无监督学习——聚类

郭嘉丰

中国科学院大学

中国科学院计算技术研究所

大纲

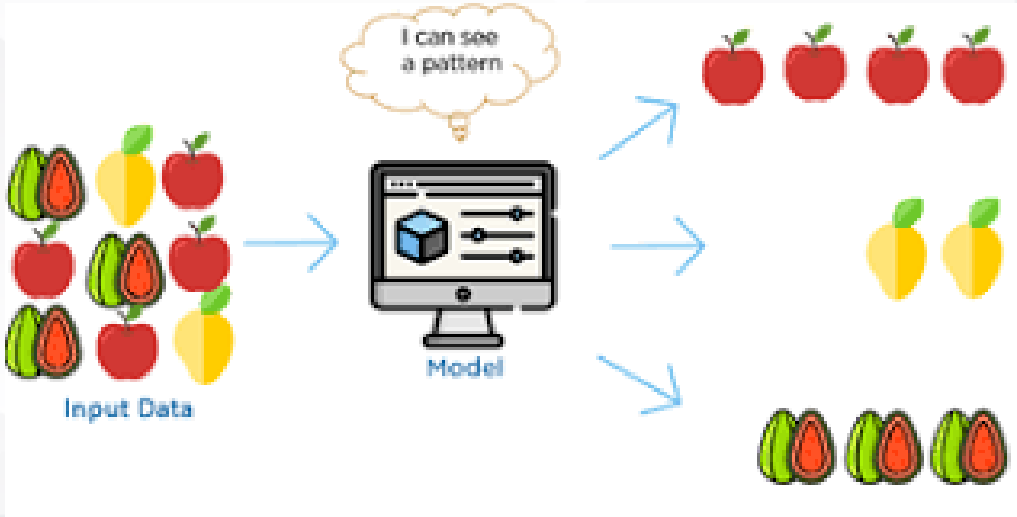
- 简介
- 距离计算
- 聚类算法
  - K均值聚类
  - 高斯混合模型和EM 算法
  - 层次聚类
  - 基于密度聚类

有监督 vs. 无监督

- 有监督学习: 给定 $\{x^i, y^i\}_{i=1}^{\mathrm{N}}$, 学习 $\hat{y} = f(x; w)$
  - 分类: y 是类别标签
  - 回归: y 是连续值
  - 排序: y 是序值(ordinal)
- 无监督学习: 给定 $\{x^i\}_{i=1}^{N}$, 学习 $\hat{y} = f(x; w)$
  - 密度估计: y 是密度
  - 聚类: y 是类簇
  - 维度约减/可视化: y是x的低维表示

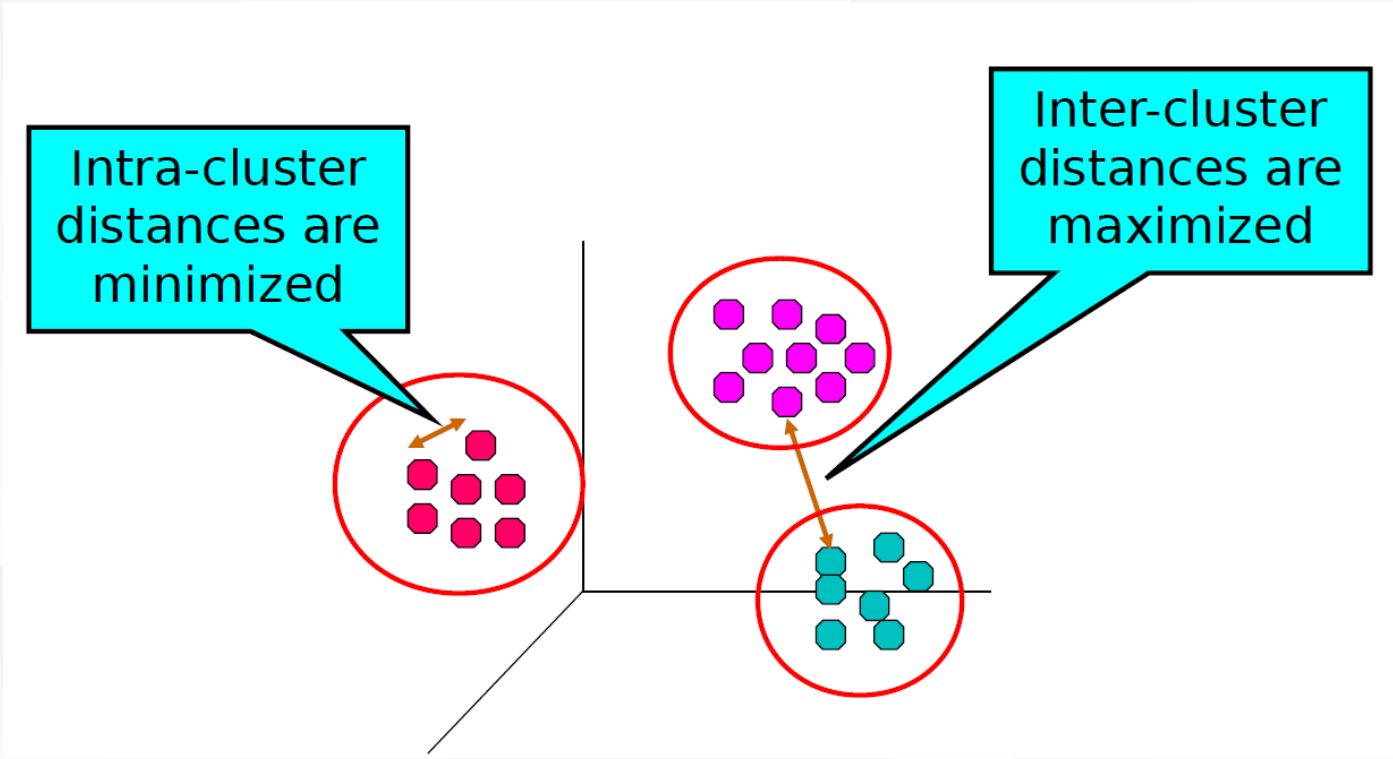
为什么我们需要无监督学习

- 原始数据易获得. 标注数据很难获取
- 节约内存和计算资源
- 减少高维数据中的噪声
- 有助于可解释的数据分析
- 经常作为监督学习的预处理步骤



聚类分析

- 寻找样本中的簇，使得同一簇内样本相似，不同簇之间样本不相似.



簇的概念并不明确

How many clusters?

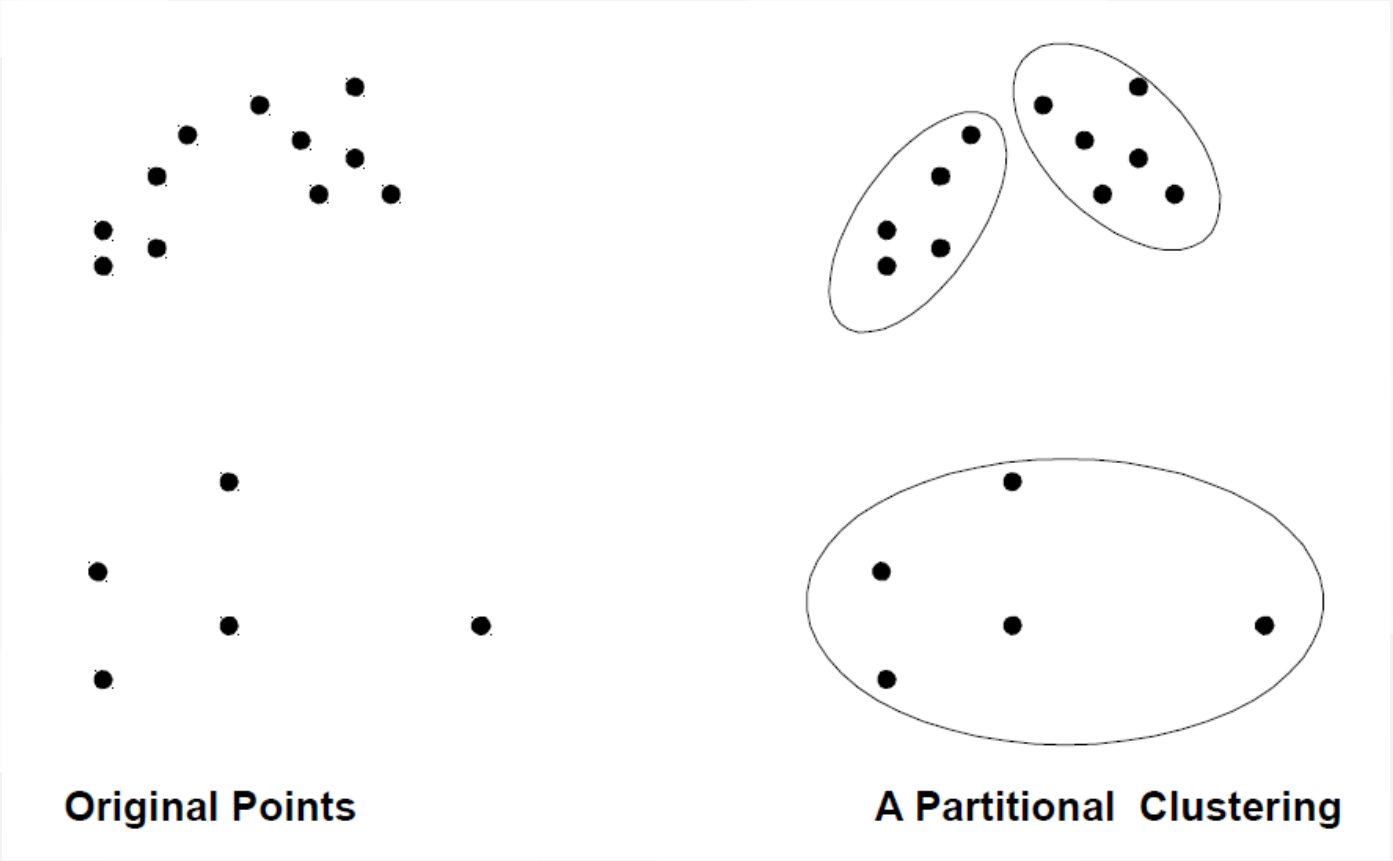
聚类的类型

- 聚类的结果是产生一个簇的集合

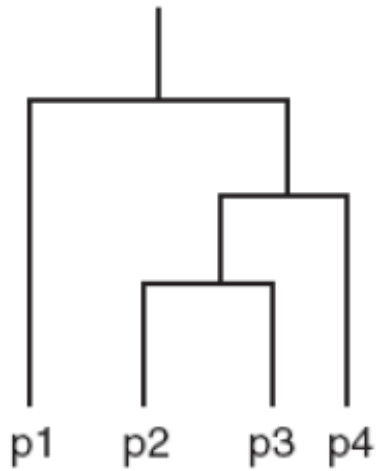
- 基于划分的聚类(无嵌套)
  - 将所有样本划分到若干不重叠的子集（簇），且使得每个样本仅属于一个子集

- 层次聚类(嵌套)
  - 树形聚类结构，在不同层次对数据集进行划分，簇之间存在嵌套

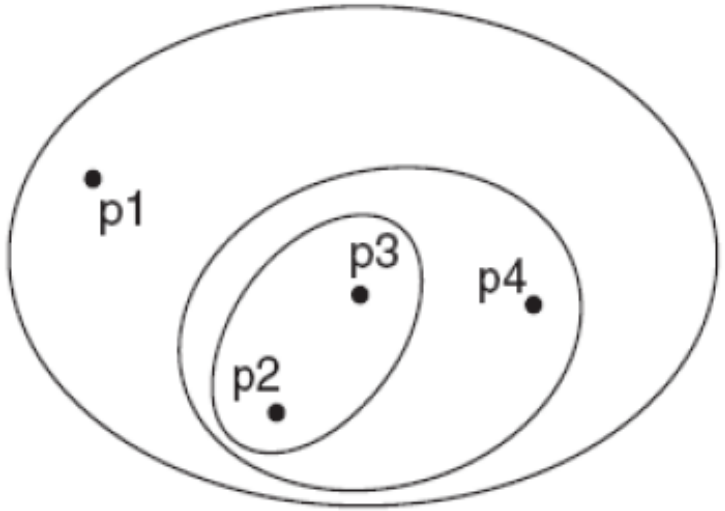
基于划分的聚类



层次聚类



(a) Dendrogram.



(b) Nested cluster diagram.

对聚类中簇集合的其他区别

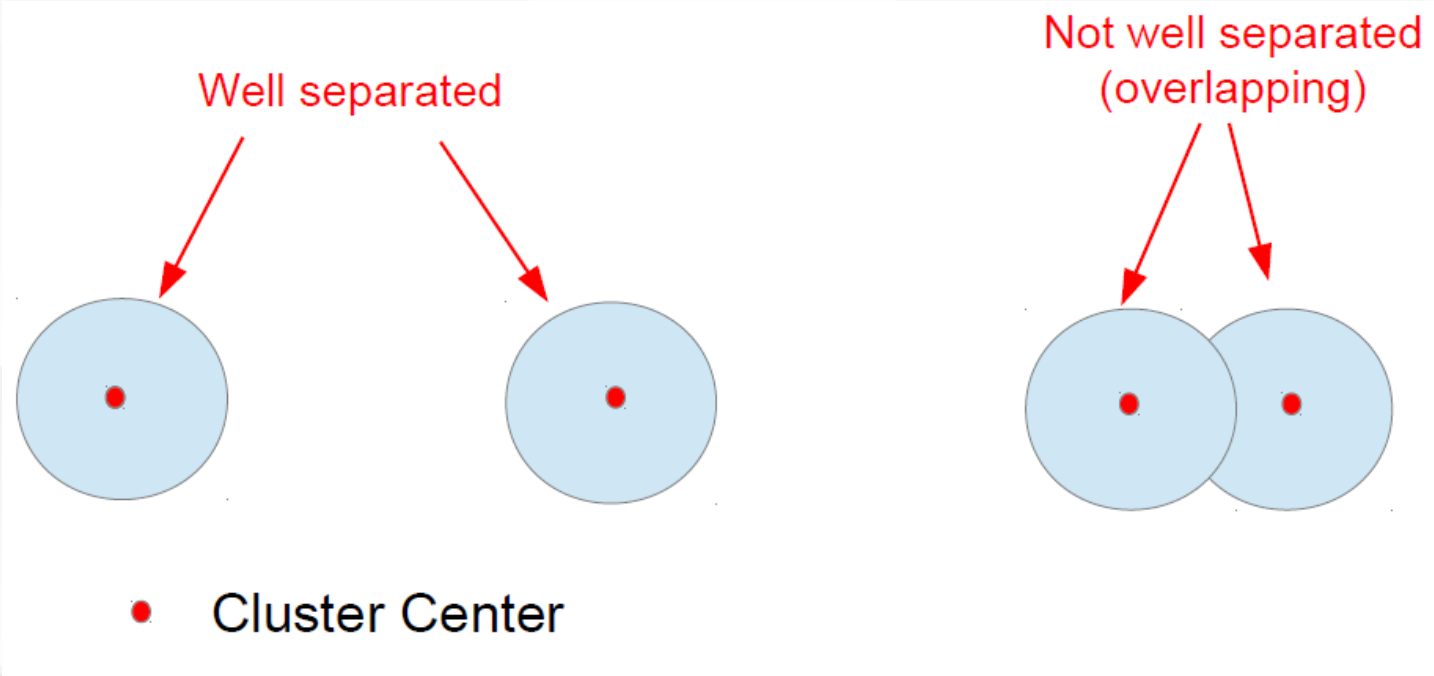
- **独占（Exclusive） vs. 非独占的（non-exclusive）**
  - 在非独占的类簇中, 样本点可以属于多个簇
- **模糊（Fuzzy） vs. 非模糊的（non-fuzzy）**
  - 在模糊聚类中, 一个样本点以一定权重属于各个聚类簇
  - 权重和为1
  - 概率聚类有相似的特性
- **部分（Partial） vs. 完备（ complete ）**
  - 在一些场景, 我们只聚类部分数据
- **异质（Heterogeneous） vs. 同质（homogeneous）**
  - 簇的大小、形状和密度的是否有很大的差别

簇的类型

- 基于中心的簇
- 基于邻接的簇
- 基于密度的簇
- 基于概念的簇

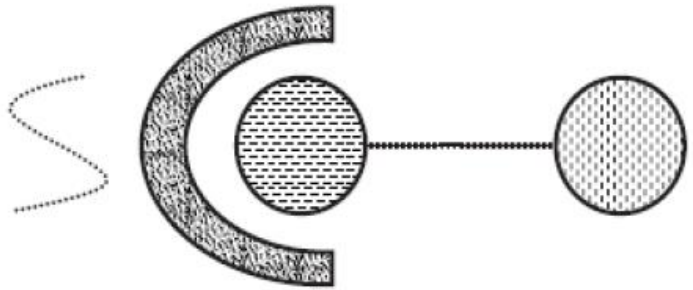
基于中心的簇

- 簇内的点和其“中心”较为相近（或相似），和其他簇的“中心”较远，这样的一组样本形成的簇
- 簇的中心常用质心(**metroid**)表示,即簇内所有点的平均, 或者用中心点(**medoid**)**表示**, 即簇内最有代表性的点



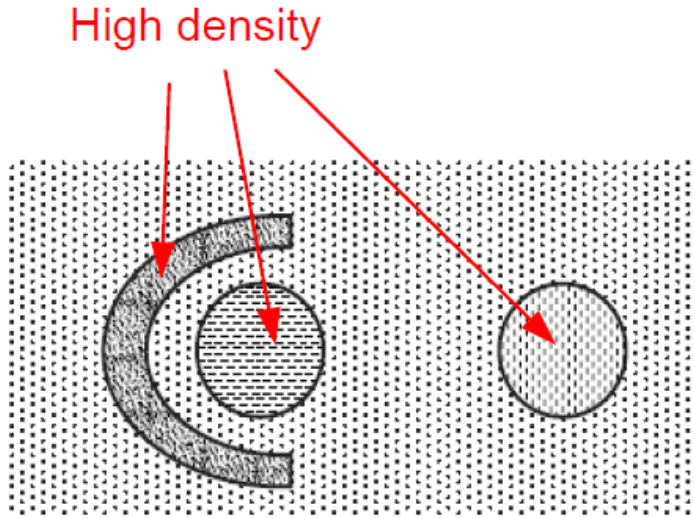
基于连续性和基于密度的簇

- 基于连续性的簇：相比其他任何簇的点，每个点都至少和所属簇的某一个点更近



(c) Contiguity-based clusters. Each point is closer to at least one point in its cluster than to any point in another cluster.

