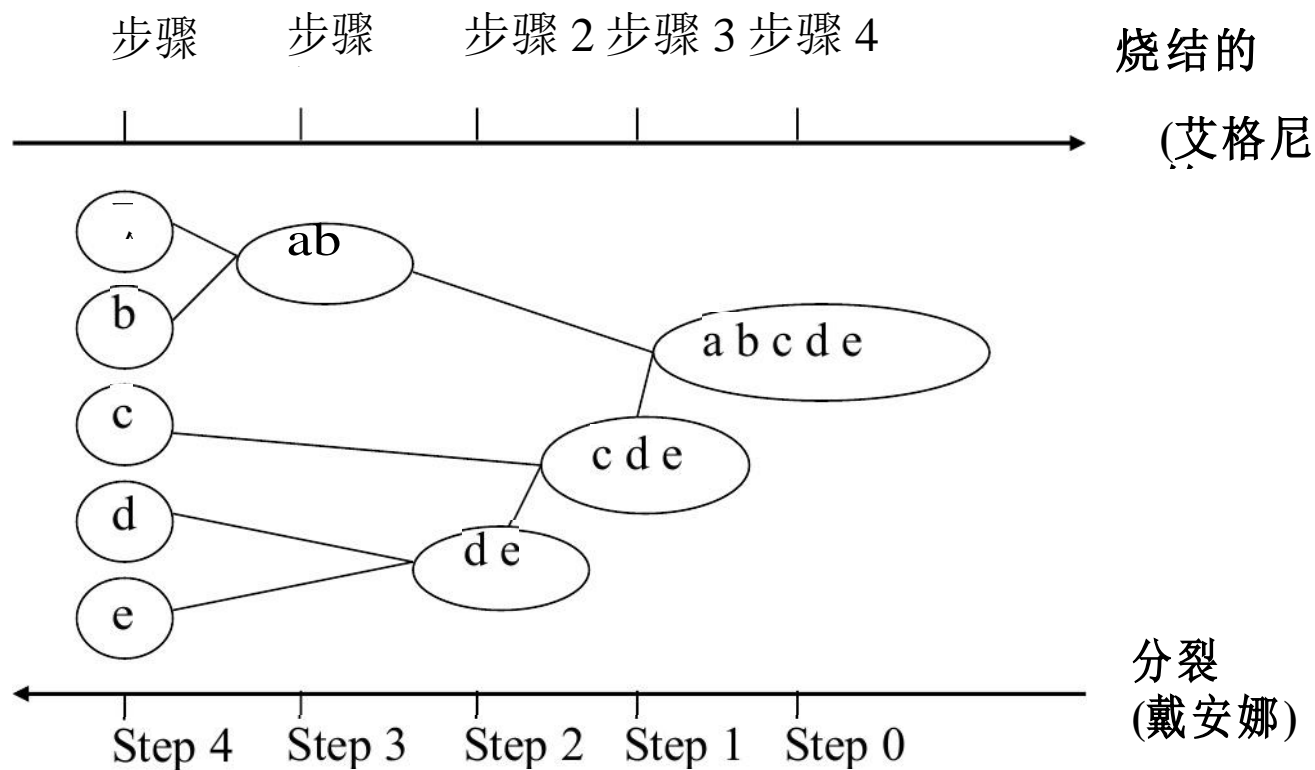
§ 仅适用于定义平均值时(分类数据呢?)

§ 需要预先指定 k , 聚类的数量 § 有噪声数据和异常值的麻烦

§ 不适合发现形状非凸的聚类

层次方法

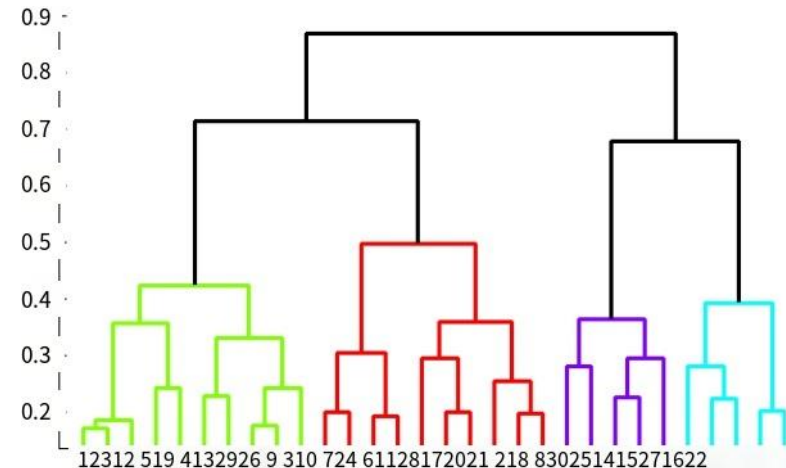
“使用距离矩阵作为聚类标准。这种方法不需要簇的个数 k 作为输入，但是需要一个终止条件