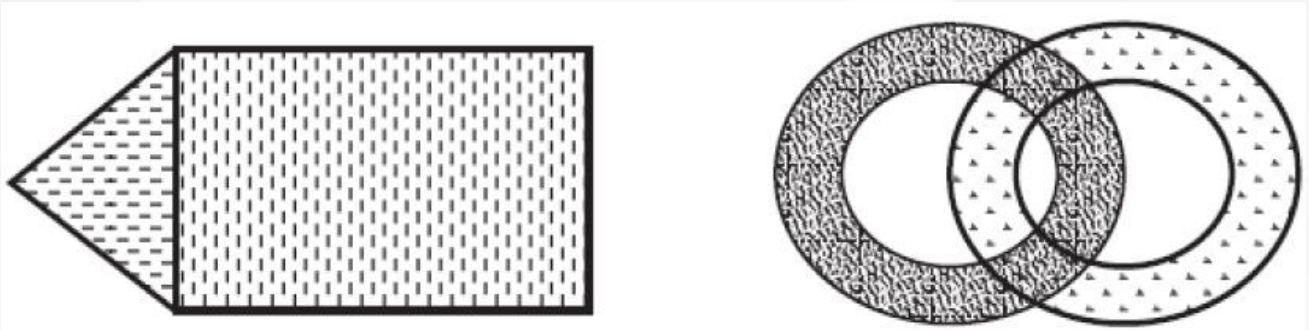
- 基于密度的簇：簇是有高密的区域形成的，簇之间是一些低密度的区域



(d) Density-based clusters. Clusters are regions of high density separated by regions of low density.

基于概念的簇

- 基于概念的簇：同一个簇共享某种性质，这个性质是从整个结合推导出来的



(e) Conceptual clusters. Points in a cluster share some general property that derives from the entire set of points. (Points in the intersection of the circles belong to both.)

- 基于概念的簇难检测，它通常不是：
  - 基于中心
  - 基于关系
  - 基于密度

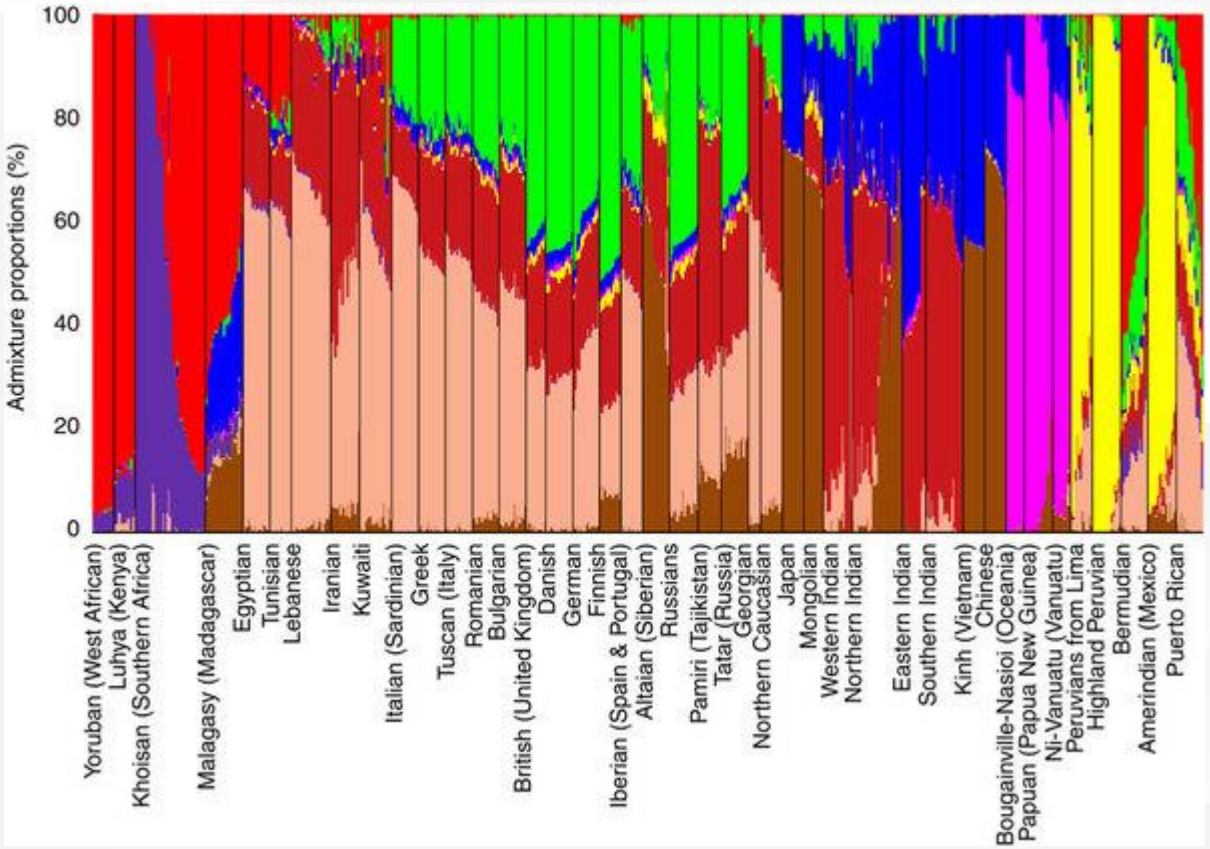
聚类的应用

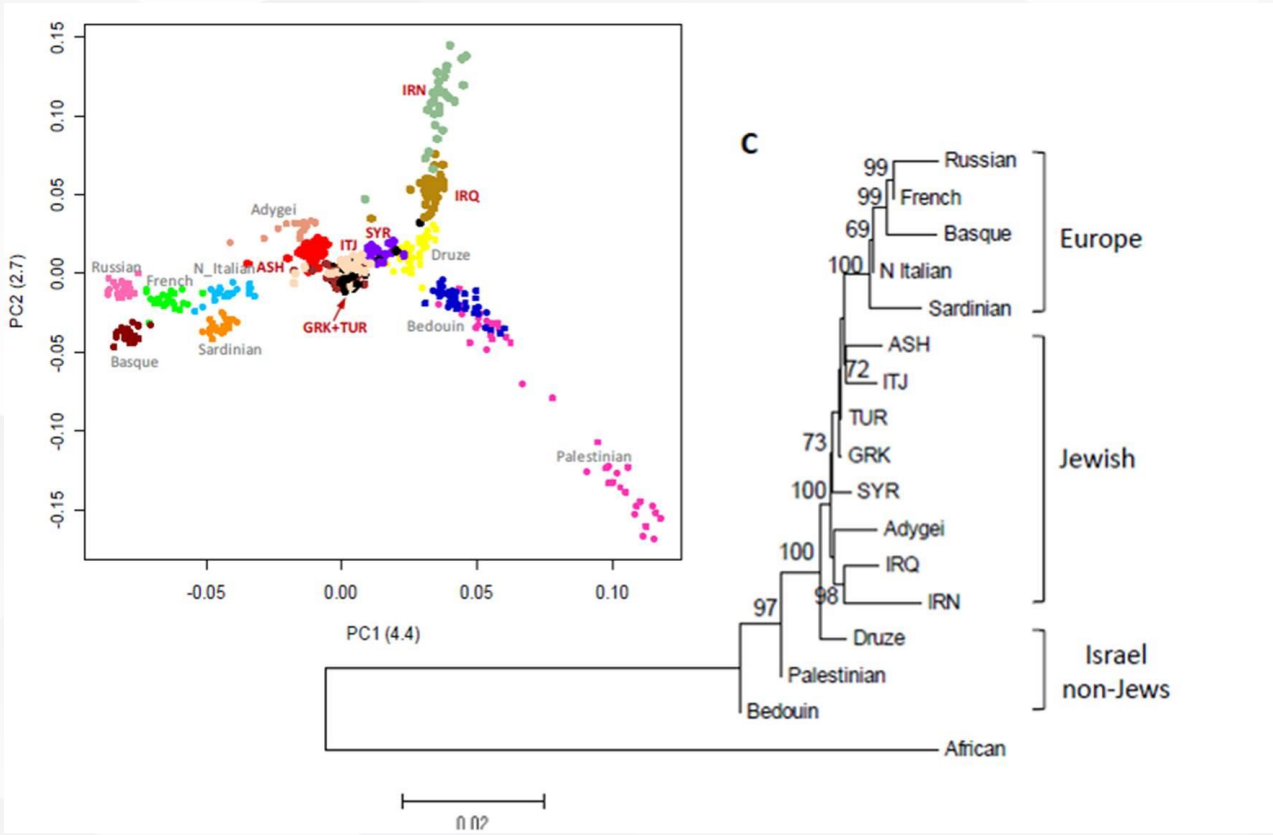
图像分割

http://people.cs.uchicago.edu/ pff/segment

聚类的应用

人类的种族分析

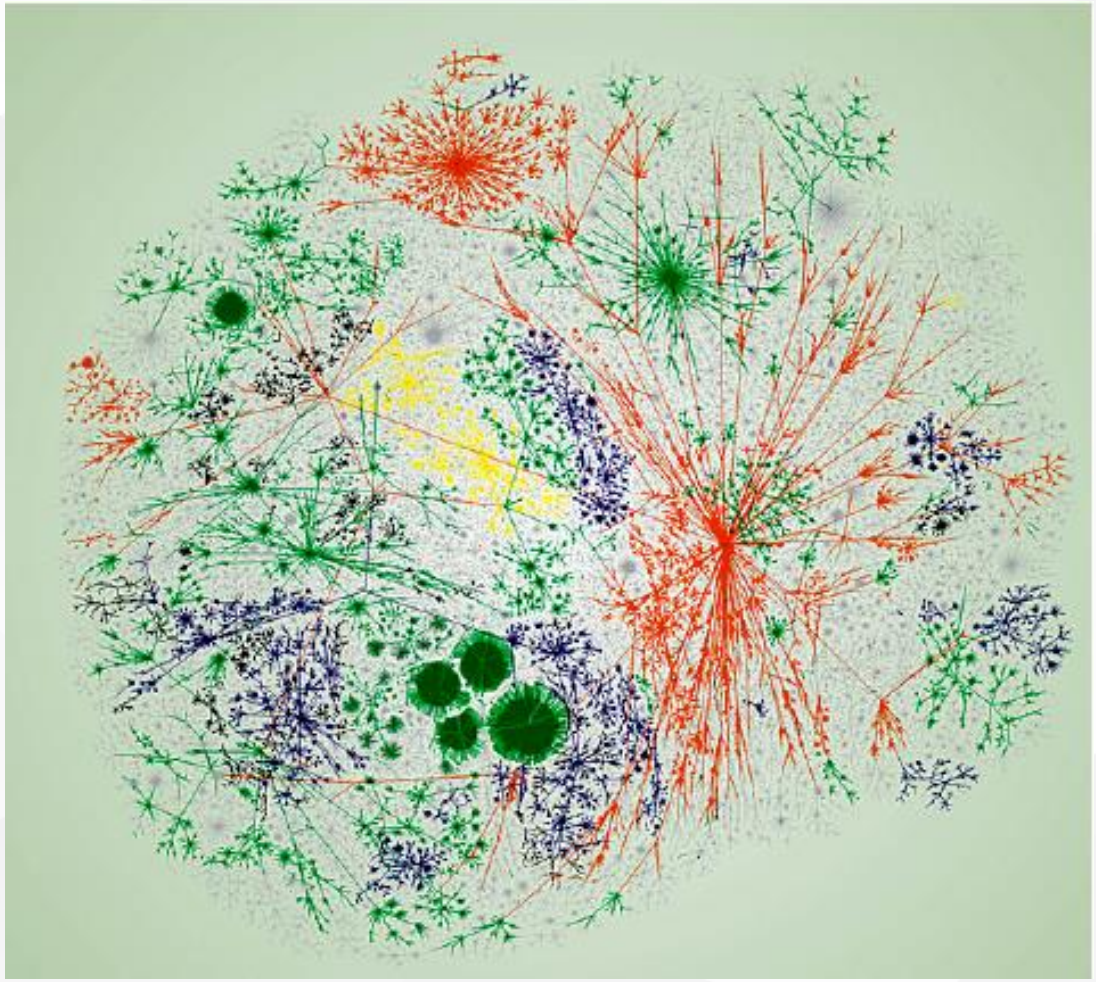




Eran Elhaik et al. Nature

聚类的应用

复杂网络分析

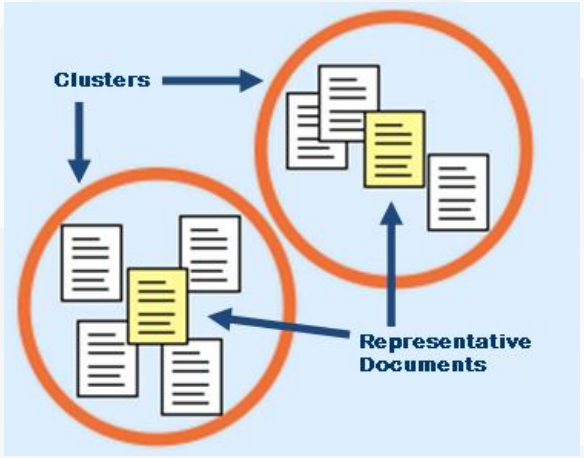


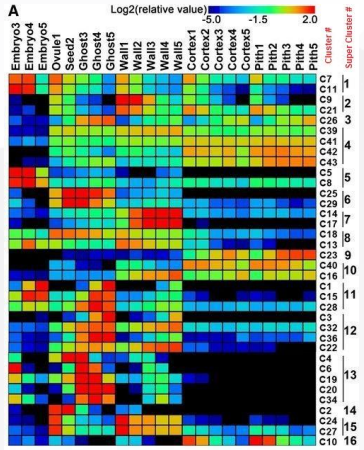
Newman, 2008

其他应用

- 用户画像：基于顾客消费历史对顾客聚类
- 商品分析：基于购买的用户对商品聚类
- 文本分析：基于相似的词对文档聚类
- 计算生物学：基于编辑距离对DNA序列聚类

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	0	3	0	3	0
User 2	4	0	0	2	0
User 3	0	0	3	0	0
User 4	3	0	4	0	3
User 5	4	3	0	4	0





聚类分析的“三要素”

- 如何定义样本点之间的“远近”
  - 使用相似性/距离函数
- 如何评价聚类出来簇的质量?
  - 利用评价函数去评估聚类结果
- 如何获得聚类的簇?
  - 如何表示簇，如何设计划分和优化的算法，算法何时停止

大纲

- 简介
- 距离函数
- 评价指标
- 聚类算法
  - K均值聚类
  - 高斯混合模型和EM 算法
  - 层次聚类
  - 基于密度的聚类

聚类分析的要素之一

- 如何衡量样本之间的“远近”？
  - 文档聚类时，我们如何衡量文档间远近？
  - 图像分割时，我们如何衡量像素点之间的远近？
  - 用户画像时，我们如何衡量用户之间的远近？

我们需要量化这些样本，并计算它们之间的距离

- 距离（Distance）/相似（Similarity）/不相似（Dissimilarity）/邻近（Proximity）函数的选择与应用相关
- 需要考虑特征的类型
  - 类别，序值，数值
- 可以从数据直接学习相似/距离函数

距离函数

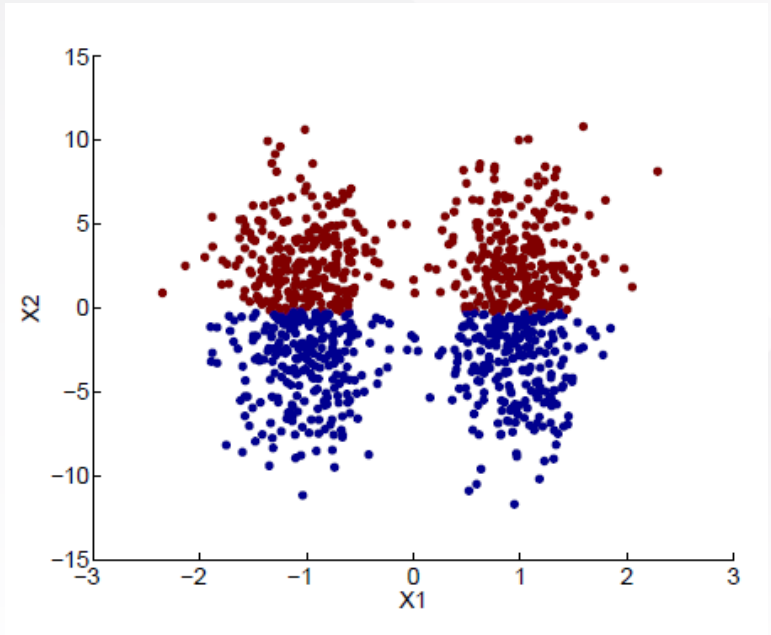
什么样的函数可以作为距离度量函数？

- 函数dist() 是一种距离度量当且仅当：
  - $dist(x_i, x_j) \geq 0$ (非负性)
  - $dist(x_i, x_j) = 0 \; iff \; x_i = x_j$ (同一性)
  - $dist(x_i, x_j) = dist(x_j, x_i)$ (对称性)
  - $dist(x_i, x_j) \leq dist(x_i, x_k) + dist(x_k, x_j)$ (直递性)
- Minkowski 距离:$dist_{mk} = (\sum_{u=1}^{n} |x_{iu} - x_{ju}|^p)^{\frac{1}{p}}$
  - p=2 Euclidean 距离: $dist_{mk} \overset{\text{def}}{=} dist_{ed}$
  - p=1 Manhattan 距离: $dist_{mk} \overset{\text{def}}{=} dist_{man}$
- 这类距离函数对特征的旋转和平移变换不敏感，对数值尺度敏感
- 如果样本特征值尺度不一致，将数据标准化

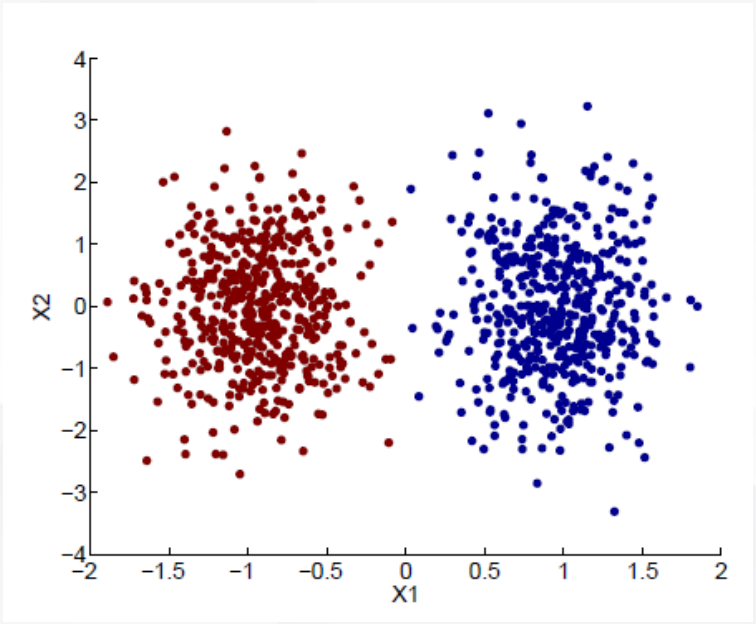
标准化

- $Y = \{X_1, X_2, \ldots, X_n\}$ 为d维原始数据, $Y \in R^{n \times d}$
- Z-score: $x_{ij} = Z(x_{ij}) = \frac{x_{ij} - \bar{x}_j}{\sigma_j}$, $\bar{x}_j$和σ_j是第j个特征的均值和标准差
- Z-score转换后的特征值得均值为0，方差为1
- 其他标准化算法还有Min-max, Decimal scaling等

标准化

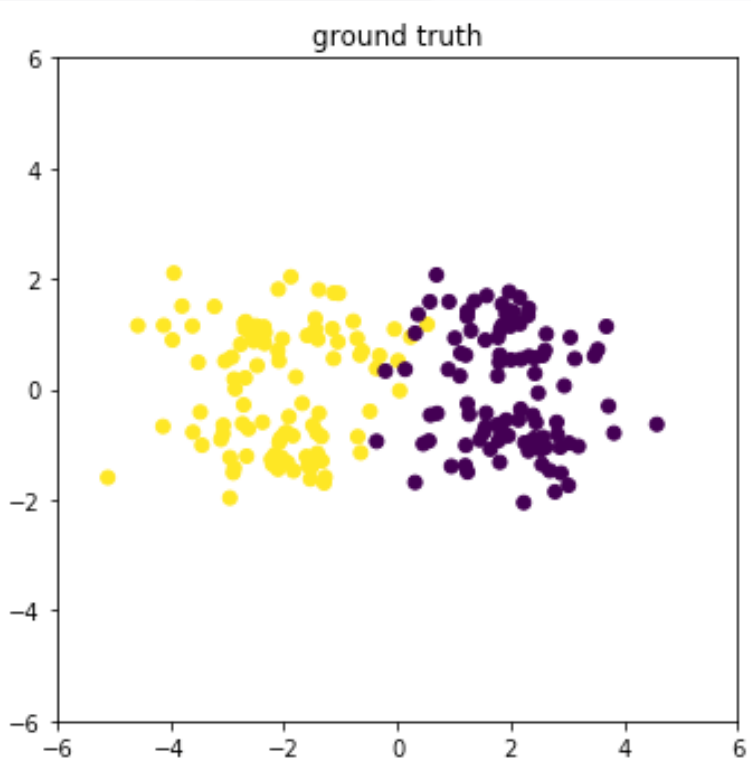


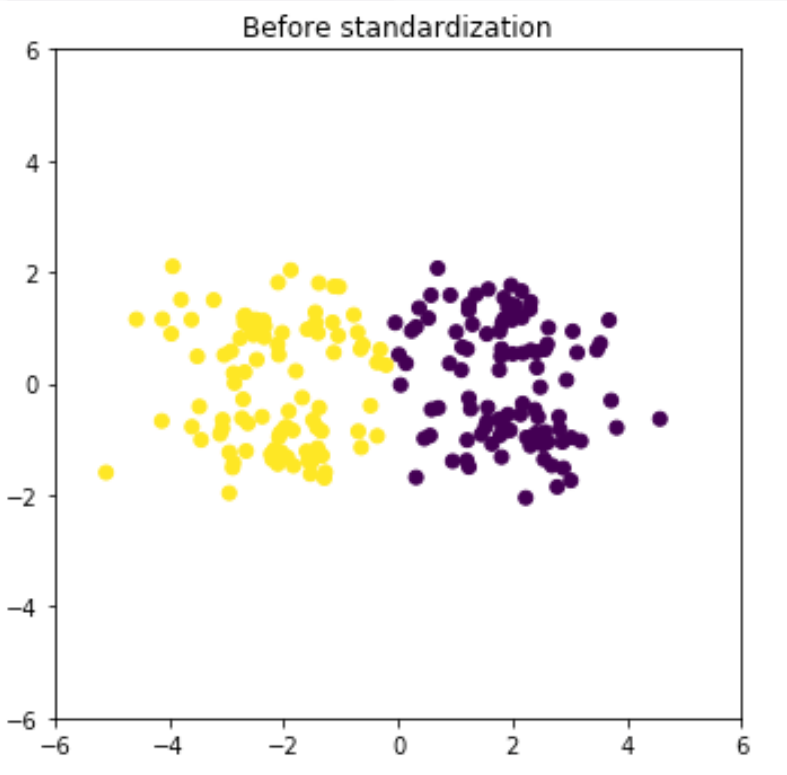
无标准化

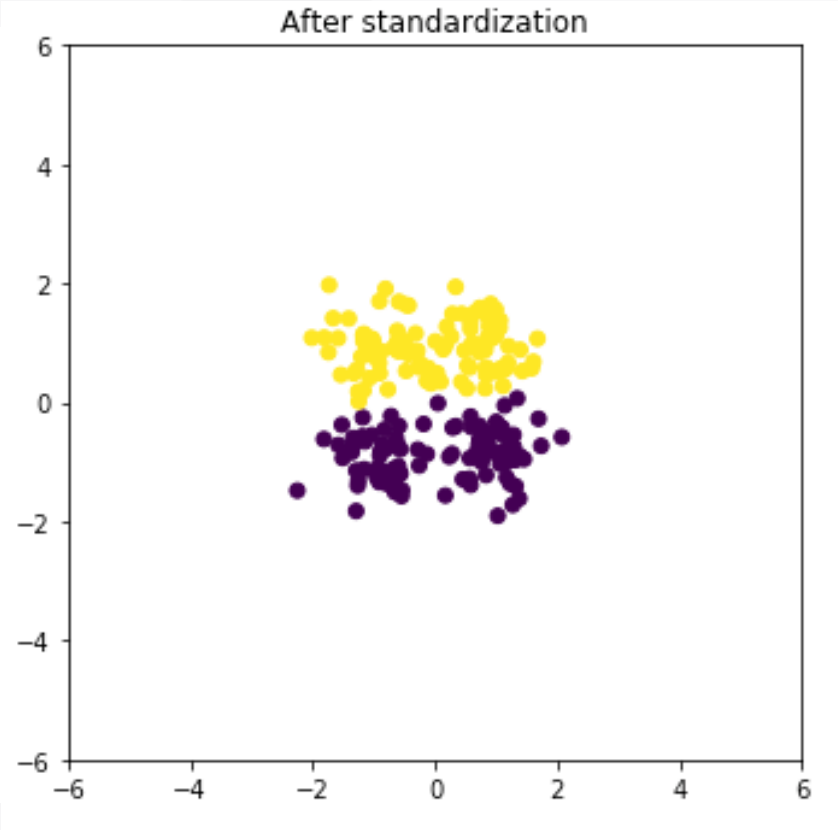


标准化

标准化并不一定起效







参考结果

无标准化

标准化

其他相似/不相似函数

- 针对二值数据的Jaccard 系数
- 刻画变量之间的相关性系数作为相似性度量
- 无序属性上的值差异性度量
- 向量间的余弦相似度(Cosine similarity)
- …

大纲

- 简介
- 距离函数
- 评价指标
- 聚类算法
  - K均值聚类
  - 高斯混合模型和EM 算法
  - 层次聚类
  - 基于密度的聚类

聚类的评价

- 聚类的性能评价:有效性指标（Validity Index）
- 评价指标:
  - 有参考模型 (外部指标)
  - 无参考模型(内部指标)

参考模型

- 数据集: $D = \{x_1, x_2, \ldots x_m\}$
- 聚类的簇: $C = \{C_1, C_2, \ldots C_k\}$
- 参考的簇: $C^* = \{C_1, C_2, \ldots C_s\}$
- λ 和λ*分别为C 和 C^* 为簇标记向量
- 样本对: $(x_i, y_i), i \leq i < j \leq m$

m(m-1)/2		参考	
		same	not
聚类结果	same	a	b
	not	c	d

- 利用(a, b, c, d)定义外部评价指标

a ↑, 一致性 ↑
b ↑, 一致性 ↓
c ↑, 一致性 ↓
d ↑, 一致性 ↑

外部指标

- Jaccard 系数(Jaccard Coefficient), 简称JC

$$JC = \frac{a}{a+b+c} \qquad JC \in [0,1], \quad JC \uparrow, \text{一致性} \uparrow$$

- FM指数(Fowlkes and Mallows index),简称FMI

$$FMI = \sqrt{\frac{a}{a+b}\frac{a}{a+c}} \qquad FMI \in [0,1], \quad FMI \uparrow, \text{一致性} \uparrow$$

- Rand指数(Rand Index),简称RI

$$RI = \frac{2(a+d)}{m(m-1)} \qquad RI \in [0,1], \quad RI \uparrow, \text{一致性} \uparrow$$

无参考模型

- 只有聚类结果，没有外部专家给出的参考结果，我们该如何评价？
  - 簇内相似性越高，聚类质量越好
  - 簇间相似度越低，聚类质量越好

无参考模型

- 簇内相似度
  - 平均距离

$$avg(C) = \frac{2}{|C|(|C|-1)} \sum_{1 \le i < j \le |C|} dist(x_i, x_j)$$

  - 最远距离

$$diam(C) = \max_{1 \le i < j \le |C|} dist(x_i, x_j)$$

- 簇间相似度
  - 最小距离

$$d_{min}(C_i, C_j) = \min_{x_i \in C_i, x_j \in C_j} dist(x_i, x_j)$$

  - 中心点间的距离

$$d_{cen}(C_i, C_j) = dist(\mu_i, \mu_j), where\ \mu_i = \frac{1}{|C_i|} \sum_{x_i \in C_i} x_i$$

内部指标

- DB指数(Davies-Bouldin Index,简称DBI)

$$DBI = \frac{1}{k}\sum_{i=1}^{k}\max_{i\neq j}\frac{avg(C_i)+avg(C_j)}{d_{cen}(\mu_i,\mu_j)}, \qquad DBI\downarrow, \text{聚类质量}\uparrow$$

- Dunn指数(Dunn Index，简称DI)

$$DI = \min_{1\le i<j\le k}\frac{d_{min}(C_i,C_j)}{\max_{1\le l\le k} diam(C_l)}, \qquad DI\uparrow, \text{聚类质量}\uparrow$$

大纲

- 简介
- 距离函数
- 评价指标
- 聚类算法
  - K均值聚类
  - 高斯混合模型和 EM 算法
  - 层次聚类
  - 基于密度的聚类

K均值聚类

- 问题：给定一组样本点$X = \{x_i\}^N$进行聚类

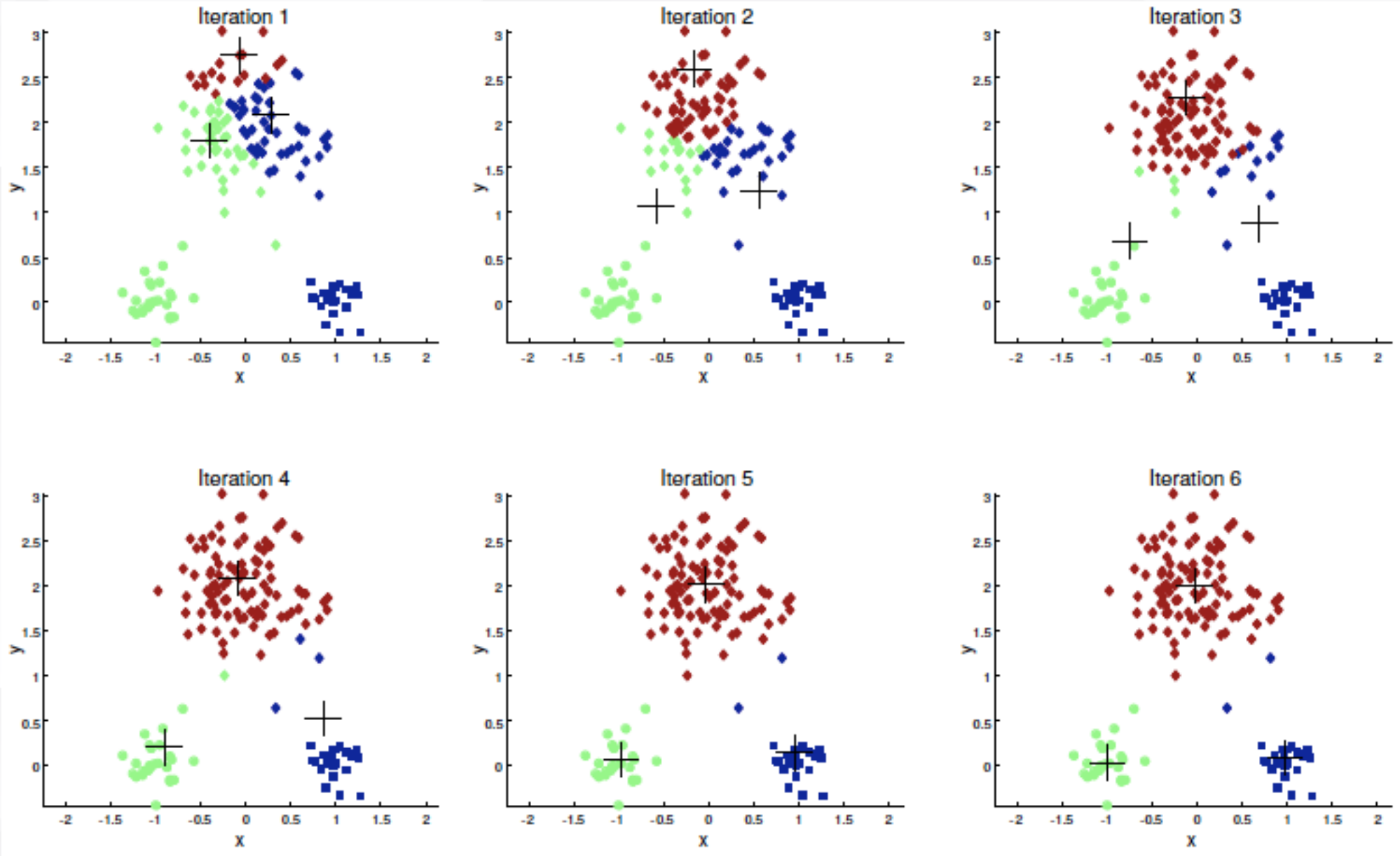
==========

- 输入：数据$D = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$，簇数目K

1. 随机选取K个种子数据点(seeds)作为K个簇中心
2. repeat
3. foreach $\mathbf{x} \in \boldsymbol{D}$ do
4. 计算$\mathbf{x}$与每一个簇中心的距离
5. 将$\mathbf{x}$指配到距离最近的簇中心
6. endfor
7. 用当前的簇内点重新计算K个簇中心位置
8. until 当前簇中心未更新

==========

K-means运行示意


Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 6
x
y

这么简单直白的算法为什么是一个好的聚类算法?