概述:聚类的方法

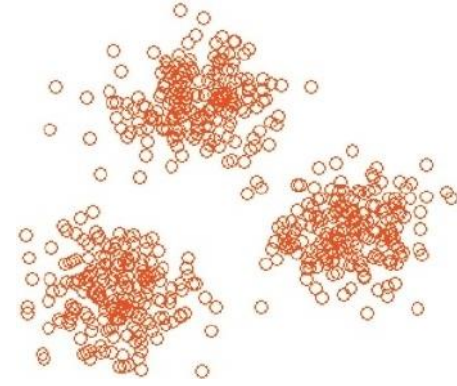
! 层次化: § 凝聚(bottom - up): §
最初, 每个点都是一个集群 § 将两者反复结合

“最近的” 集群成一个 §
分裂(自上而下):



§ 从一个集群开始, 递归地拆分它

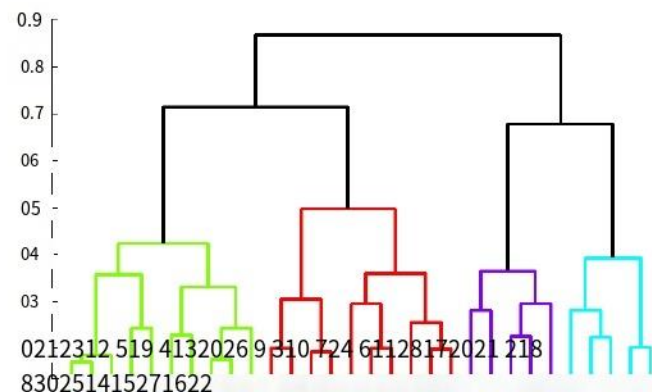
点分配: § 维护一组簇 § 点属于“最近”的簇



分层聚类

， 关键操作：

反 复 合 并
两个最近的簇



！ 3个重要问题：§ 1)如何表示包含一个以上点的簇？

§ 2)如何确定簇的“贴近度”？

§ 3)什么时候停止组合集群？

最接近的簇对

‘定义最接近的集群对的许多变体’ 单链接

§ 最余弦相似(单链接)的相似性 ‘完全链接’