假设是什么？优化目标是什么?

K-means的基本假设与优化目标

■K均值（K-means）聚类：基于划分的聚类方法

1. 如何表示簇？
   - 每个簇都用其质心（centroid）或者叫原型（prototype）μ_k表示
2. 如何划分节点？
   - 距离使用欧式距离进行度量
   - 每个节点都划分到最近的那个质心的簇中
   - $r_{ik} \in \{0,1\}$ 为从属度，指示样本x_i是否属于簇k，且$\sum_{k=1}^{K} r_{ik} = 1$
3. 优化目标：

K-means暗含了对簇的什么假设？

损失函数 $J = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} ||x_i - \mu_k||^2$，平方误差和(SSE)

如何优化?

- 如何最小化损失函数 J w.r.t (r_{ik},u_k)?
- Chicken and egg problem
  - 如果中心点已知，我们可以对所有样本点进行从属划分
  - 如果从属关系已知，我们可以计算中心点
- 我们采用迭代的方式

如何优化?

- 固定μ_k, 最小化J w.r.t. r_{ik}
  - 分配每个样本点到其最近的中心点(prototype)所在的簇
- 固定r_{ik} , 最小化 J w.r.t. μ_{ik}
  - 计算每个簇的点的均值作为中心

i.e., $\mu_k = \frac{\sum_i r_{ik} x_i}{\sum_i r_{ik}}$

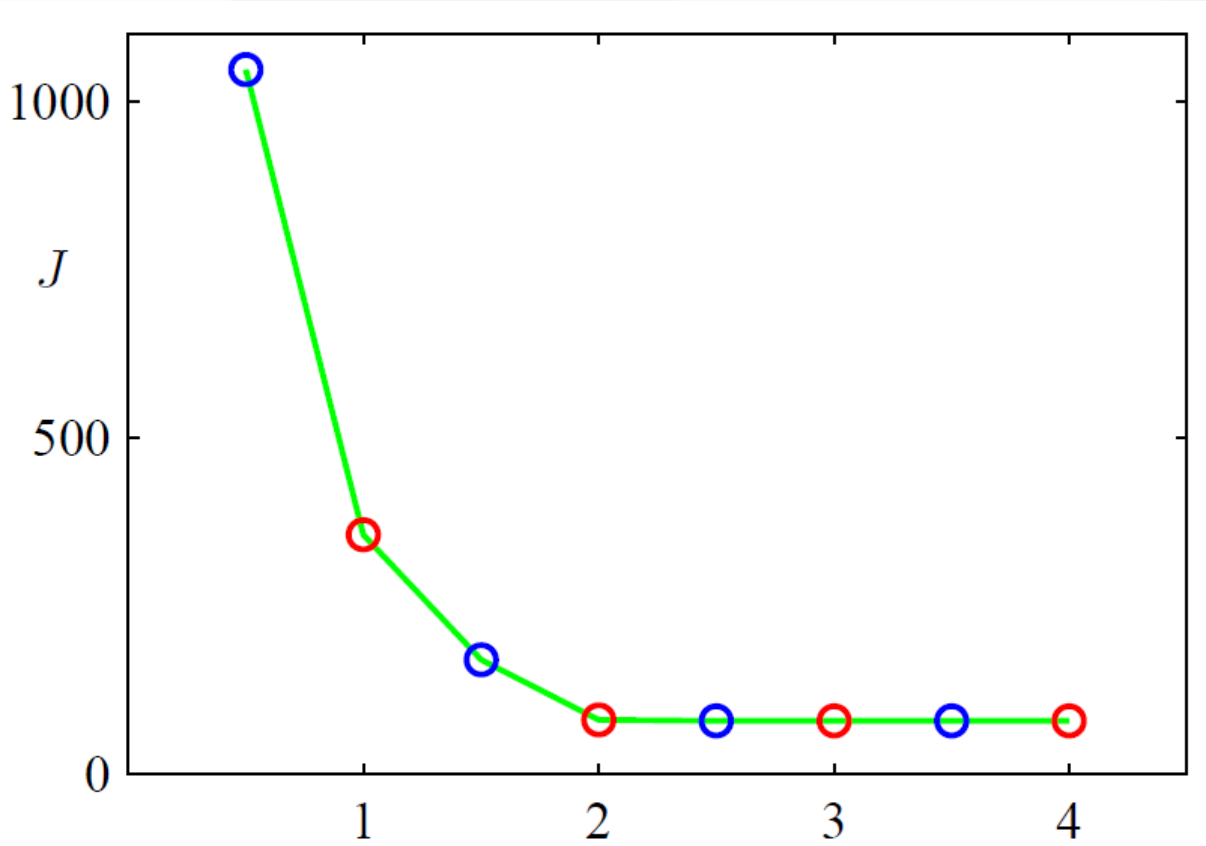
这个过程能保证停下来么?

优化过程会收敛么?

k-means收敛性

- k-means 是在损失函数上进行坐标下降(coordinate descent)的优化
- 损失函数J 单调下降, 所以损失函数值会收敛, 所以聚类结果也会收敛
- k-means 有可能会在不同聚类结果间震荡，但是在实际中较少发生
- J 是非凸的(non-convex), 所以损失函数J上应用坐标下降法不能够保证收敛到全局的最小值. 一个常见的方法是运行k-means多次，选择最好的结果

损失函数 J 随着迭代次数变化

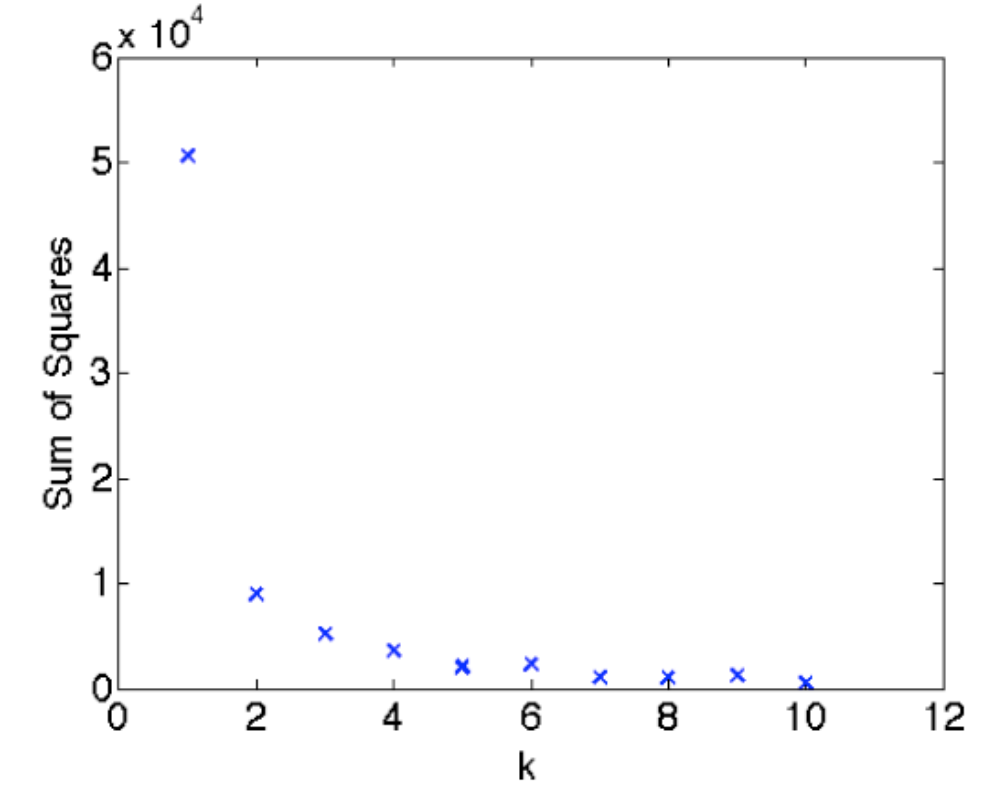


谁能告诉我到底该聚成几个类?

参数K如何确定?

如何选择K？

- K是该算法的超参
- 损失函数J 一般 随着K的增大而递减 .

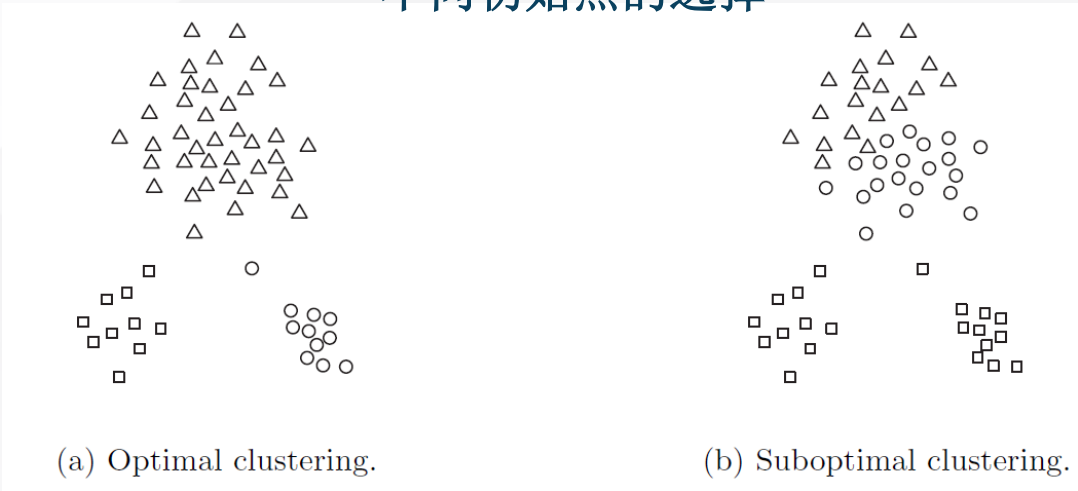


如何选择K？

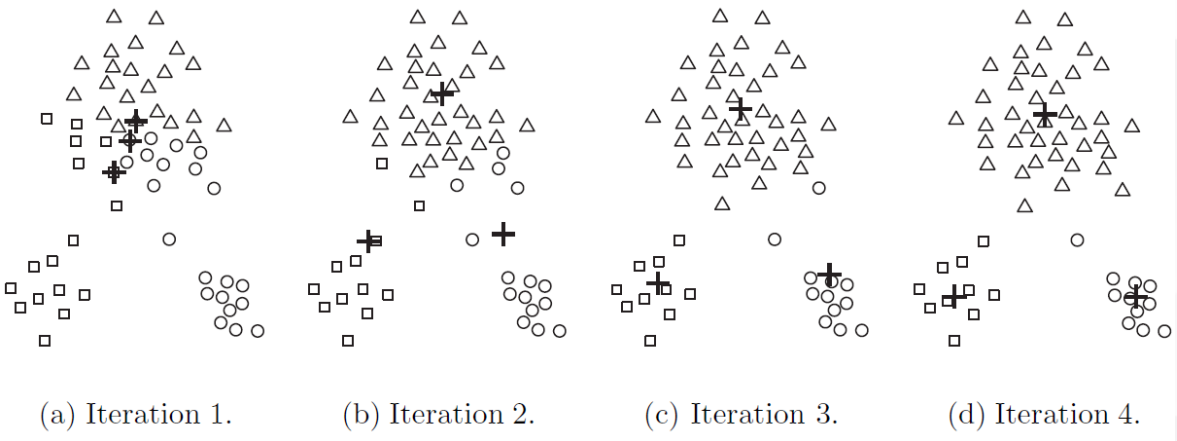
- 间隔统计：分析损失函数J随K增大递减的间隔
- 交叉检验: 把原始数据集分裂两个子集.在其中一个数据集上估计中心点(prototypes)，然后在另一个集合上计算损失函数
- 簇的稳定性: 通过对原始数据的重采样或者分裂，度量簇的改变程度.
- 非参数方法: 为K加上一个先验， Bayesian non-parametric
  - 例如：中国餐馆过程(Chinese Restaurant Process)

如何初始化K-means?

不同初始点的选择

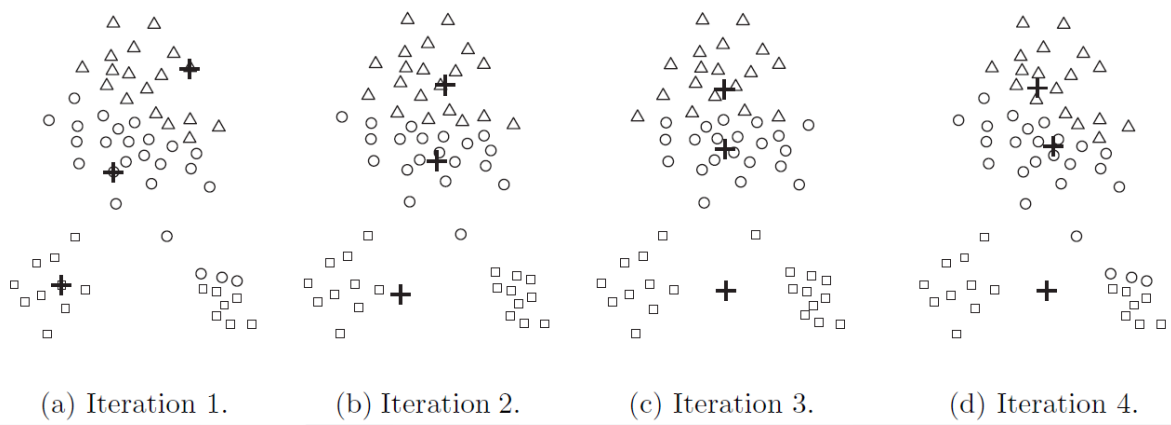


(a) Optimal clustering.　　(b) Suboptimal clustering.



(a) Iteration 1.　(b) Iteration 2.　(c) Iteration 3.　(d) Iteration 4.

Good Clustering



(a) Iteration 1.　(b) Iteration 2.　(c) Iteration 3.　(d) Iteration 4.

Poor Clustering

如何初始化K-means?

- 即便存在K个真实的簇，正好选到K个簇的中心的机会也将会是很小的

- 一些启发式做法
  - 随机选择K个数据点作为中心点
  - 选择第i+1个中心时，选择与距离之前选出的中心点距离最远的

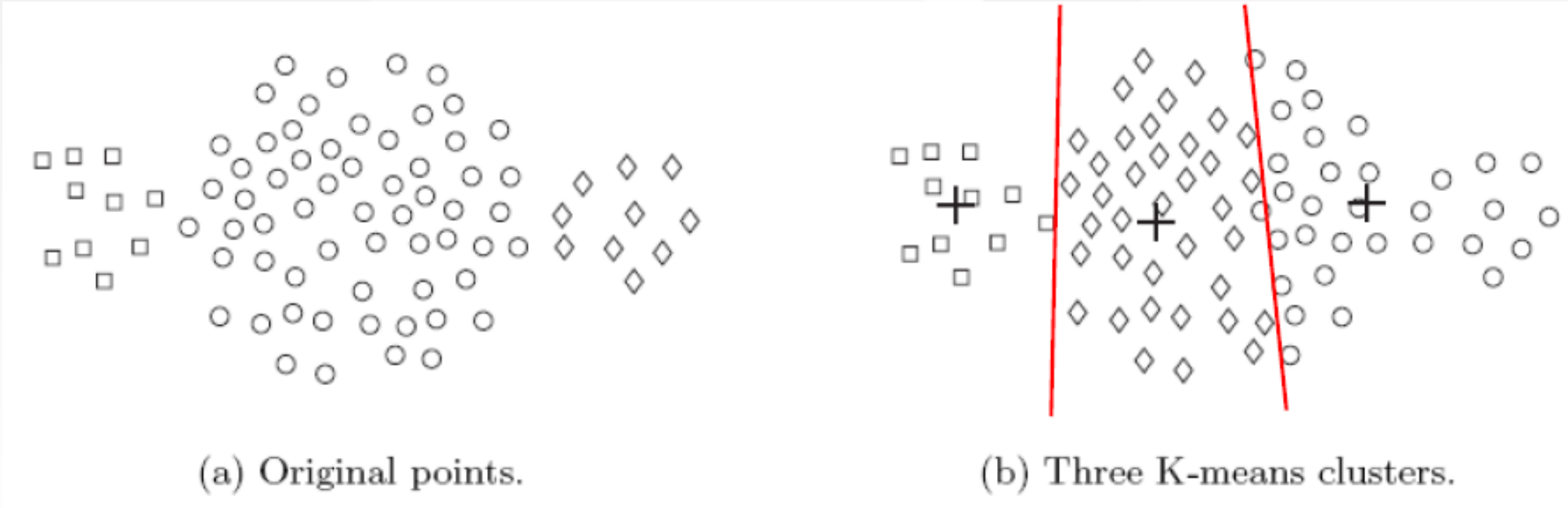
预处理和后处理

- 预处理
  - 归一化数据 (e.g.,缩放到单位标准差)
  - 消除离群点
- 后处理
  - 删除小的簇：可能代表离群点
  - 分裂松散的簇：簇内节点间距离之和很高
  - 合并距离较近的簇

K-means的局限性

- K-means 存在问题，当簇具有不同的
  - 尺寸
  - 密度
  - 非球形
- K-means可能得不到理想的聚类结果

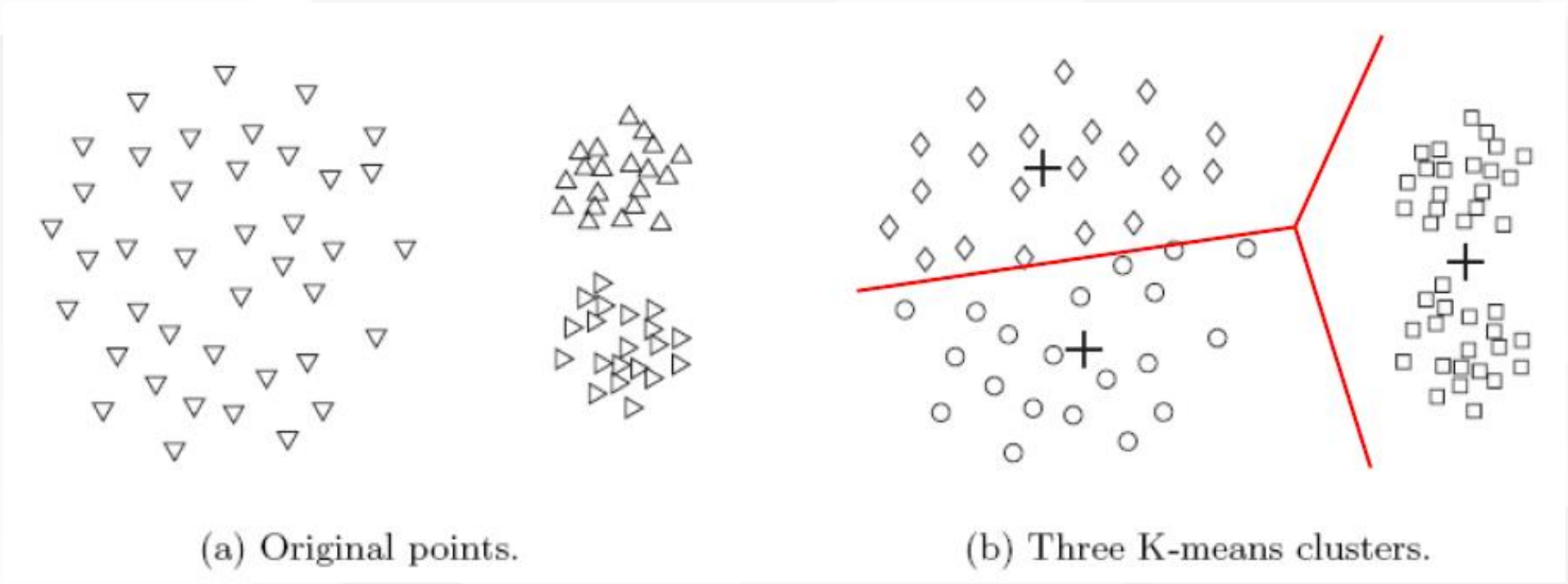
K-means的局限性: 不同的尺寸



(a) Original points.

(b) Three K-means clusters.

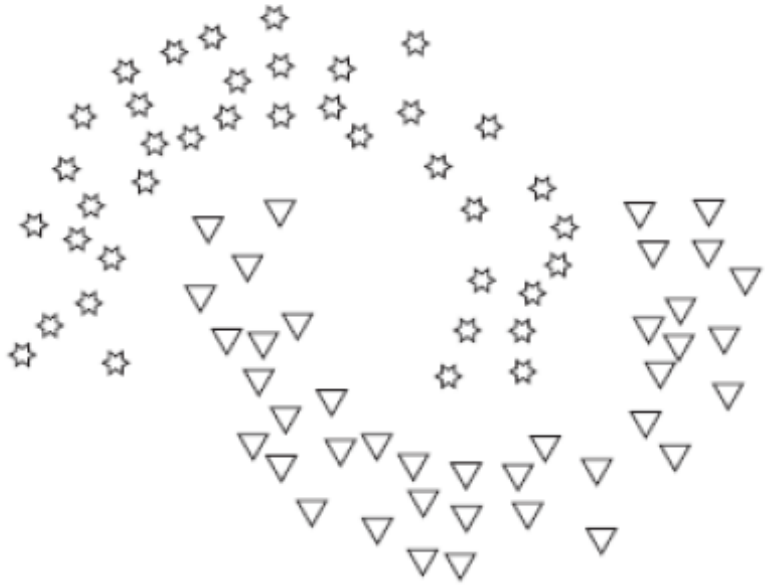
K-means的局限性: 不同的密度



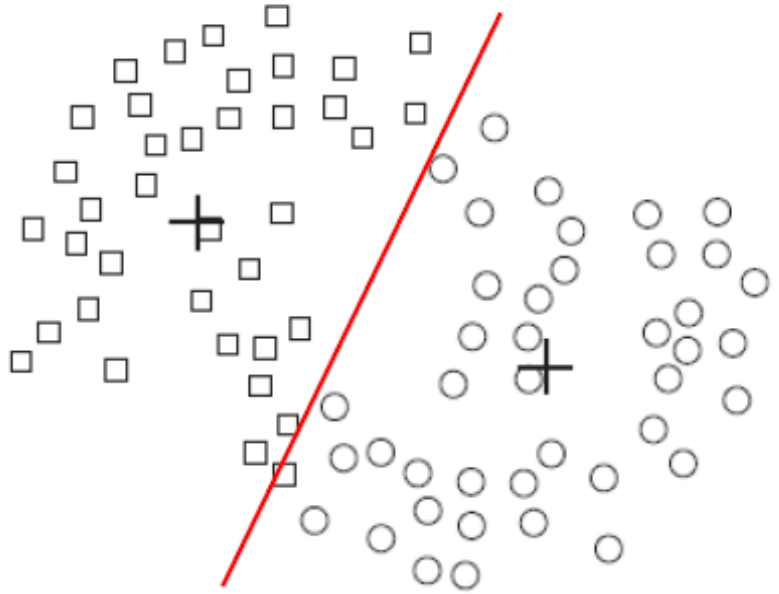
(a) Original points.

(b) Three K-means clusters.

K-means的局限性: 非凸的形状



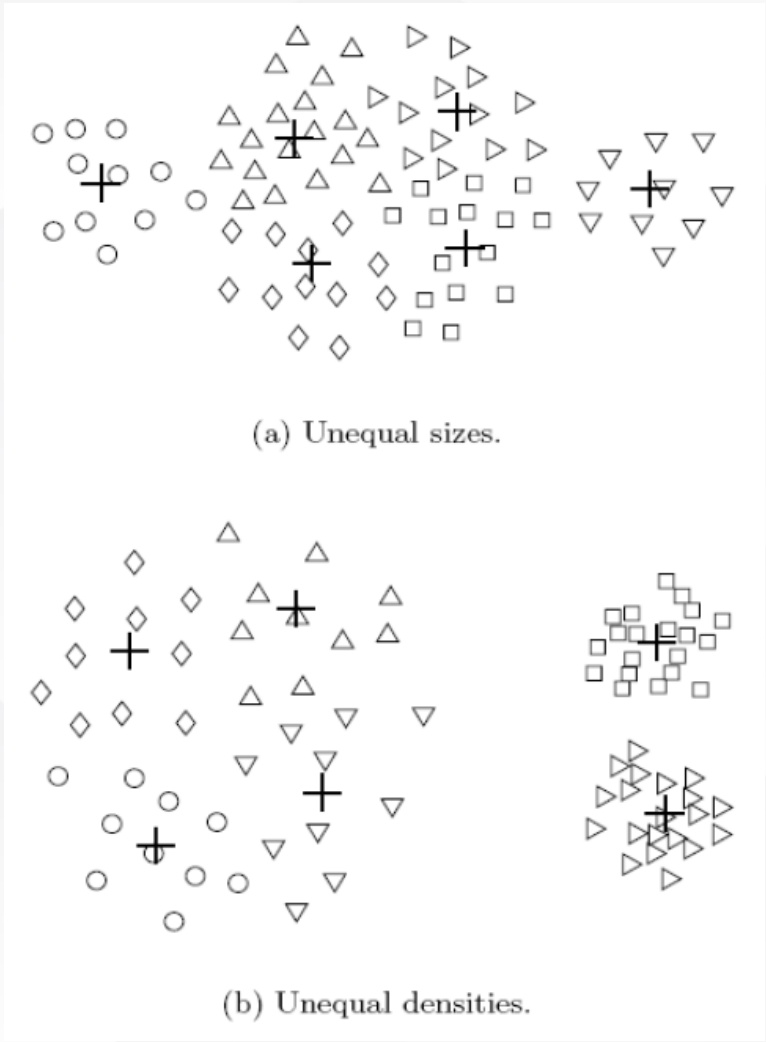
(a) Original points.



(b) Two K-means clusters.

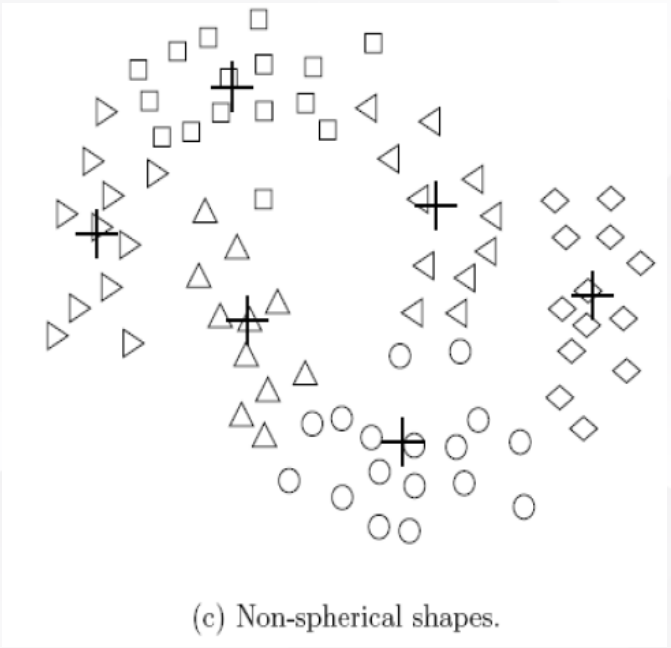
克服 K-means 的这些局限性

- 使用更大数量的簇
- 几个小的簇表示一个真实的簇
- 使用基于密度的方法



(a) Unequal sizes.

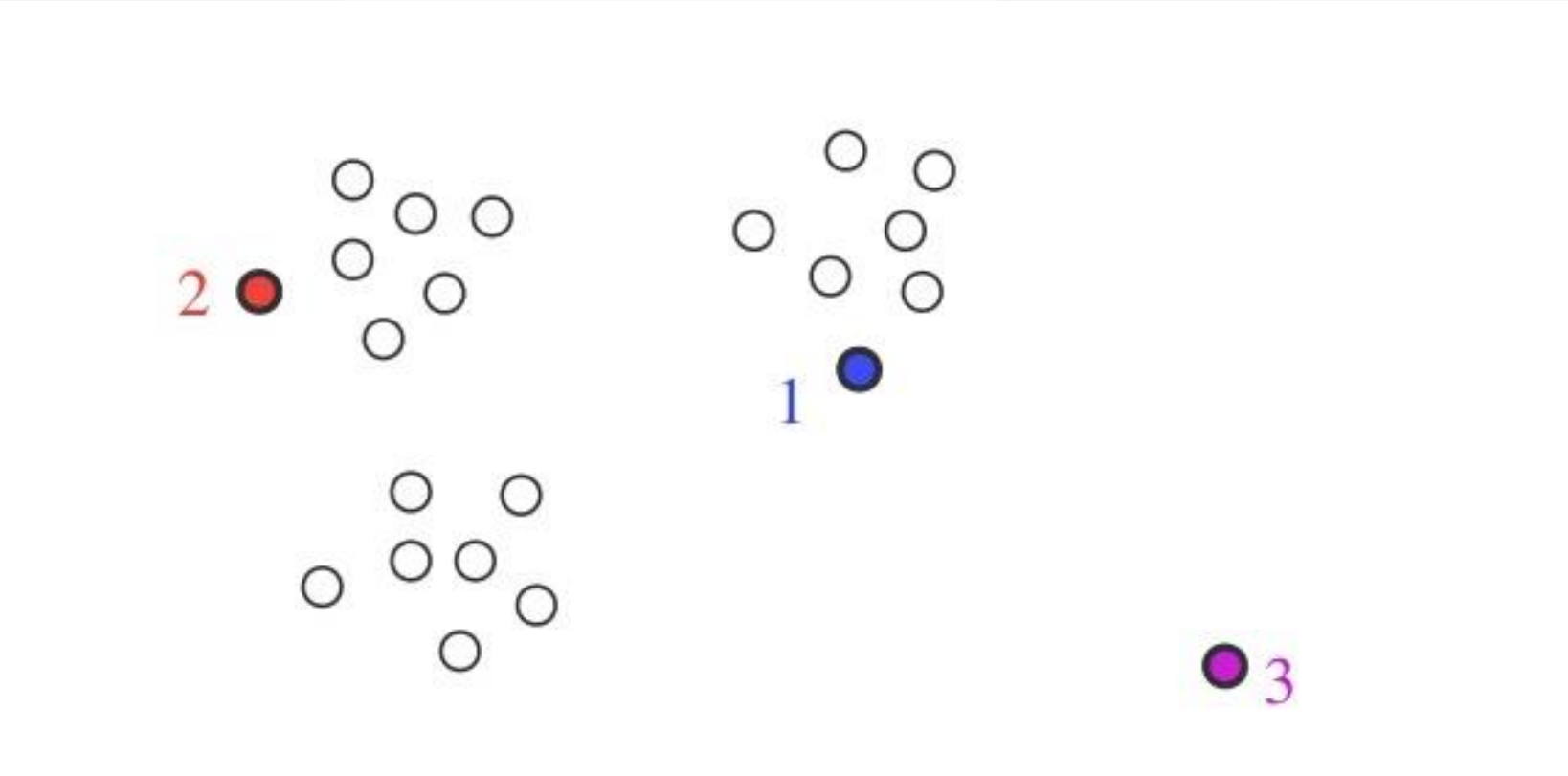
(b) Unequal densities.



(c) Non-spherical shapes.