§ 相似性的“最远”点，最小的余弦相似

! 重心

§ 其质心(重心)与余弦最相似的集群

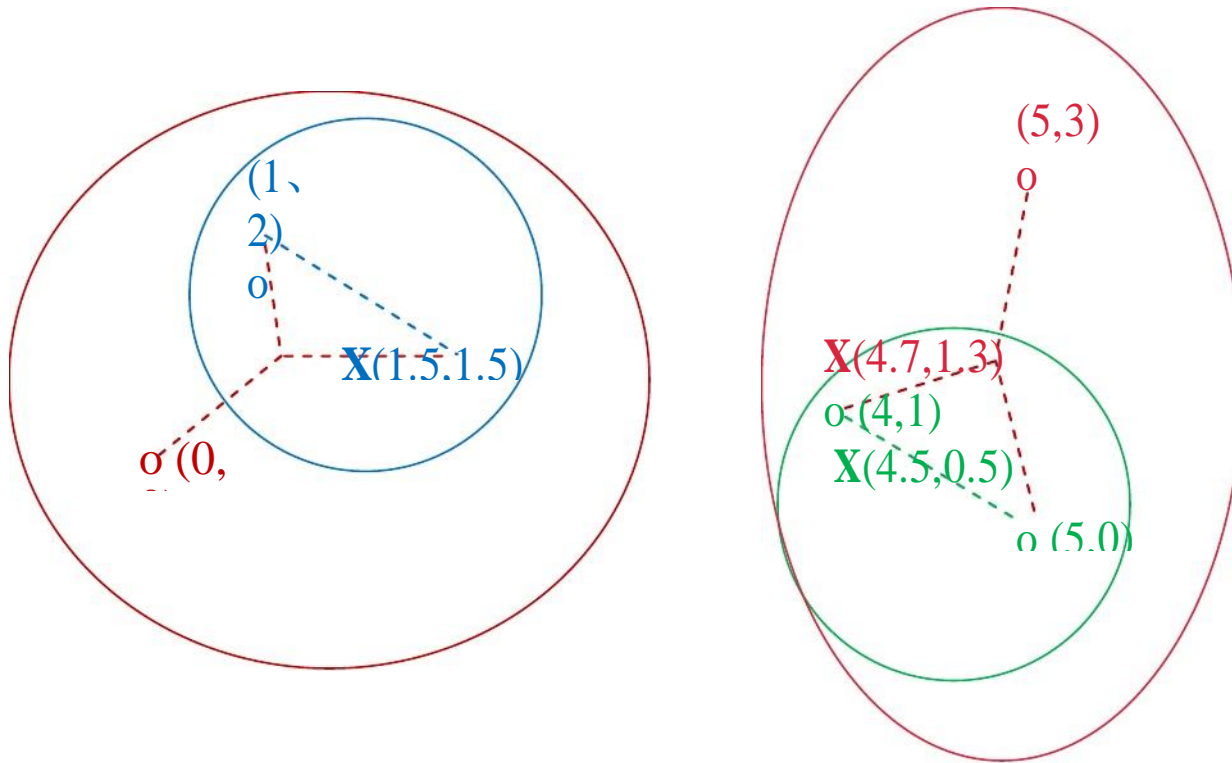
! **Average-link**

§ 元素对之间的平均余弦

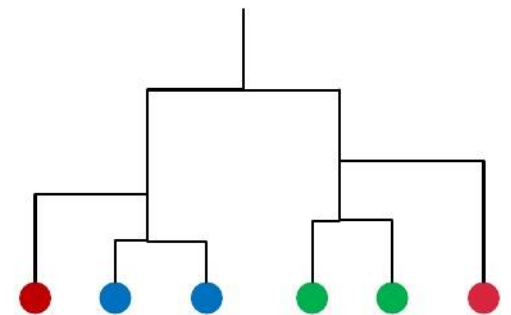
分层聚类

- ， **关键操作**:反复合并两个最近的簇
- ， **(1)如何表示由多个点组成的簇?** § **关键问题**:当你合并集群时，如何表示每个集群的“位置”，以区分哪一对集群是最近的?
- ， **欧氏情况**:每个簇都有一个**质心**=其(数据)点的平均值， **(2)如何确定簇的“接近度”?** § 通过质心的距离来衡量簇的距离

示例:分层集群



数据: o ...数据点
点 x ...质心



在非欧几里得的情况下呢？

那非欧几里得的情况呢？“我们唯一能谈论的“地点”是点本身

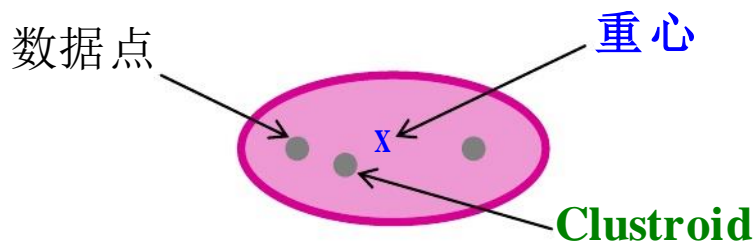
§ 也就是说，没有两个点的“平均值”

方法 1:

§ (1)如何表示由许多点组成的簇?*clustroid*=与其他点“最近”的(数据)点 § (2)如何确定簇的“接近度”?在计算簇间距离时，把 *clustroid* 当作质心来对待

“亲密”点？

(1) 如何表示由多个点组成的簇？“最近”的可能含义： δ 到其他点的最小最大距离 δ 到其他点的最小平均距离 δ 到其他点的距离的最小平方和 δ 对于距离度量 d ，簇 c 的 clusterid c 为： $\min_c d(x, c)$



对 3 个
数据点
进行聚
类

)²

x_{IC}

质心是集群中所有(数据)点的平均值。这意味着质心是一个“人工”点。

Clustroid是一个与簇中所有其他点“最近”的现有(数据)点。 51

定义集群的“接近度”

“(2)如何确定簇的“贴近度”？

§ 方法 2:

簇间距离=任意两点之间的最小距离，每个簇一个 §

方法 3:

选择一个簇的“凝聚”的概念，例如，从簇的最大距离 § 合并其联合最凝聚的簇

凝聚力

、接近 3.1: 使用合并的簇的直径=簇中点之间的最大距离

、接近 3.2: 使用集群中点之间的平均距离’ 接近 3.3: 使用基于密度的方法 § 取直径或平均距离，例如，除以集群中点的数量

实现

▼ Naïve 层次聚类的实现:

§ 在每一步，计算所有聚类对之间的成对距离，然后合并 § $O(N^3)$

，使用优先队列仔细实现可以将时间减少到 $O(N^2 \log N)$ § 对于内存中容纳不下的真正的大数据集来说，仍然太昂贵

其他分层聚类方法

- “凝聚聚类方法的主要缺点 § 不能很好地扩展:时间复杂度至少为 $O(n^2)$, 其中 n 是总对象的数量
 - § 永远不能撤销以前做过的事情
- ‘融合层次与基于距离的聚类 § BIRCH:使用 cf 树, 增量地调整子聚类的质量
 - § CURE:从簇中选择分散良好的点, 然后以指定的分数将它们缩小到簇的中心

方法论

“划分方法” “层次方法”
“基于模型的方法”
“基于密度的方法”

密度方法

基于密度的聚类(局部聚类准则)，例如密度连通的点

一项主要功能:

- § 发现任意形状的簇

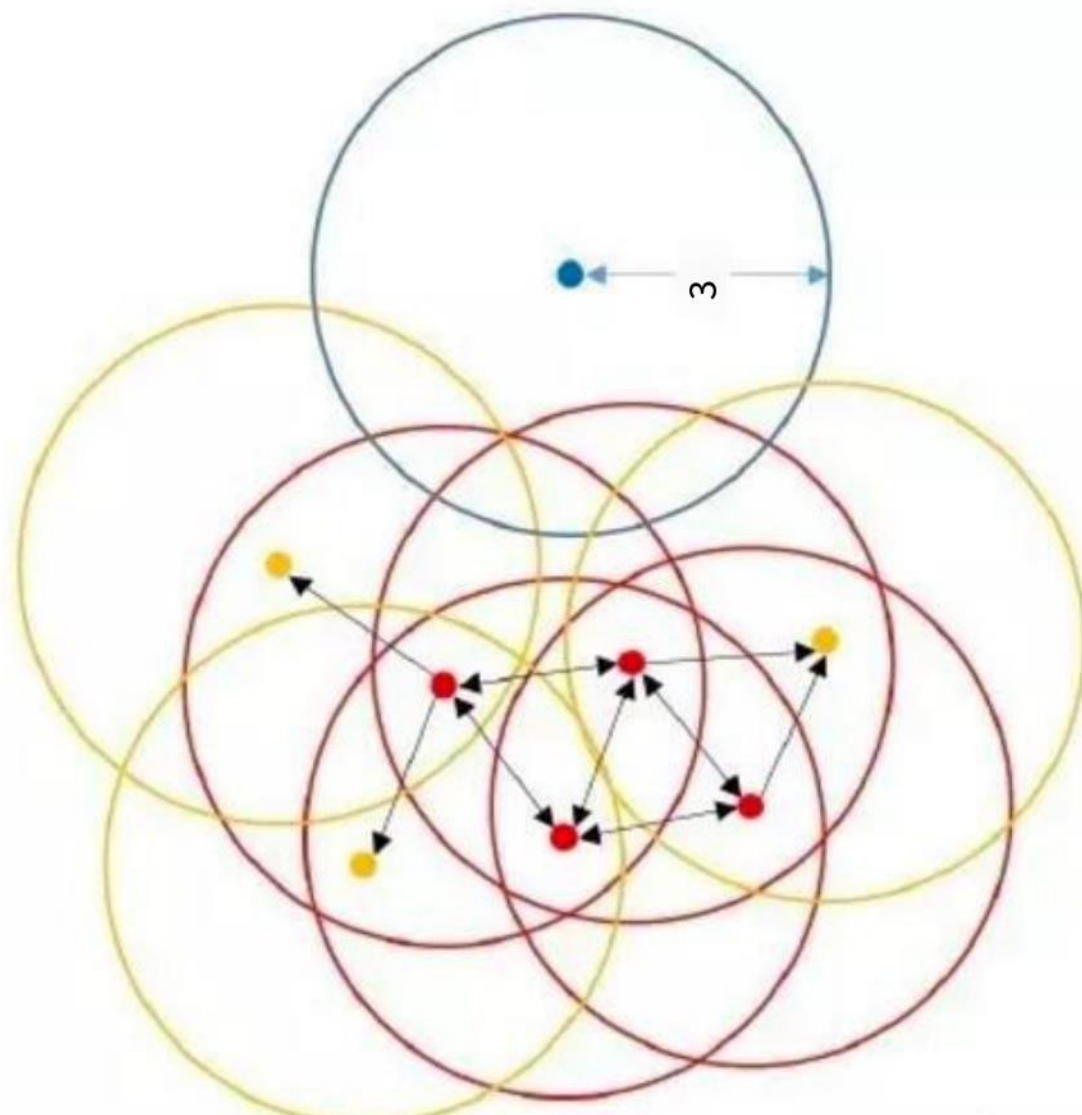
- § 处理噪音

- § 一次扫描

- § 需要密度参数作为终止条件。

- § DBSCAN (Ester et al., 1996)

- § DENCLUE (Hinneburg & Keim, 1998)



定义(1)

两个参数:

§ *Eps*: 邻居的最大半径

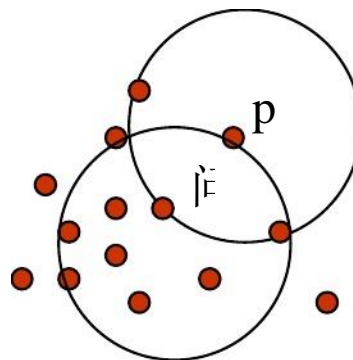
§ *MinPts*: 点的 *eps* 邻域内的最小点数

$N_{Eps}(p) = \{q \in D \mid dist(p, q) \leq Eps\}$

直接密度可达: 一个点 *p* 是直接密度-
从 *q* wrt 点可达。 *Eps*, *MinPts* iff

§ 1) *p* 属于 $N_{Eps}(q)$ § 2) *q* 是
一个核心点:

$$|N_{Eps}(q)| \geq MinPts$$



$MinPts = 5$

$Eps = 1 \text{ cm}$

定义(2)

■ Density-reachable:

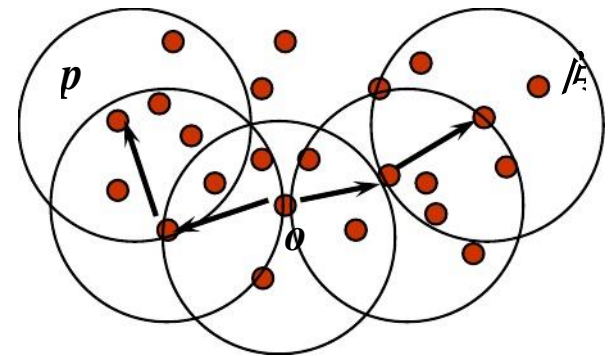
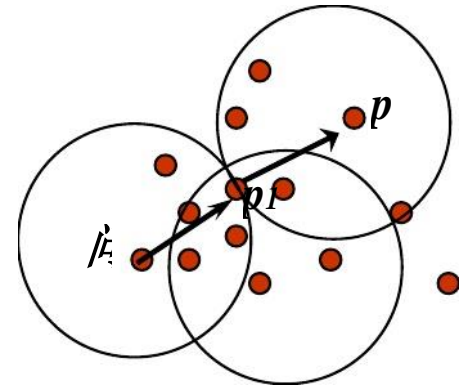
§ 从 q wrt 点出发, p 点是密度可达的。