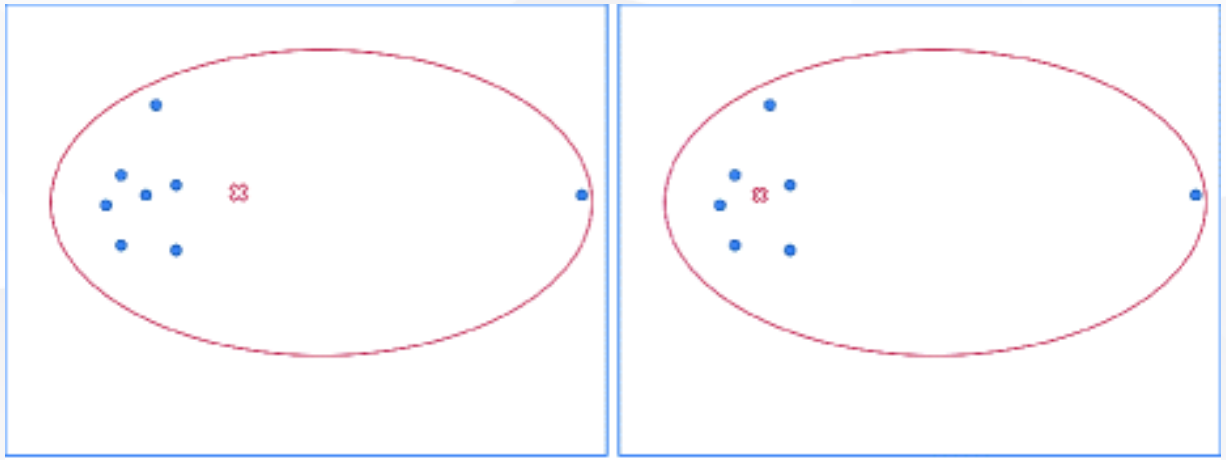
K-means的局限性

- 离群点带来的问题



K-medoids

- 均值作为原型易受影响
  - 原型是簇的中位点(median)
- 部分情况只知道数据样本间的相似矩阵
  - 只需要数据间的相似度量即可迭代计算



K-means的局限性

- 硬划分数据点到簇，当数据上出现一些小的扰动，可能导致一个点划分到另外的簇
- 假定簇为球形且每个簇的概率相等
- **解决方案: 高斯混合模型**

大纲

- 简介
- 距离函数
- 评价指标
- 聚类算法
  - K均值聚类
  - 高斯混合模型和EM算法
  - 层次聚类
  - 基于密度的聚类

多变量高斯分布

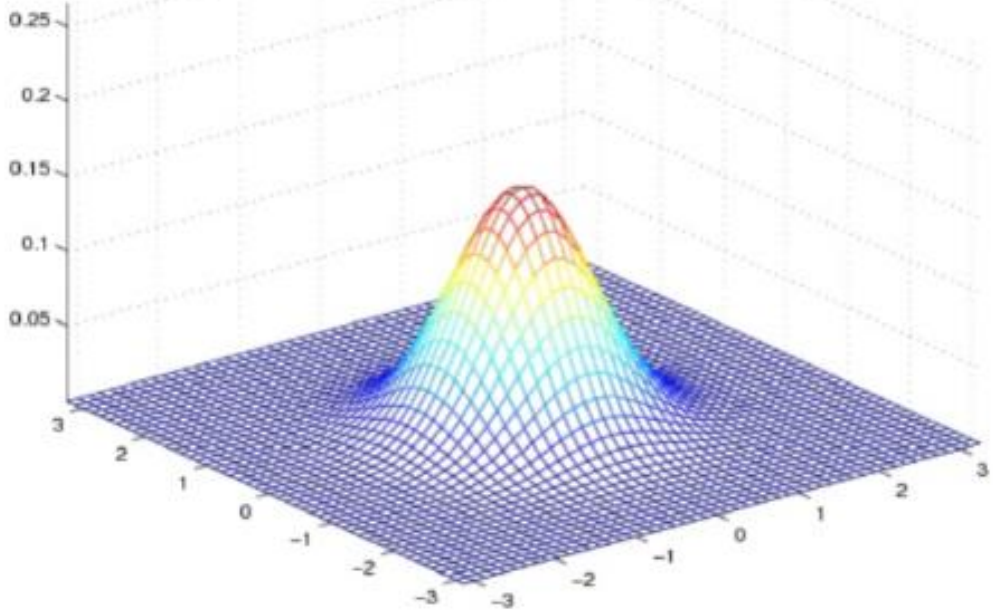


Figure 6: Multivariate Normal Distribution

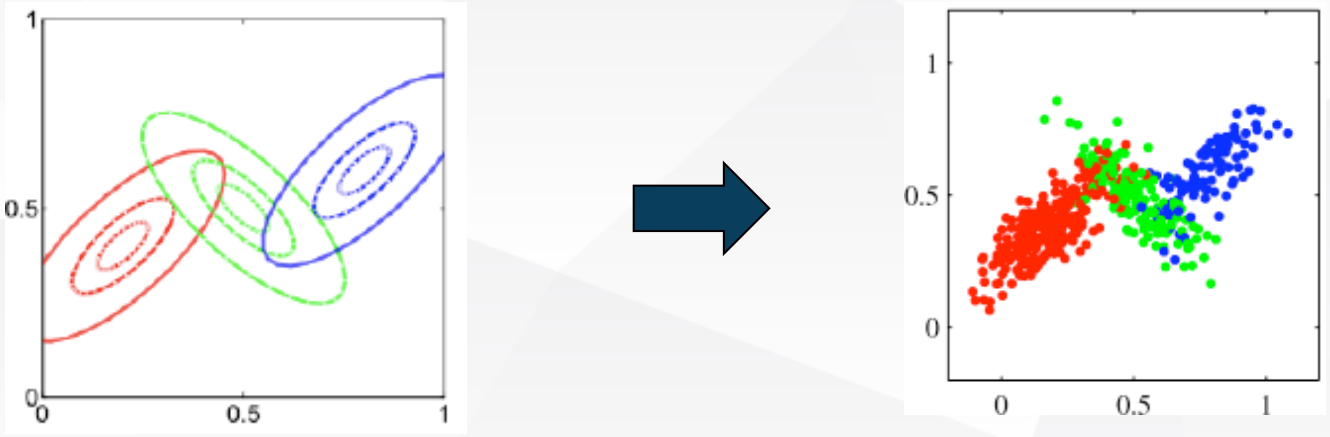
$$x \sim \mathcal{N}(\mu, \Sigma)$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)}$$

高斯混合模型

- 概率解释: 假设有K个簇，每一个簇服从高斯分布，以概率π_k随机选择一个簇 k，从其分布中采样出一个样本点，如此得到观测数据
- 似然函数(Likelihood)

$$P(\boldsymbol{x};\theta)=\sum_{k=1}^{K}\pi_k\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\ \text{其中}\ \sum_{k=1}^{K}\pi_k=1,\ \ 0\le\pi_k\le 1$$



为高斯混合模型引入隐变量

- 为每个样本点$\boldsymbol{x}$关联一个K维的隐变量$\boldsymbol{z}=(z_1,\dots z_K)$指示样本所属的簇，因此$\boldsymbol{z}$采用独热(one-hot)表示

$$P(z_k=1)=\pi_k$$

$$P\left(\boldsymbol{x}\middle|\boldsymbol{z}\right)=\prod_{k=1}^{K}\mathcal{N}\left(\boldsymbol{x}\middle|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right)$$

$$P(\boldsymbol{x})=\sum_{\boldsymbol{z}}P(\boldsymbol{x}|\boldsymbol{z})P(\boldsymbol{z})=\sum_{k=1}^{K}\pi_k\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$$

从属度

■ 给定$\boldsymbol{x}$，可以计算$\boldsymbol{z}$的条件概率

$$\gamma(z_k) = p(z_k = 1|\boldsymbol{x}) = \frac{P(z_k = 1)p(\boldsymbol{x}|z_k = 1)}{\sum_{k=1}^{K}(z_k = 1)p(\boldsymbol{x}|z_k = 1)}$$

$$= \frac{\pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

■ $\gamma(z_k)$ 可以看做是对$\boldsymbol{x}$从属于第k个簇的一种估计或者“解释”

混合高斯模型的学习过程困难

- 参数学习：极大似然估计
  - 最大化以下的对数似然函数(log likelihood)

$$\log P(\boldsymbol{x}|\theta) = \sum_{i=1}^{n} \log\{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}$$

- 为什么这个函数优化很困难?
  - log 里有求和，导致求导困难，所有参数耦合在一起

如何去解?

- 我们先写下似然函数取最大值时满足的条件， $\log P(\boldsymbol{x}|\theta)$对$\boldsymbol{\mu}_k$求导

$$0 = -\sum_{i=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}(x_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}}_{\gamma(z_{ik})} \boldsymbol{\Sigma}_k (x_i - \boldsymbol{\mu}_k)$$

- 我们获得

$$\boldsymbol{\mu}_k = \frac{\sum_i \gamma(z_{ik})\, x_i}{\sum_i \gamma(z_{ik})}$$

如何求解?

- 类似地, 我们得到

$$\pi_k = \frac{\sum_i \gamma(z_{ik})}{N}, \qquad \Sigma_k = \frac{\sum_i \gamma(z_{ik})\,(x_i - \boldsymbol{\mu}_k)(x_i - \boldsymbol{\mu}_k)^T}{\sum_i \gamma(z_{ik})}$$

- 注意这不是封闭解，$\gamma(z_{ik})$依赖于参数

- 这暗示了一种寻找解的迭代方案，这就是EM算法在GMM下的实例
  - E步：给定当前参数的估计值计算后验概率，或者叫从属度
  - M步: 基于当前的从属度，重新估计参数的值

EM(Expectation-Maximization) 算法

1. 初始化参数并计算对数似然
2. E步: 基于当前参数计算从属度

$$\gamma(z_{ik}) \triangleq E(z_{ik}) = \frac{\pi_k N(x_i|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_k N(x_i|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)}$$

3. M步: 基于当前从属度重新估计参数

$$\pi_k = \frac{\sum_i \gamma(z_{ik})}{N},\ \mu_k = \frac{\sum_i \gamma(z_{ik}) x_i}{\sum_i \gamma(z_{ik})},\ \Sigma_k = \frac{\sum_i \gamma(z_{ik})(x_i-\mu_k)(x_i-\mu_k)^T}{\sum_i \gamma(z_{ik})}$$

3. 迭代到似然函数收敛

- 该算法将收敛到似然函数的局部最优

EM算法的另一个视角

- 目标函数是对数似然函数

$$\log P(\boldsymbol{X}|\boldsymbol{\theta}) = \log \sum_{\boldsymbol{Z}} P(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})$$

- 如果我们知道完整数据 $\{\boldsymbol{X}, \boldsymbol{Z}\}$, 那么完整数据的对数似然函是很直接的

$$\log P(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})$$

- 然而, 关于隐变量$\boldsymbol{Z}$ 的值的信息只能从后验分布得到 $p\left(\boldsymbol{Z}\middle|\boldsymbol{X}, \boldsymbol{\theta}^{\boldsymbol{old}}\right)$. 因此, 我们就不妨考虑其期望值 (即E步)

$$\mathrm{Q}\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{\boldsymbol{old}}\right) = \mathrm{E}\{\log P(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})\} = p\left(\boldsymbol{Z}\middle|\boldsymbol{X}, \boldsymbol{\theta}^{\boldsymbol{old}}\right) \log P(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})$$

- 在 M 步, 我们最大化期望

$$\boldsymbol{\theta}^{\boldsymbol{new}} = \arg\max_{\boldsymbol{\theta}} \mathrm{Q}\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{\boldsymbol{old}}\right)$$

通用的EM算法

1. 初始化参数 $\boldsymbol{\theta}^{\boldsymbol{old}}$
2. E步: 计算

$$p\left(\boldsymbol{Z}\middle|\boldsymbol{X},\boldsymbol{\theta}^{\boldsymbol{old}}\right)$$

3. M步: 计算 $\boldsymbol{\theta}^{\boldsymbol{new}}$ 通过

$$\boldsymbol{\theta}^{\boldsymbol{new}} = \arg\max_{\boldsymbol{\theta}} \mathrm{Q}\left(\boldsymbol{\theta},\boldsymbol{\theta}^{\boldsymbol{old}}\right)$$

其中

$$\mathrm{Q}\left(\boldsymbol{\theta},\boldsymbol{\theta}^{\boldsymbol{old}}\right) = \mathrm{E}\{\log P(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\theta})\} = p\left(\boldsymbol{Z}\middle|\boldsymbol{X},\boldsymbol{\theta}^{\boldsymbol{old}}\right)\log P(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\theta})$$

4. 检查似然函数或者参数的值是否收敛，如果没有达到收敛条件，那么令 $\boldsymbol{\theta}^{\boldsymbol{old}} \leftarrow \boldsymbol{\theta}^{\boldsymbol{new}}$，重复第2步

高斯混合模型回顾

- 对于完整数据的对数似然函数如下

$$\log P(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}) = \log P(\boldsymbol{Z}|\boldsymbol{\pi})\, P(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{\mu},\boldsymbol{\Sigma}) = \sum_{i=1}^{N}\sum_{k=1}^{K} z_{ik}\{log\pi_k + logN(x_i|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\}$$

- 和不完整数据的对数似然函数对比

$$\log P(\boldsymbol{X}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}) = \sum_{i=1}^{n} \log\left\{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\right\}$$

  - 注意π_k 和(μ_k,Σ_k) 在完整数据似然函数中解耦合，存在平凡的封闭解

- 完整数据的对数似然函数的期望如下

$$\mathbb{E}_Z\{\log P(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma})\} = \sum_{i=1}^{N}\sum_{k=1}^{K} \mathbb{E}[z_{ik}]\{log\pi_k + logN(x_i|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\}$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma(z_{ik})\{log\pi_k + logN(x_i|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\}$$

通用EM算法

- EM算法是一种寻找含有隐变量的概率模型的最大似然解的通用技术
- **为什么我们通过这种启发式的方式推导出来的EM算法，但确实是在最大化似然函数？**

$$\boldsymbol{\theta}^{new} = \arg\max_{\boldsymbol{\theta}} \mathrm{Q}\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}\right) \quad \Rightarrow \quad \boldsymbol{\theta} = \arg\max_{\boldsymbol{\theta}} \mathrm{P}(\boldsymbol{x}, \boldsymbol{\theta})$$

通用EM算法

- 考虑一个概率模型
  - 可观测变量**X** 和隐变量 **Z**
  - 联合概率分布为$P(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})$
- 我们的目标是最大化如下的似然函数

$$P(\boldsymbol{X}|\boldsymbol{\theta}) = \sum_Z P(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})$$

- 我们假定直接优化$P(\boldsymbol{X}|\boldsymbol{\theta})$很困难, 但是优化完整数据的似然函数$P(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})$显著的容易

通用EM算法

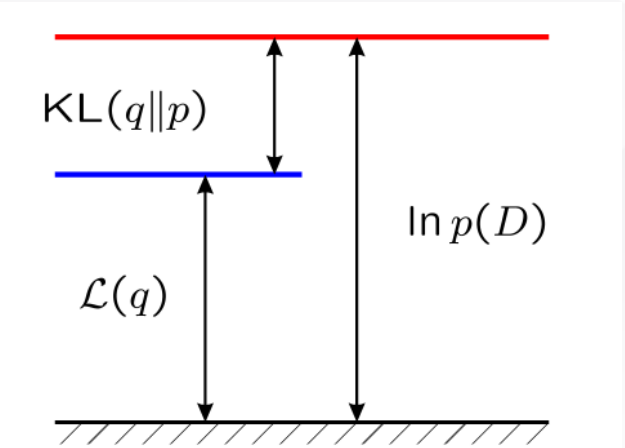
对于任意的分布q(Z), 下列的分解成立

$$\ln P(\boldsymbol{X}|\boldsymbol{\theta}) = \mathcal{L}(q,\boldsymbol{\theta}) + KL(q||p)$$

其中

$$\mathcal{L}(q,\boldsymbol{\theta}) = \sum_{z} q(\boldsymbol{Z}) ln\left\{\frac{P(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\theta})}{q(\boldsymbol{Z})}\right\}$$

$$KL(q||p) = -\sum_{z} q(\boldsymbol{Z}) ln\left\{\frac{p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta})}{q(\boldsymbol{Z})}\right\}$$



KL(q||p)≥0, $\mathcal{L}(q,\boldsymbol{\theta})$ 是$\ln P(\boldsymbol{X}|\boldsymbol{\theta})$的下界

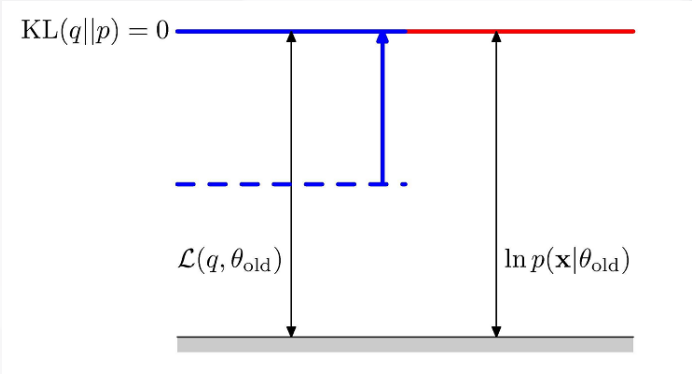
EM: 变分观点

- 针对一个自由形式的q 分布，如果我们最大化$\mathcal{L}(q, \boldsymbol{\theta})$

E 步

$$q(\boldsymbol{Z}) = P(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{\boldsymbol{old}})$$

这就正好得到了真正的后验分布

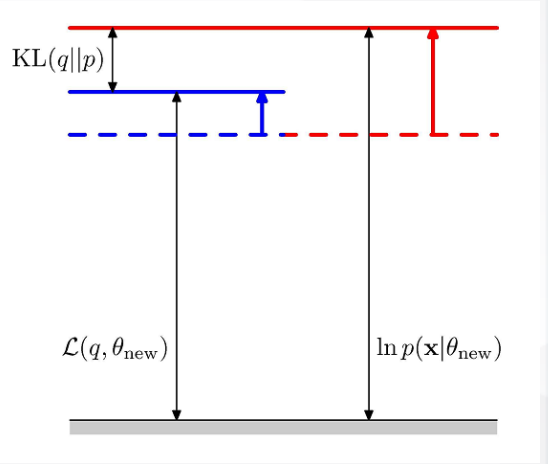


- 这时候，原来的下界成为

M 步

$$\mathcal{L}(q, \boldsymbol{\theta}) = \sum_z P\left(\boldsymbol{Z}\middle|\boldsymbol{X}, \boldsymbol{\theta}^{\boldsymbol{old}}\right) ln \frac{P(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})}{P\left(Z\middle|X, \boldsymbol{\theta}^{\boldsymbol{old}}\right)}$$

$$= \mathrm{Q}\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{\boldsymbol{old}}\right) + \text{const}$$



L作为θ的函数正好是完整数据对数似然的期望 （加上某个常数）

EM算法收敛性保证

$$\ln P\left(\boldsymbol{X}\middle|\boldsymbol{\theta}^{(t+1)}\right) = \mathcal{L}\left(q, \boldsymbol{\theta}^{(t+1)}\right) + KL(q||p)$$