$Eps, MinPts$ 如果有一个点链 $p_1, \dots, p_n, p_1 = q, p_n = p$ 使得 p_{i+1} 直接密度可达

Density-connected

§ 点 p 与点 q wrt 是密度相连的。 $Eps,$

■ $MinPts$, 如果有一个点 o 使得 p 和 q 从 o 到 t 都是密度可达的。 Eps 和 $MinPts$ 。

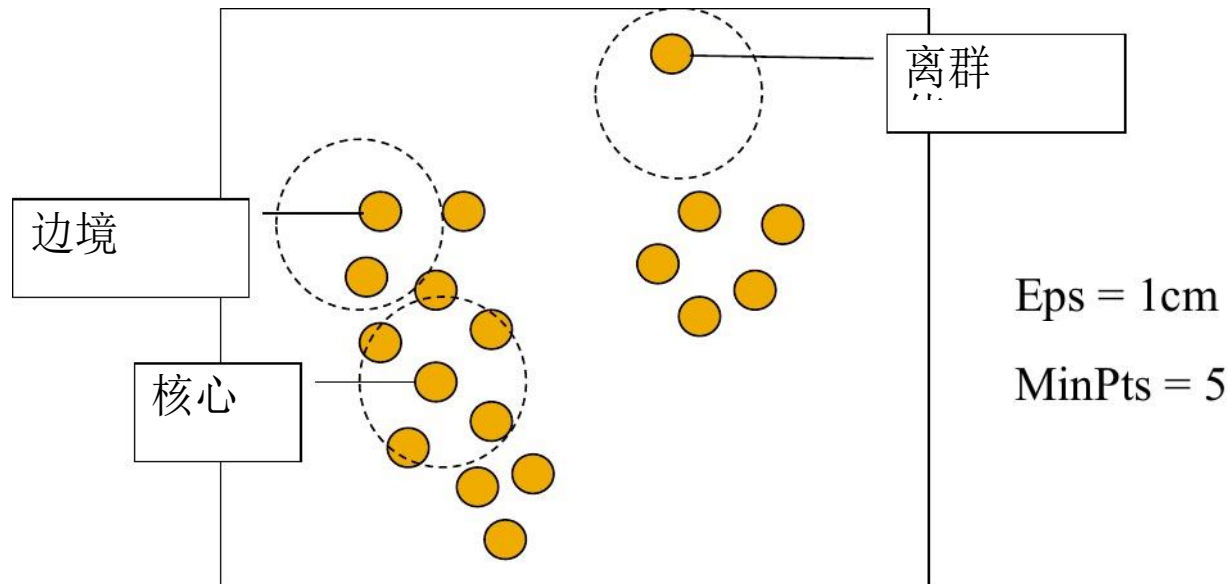


基于密度的空间聚类

含噪声的应用

*依赖于 *基于密度的簇* 的概念: 一个簇被定义为密度连通的点的最大集合

发现含有噪声的空间数据库中任意形状的簇



DBSCAN:算法

§ 任意选择一个点 p

§ 从 p wrt Eps 和 $MinPts$ 中检索所有密度可达的点。

§ 如果 p 是核心点，则形成集群。

§ 如果 p 是一个边界点，那么从 p 出发没有点是密度可达的，DBSCAN 会访问数据库的下一个点。

§ 继续这个过程，直到所有的点都被处理完。

算法: DBSCAN,一种基于密度的聚类算法

输入:

D:一个包含 n 个对象的数据集

ϵ :

MinPts: 领域密度阈值输出

: 基于密度的簇的集合方法

:

1. 标记所有对象为 unvisited;

2. 做

3. 随机选择一个 unvisited 对象 p ;

4. 标记 p 为 visited;

5. If p 的 ϵ -领域至少有 MinPts 个对象

6. 创建一个新簇 C , 并把 p 添加到 C ;

7. 令 N 为 p 的 ϵ -领域中的对象集合

8. For N 中每个点 p

9. 未访问的;

10. 拜访过;

11. If p' 的 ϵ -领域至少有 MinPts 个对象, 把这些对象添加到 N

12. 如果 p 还不是任何簇的成员, 把 p 添加到 C ;

13. 结束,

14. 输出 C ;

15. Else 标记 p 为噪声;

16. Until 没有标记为 unvisited 的对象;

评价聚类好坏

“内部标准:一个好的聚类将产生高质量的聚类,其中:

- § 类内(即簇内)相似度高

- § 类间相似度低

- § 聚类的质量取决于文档表示和使用的相似性度量

聚类质量的外部标准

- “质量是通过它发现金标准数据中部分或全部隐藏模式或潜在类别的能力来衡量的”
- ‘根据真实数据评估聚类……需要 *标记数据*’
- ‘假设文档具有 C 个黄金标准类，而我们的聚类算法产生 K 个聚 $_1$ 类， $\omega_2, \omega, \dots_K \omega$ 具有 n_i member。