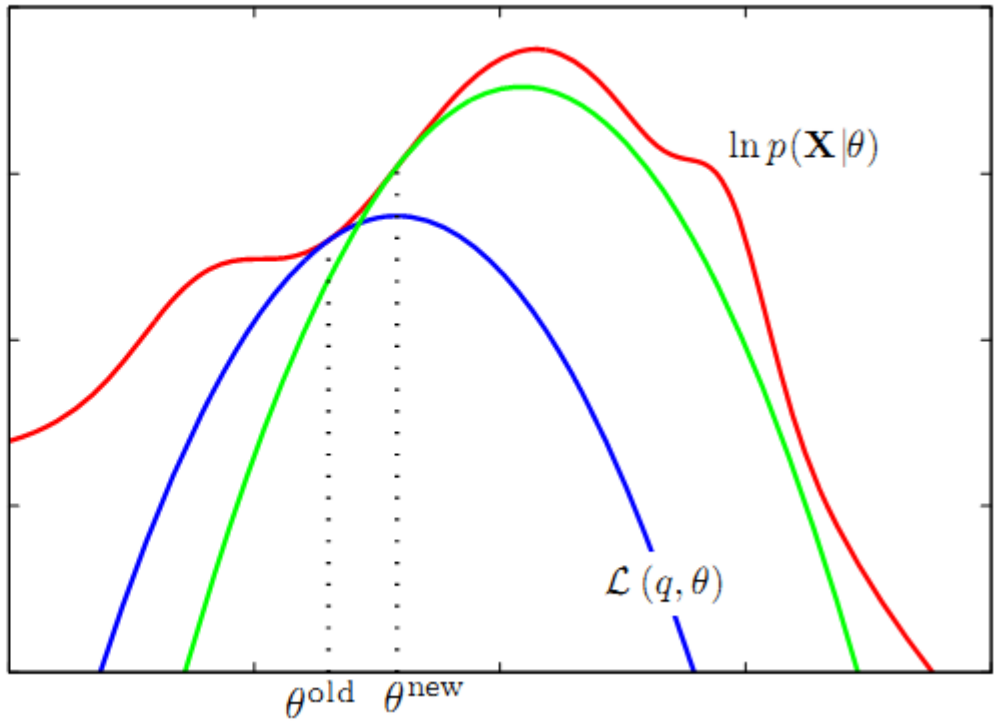
$$\geq \sum_{x}\sum_{z} P\left(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)}\right) \log \frac{P\left(\boldsymbol{X}, \boldsymbol{Z}\middle|\boldsymbol{\theta}^{(t+1)}\right)}{P\left(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)}\right)}$$

$$\geq \sum_{x}\sum_{z} P\left(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)}\right) \log \frac{P\left(\boldsymbol{X}, \boldsymbol{Z}\middle|\boldsymbol{\theta}^{(t)}\right)}{P\left(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)}\right)}$$

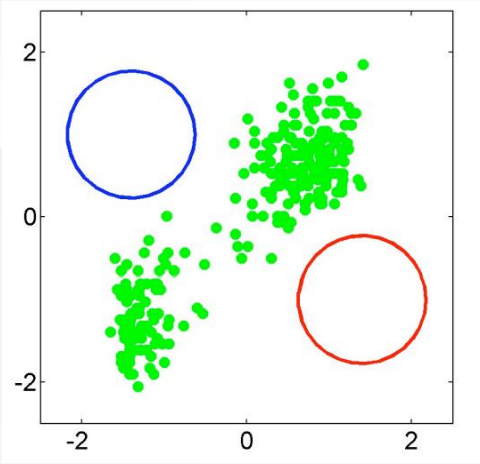
$$= \ln P\left(\boldsymbol{X}\middle|\boldsymbol{\theta}^{(t)}\right)$$

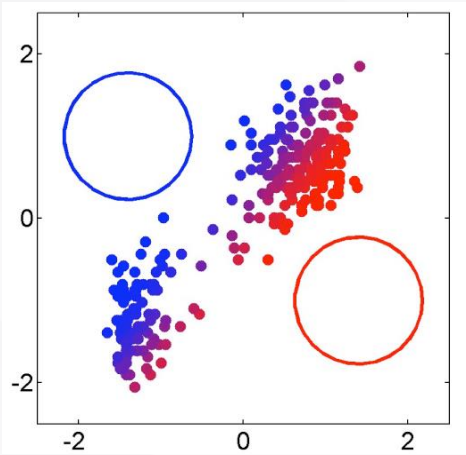
EM算法

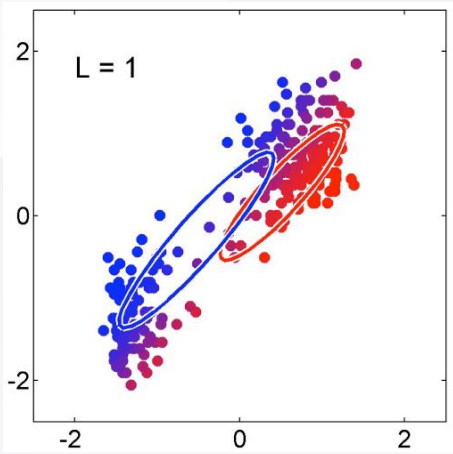
The EM algorithm involves alternately computing a lower bound on the log likelihood for the current parameter values and then maximizing this bound to obtain the new parameter values. See the text for a full discussion.

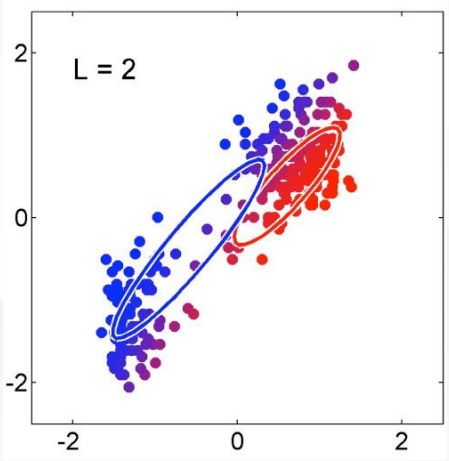


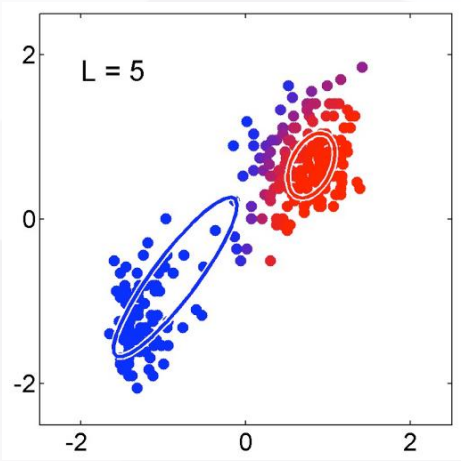
例子

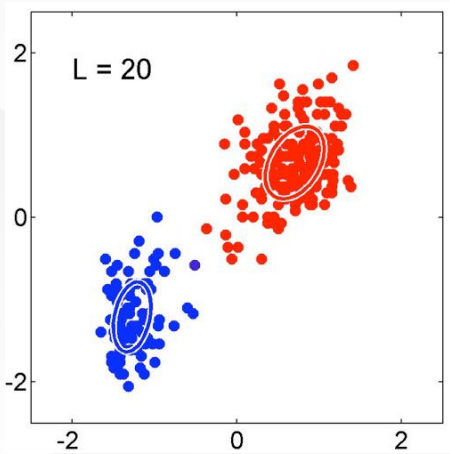












高斯混合模型: 与K-means的联系

- 高斯混合模型的E步是一个软划分版本的 K-means. $r_{ik} \in [0,1]$
- 高斯混合模型的M步估计除了估计均值外还估计协方差矩阵
- 当所有π_k 相等， $\sum_k = \delta^2 \mathrm{I}$， 当$\delta^2 \to 0$， $r_{ik} \to \{0,1\}$, 那么两个方法是一致的

K-means vs 高斯混合模型(GMM)

K-means

- 损失函数: 最小化平方距离的和.
- 样本点硬划分到某个簇
- 假定样本属于每个簇的概率相等，且为球形簇

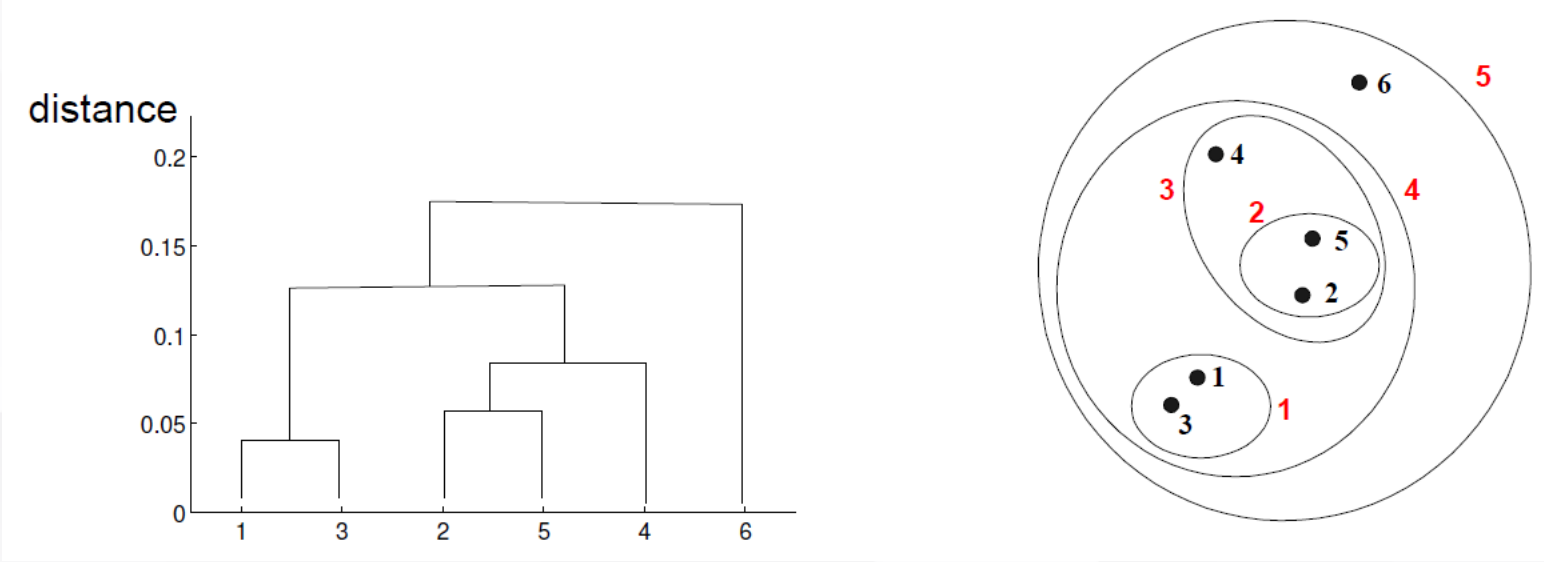
GMM

- 最小化负对数似然
- 点到簇的从属关系为软分配.
- 可以被用于非球形簇，且各个簇概率不同

Outline

层次聚类

- 产生树形嵌套的聚类簇
- 可以被可视化为树状图(dendrogram)
  - 树形的示意图，记录了簇合并或分割的序列

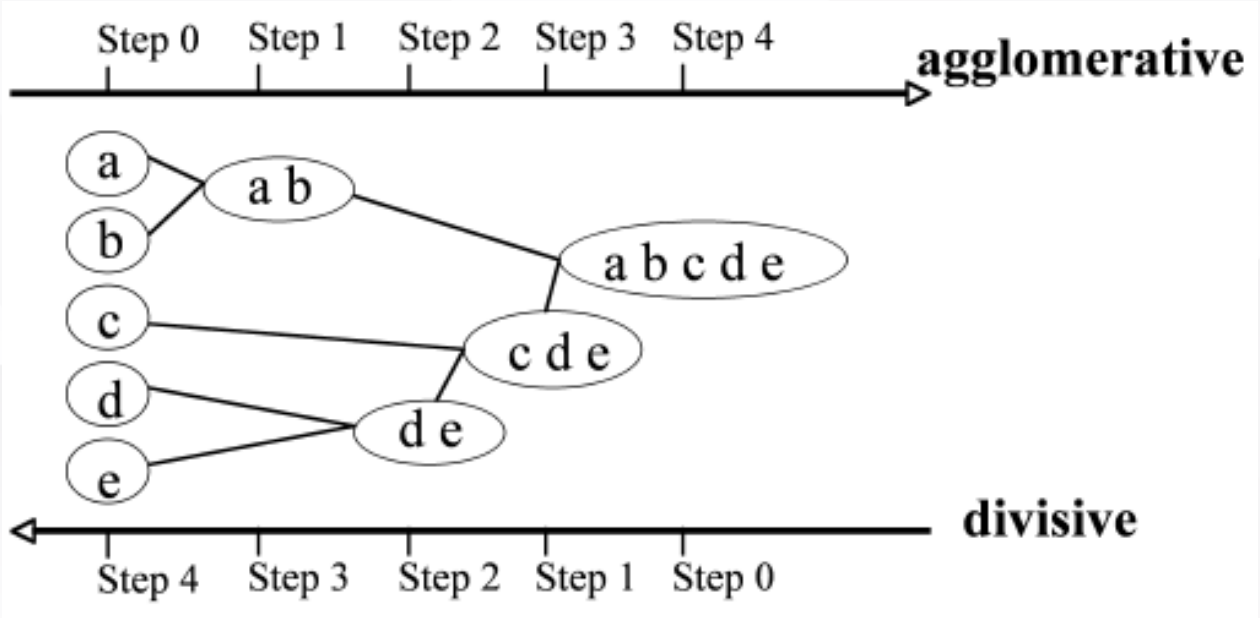


层次聚类的优点

- 不需要提前假定聚类的簇数
  - 通过选择树状图的某一层可以获得任意簇数量的聚类结构
- 聚类结果可能对应着有意义的分类体系
  - 例如在生物科学中 (e.g., 门纲目科, 人类种系, ...)

层次聚类

- 自底向上(凝聚式): 递归的合并相似度最高/距离最近的两个簇
- 自顶向下 (分列式): 递归地分裂最不一致的簇（例如：具有最大直径的簇）
- 用户可以在层次化的聚类中选择一个分割，得到一个最自然的聚类结果（例如，各个簇的簇间相似性高于一定阈值）



凝聚式聚类算法

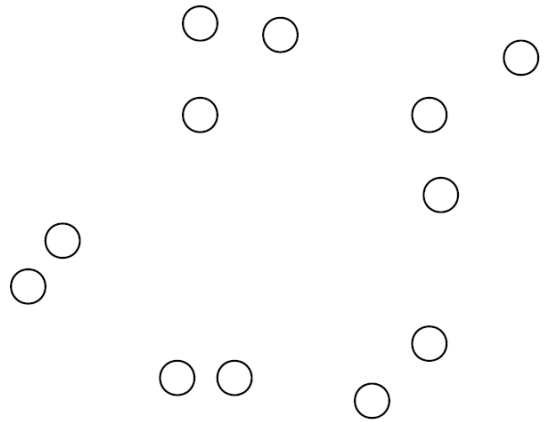
- 相较于分列式，凝聚式是更加流行的层次聚类技术
- 基本算法非常直观

1. Compute the proximity matrix
2. Let each data point be a cluster
3. Repeat
4. Merge the two closest clusters
5. Update the proximity matrix
6. Until only a single cluster remains