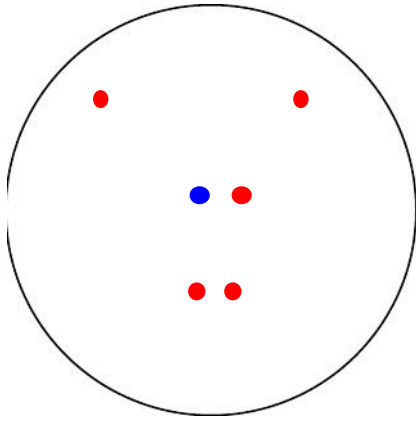
集群质量的外部评价

- 简单衡量: 纯度, 集群中优势类 π 与集群 i 大小 ω 之间的比例

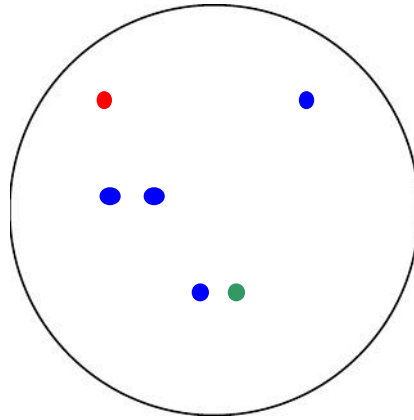
$$\text{纯度}(w_i) = \frac{1}{\omega_i} \max_j \pi_{ij}$$

- 偏倚是因为拥有 n 个簇可以最大化纯度
- “其他是簇中类的熵(或类和簇之间的互信息)

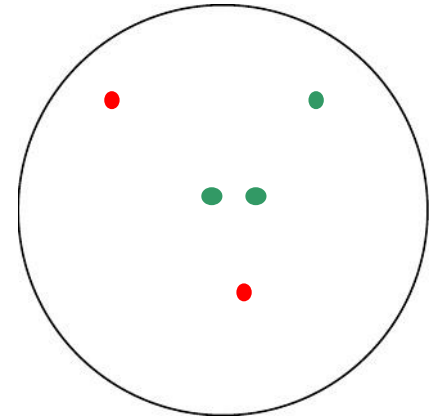
纯洁的例子



集群我



集群二世



集群三世

簇 I: 纯度 = $1/6 (\max(5, 1, 0)) = 5/6$

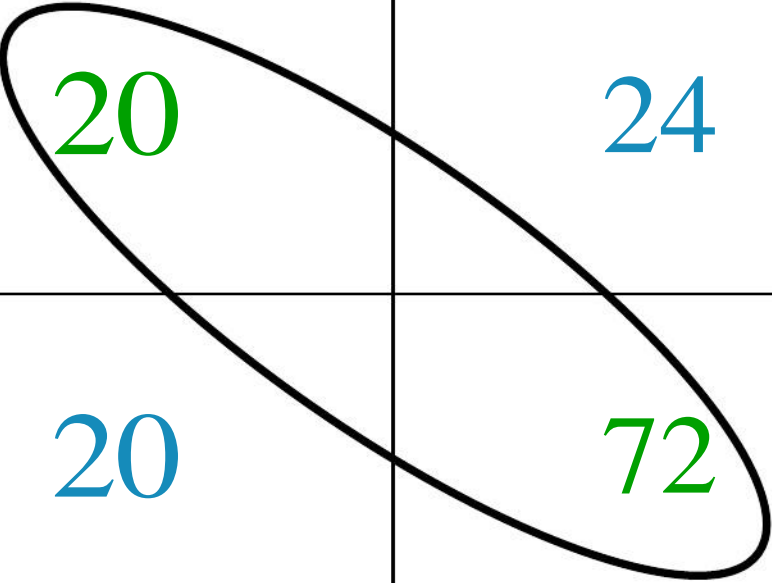
聚类 II: 纯度 = $1/6 (\max(1, 4, 1)) = 4/6$ 聚类 III:

纯度 = $1/5 (\max(2, 0, 3)) = 3/5$

兰特指数衡量货币对之间的决定。这里

$RI = 0.68$

点数	聚类中的同 一	聚类中的不 同
同一类在 基本真理	20	24
不同的类别在 基本真理	20	72



Rand指数与聚类f测度

$$RI = \frac{A + D}{A + B + C + D}$$

与标准查准率和查全率相比:

$$P = \frac{A}{A + B}$$

$$R = \frac{A}{A + C}$$

人们还定义和使用聚类 F- measure，这可能是一个更好的度量。

最后的话……

在聚类中，聚类是在没有人工输入的情况下从数据中推断出来的(无监督学习)

然而，在实践中，它有点不太清楚：有很多方法影响聚类的结果：聚类的数量，相似性度量，文档的表示，……