- 关键是如何计算簇之间的近似程度(proximity) →不同的定义簇间距离的方法，将得到不同的聚类算法

起始状态

- 开始时每个点是一个簇，计算一个相似度矩阵

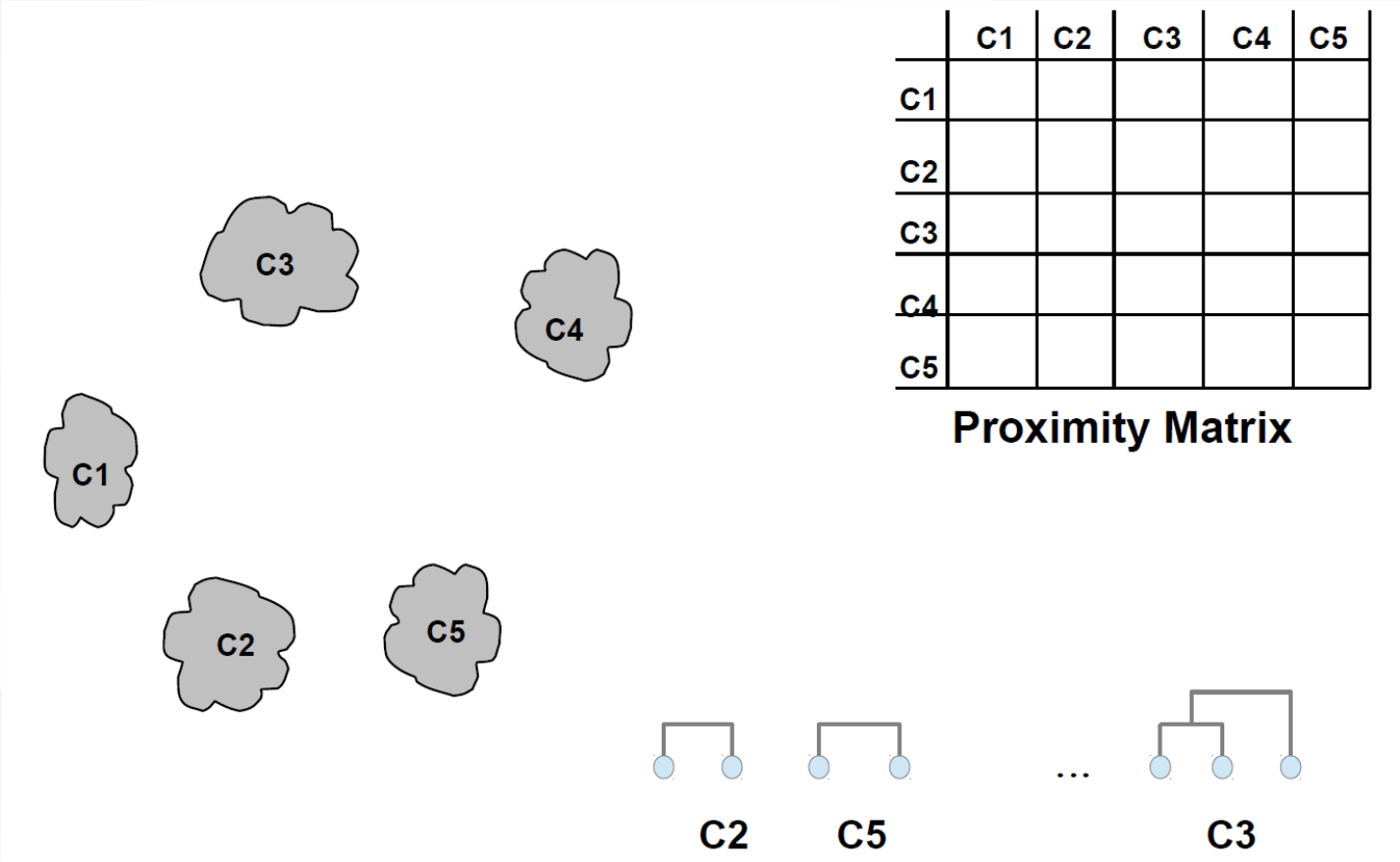


	p1	p2	p3	p4	p5	. .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

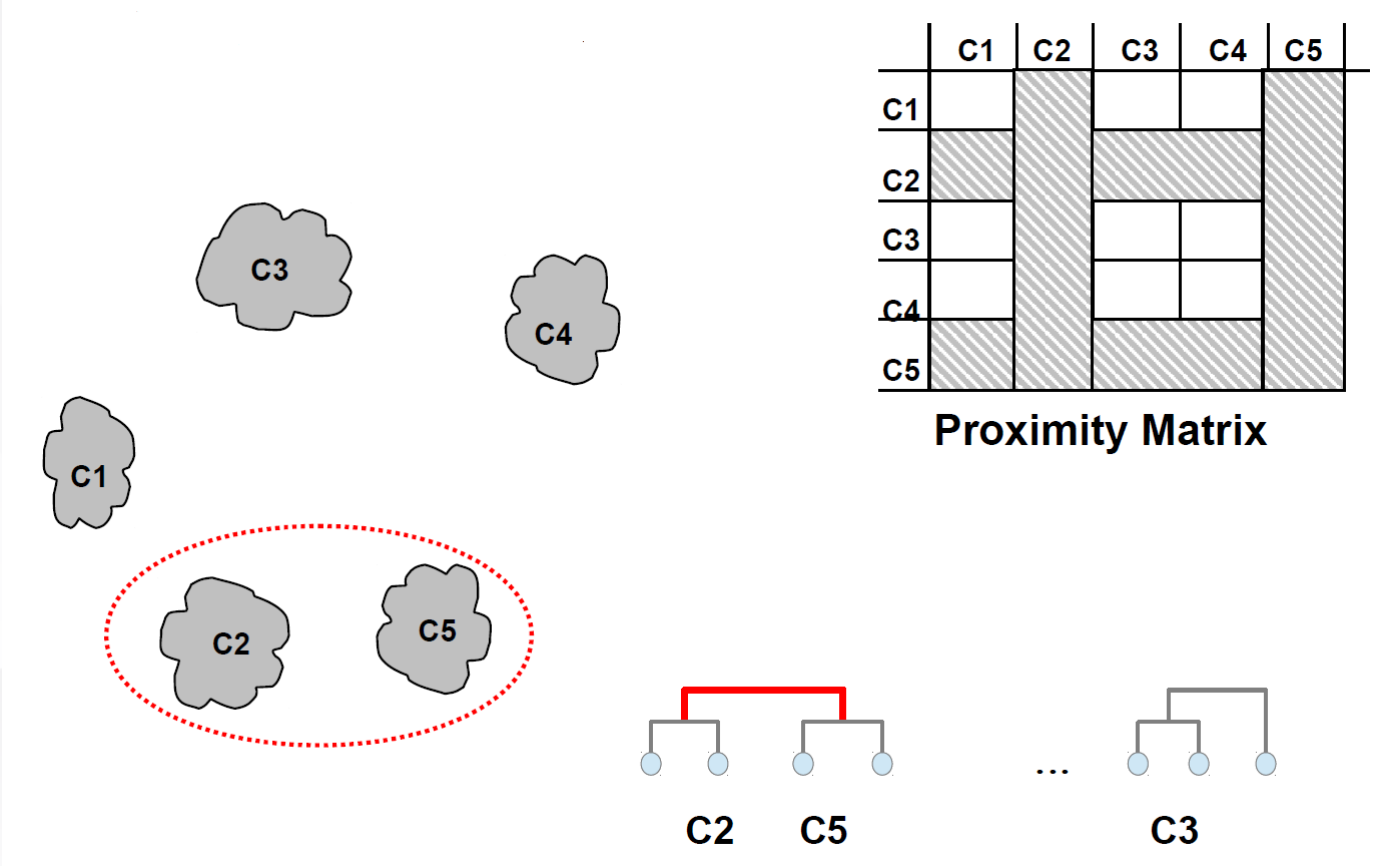
中间过程

- 经过一些合并步骤后，我们可以获得一些簇



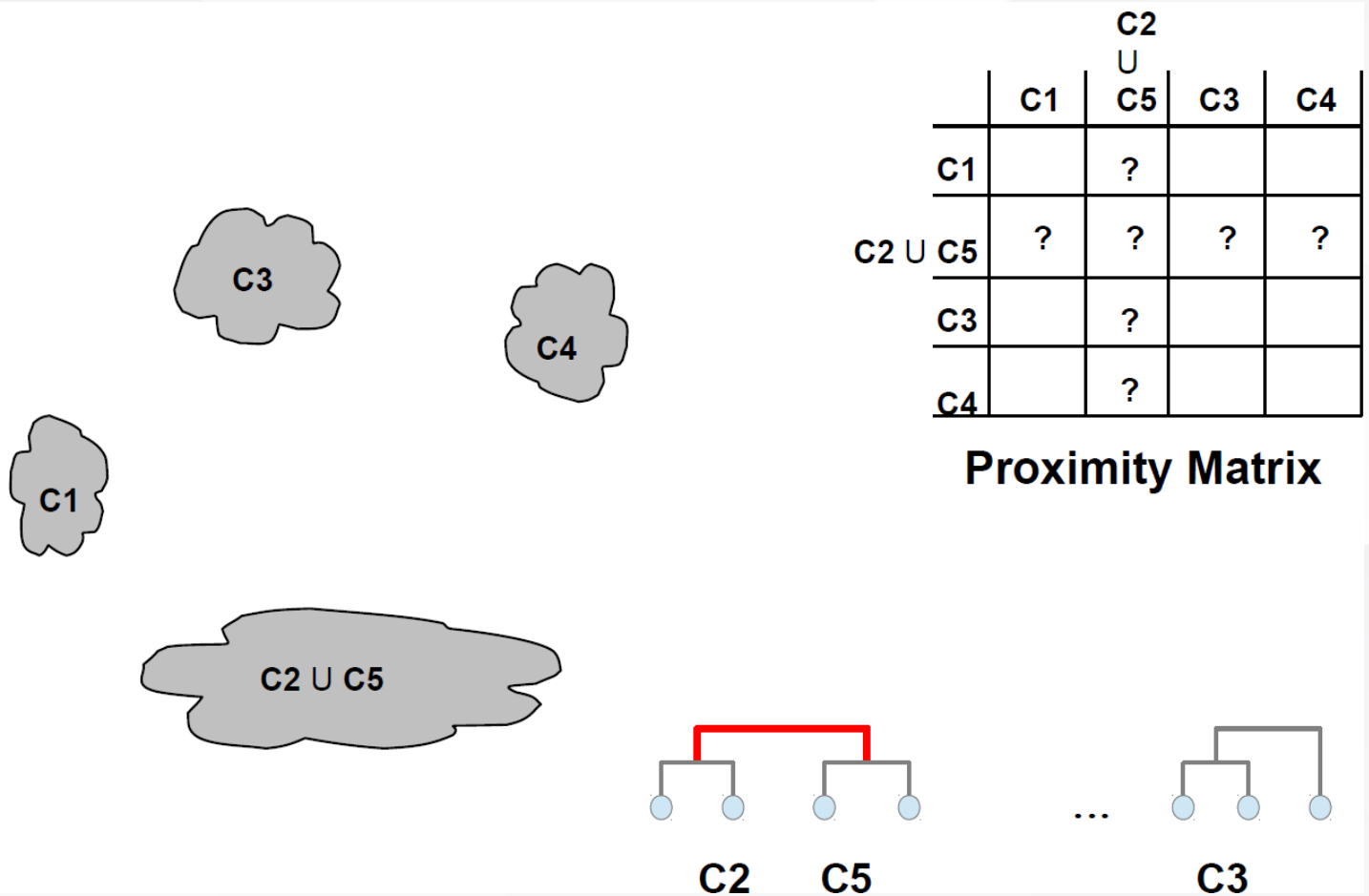
中间过程

- 我们合并最近的两个簇 (C2 和 C5) 并更新相似度矩阵



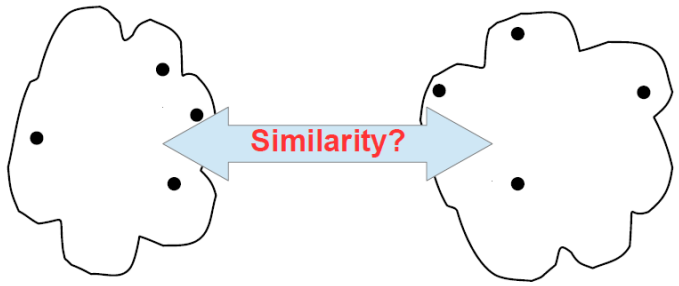
合并后

- 问题在于 “我们如何更新相似度矩阵?”



如何定义簇间相似性

- 最小距离(MIN)
- 最大距离(MAX)
- 平均距离(Group Average)
- 中心点距离(Distance Between Centroids)
- 其他由某种目标函数推导出来的方法
  - Ward’s 方法使用平方误差



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

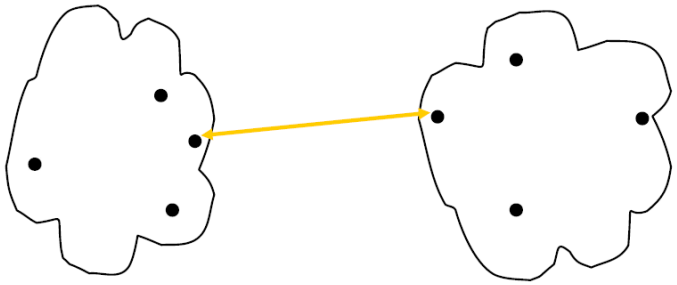
Proximity Matrix

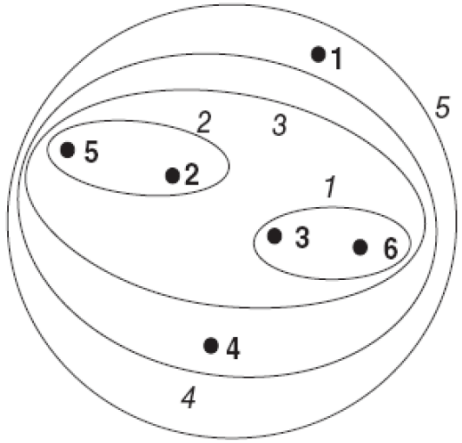
如何定义簇间相似性

- 最小距离(MIN) (Single link)
- 最大距离(MAX)
- 平均距离(Group Average)
- 中心点距离(Distance Between Centroids)
- 其他由某种目标函数推导出来的方法
  - Ward’s 方法使用平方误差

优势: 可形成非球形、非凸的簇

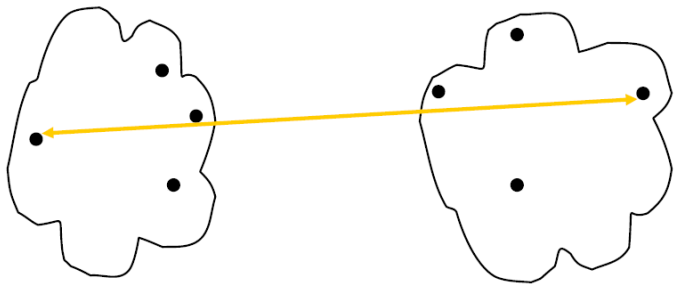
问题: 链式效应

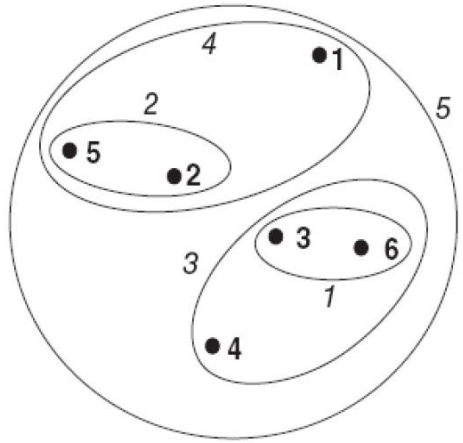




如何定义簇间相似性

- 最小距离(MIN)
- 最大距离(MAX) (complete link)
- 平均距离(Group Average)
- 中心点距离(Distance Between Centroids)
- 其他由某种目标函数推导出来的方法
  - Ward's 方法使用平方误差





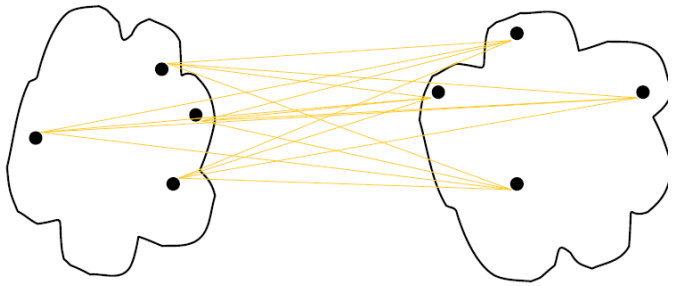
优势: 对噪声更加鲁棒 (不成链)

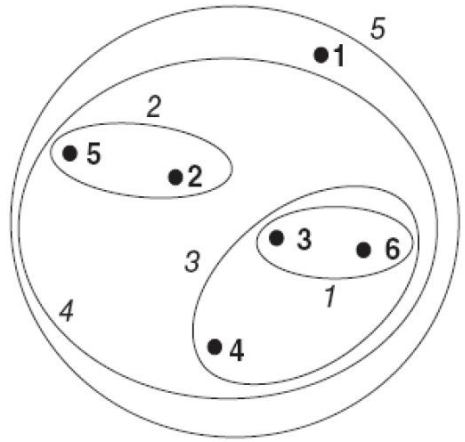
问题: 趋向于拆开大的簇,偏好球形(globular/round)簇

如何定义簇间相似性

- 最小距离(MIN)
- 最大距离(MAX)
- 平均距离(Group Average)(average-linkage)
- 中心点距离(Distance Between Centroids)
- 其他由某种目标函数推导出来的方法
  - Ward's 方法使用平方误差

MIN 和 MAX 的折中方案

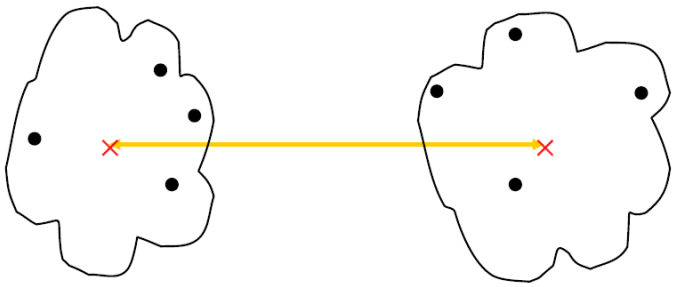




如何定义簇间相似性

- 最小距离(MIN)
- 最大距离(MAX)
- 平均距离(Group Average)
- 中心点距离(Distance Between Centroids)
- 其他由某种目标函数推导出来的方法
  - Ward's 方法使用平方误差

问题: 反向效应 (后边合并的簇间距离可能比之前合并的簇间距离更近)



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

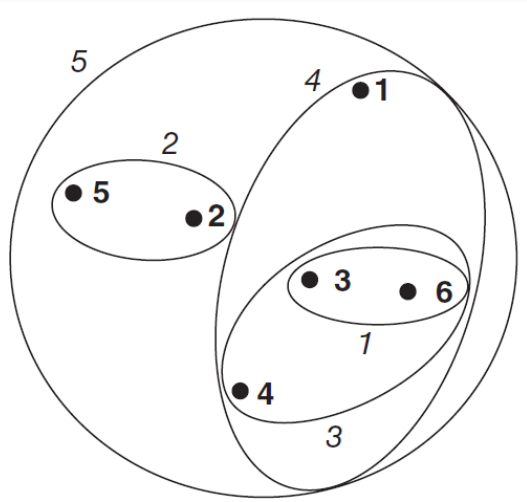
Proximity Matrix

如何定义簇间相似性

- 最小距离(MIN)
- 最大距离(MAX)
- 平均距离(Group Average)
- 中心点距离(Distance Between Centroids)
- 其他由某种目标函数推导出来的方法
  - Ward's 方法使用平方误差

两个簇的相似性基于两个簇融合后平方误差的增加

- 更少受噪声和离群点影响
- 倾向于球形簇
- K-means的层次化版本
  - 可以用于初始化K-means



层次聚类的限制

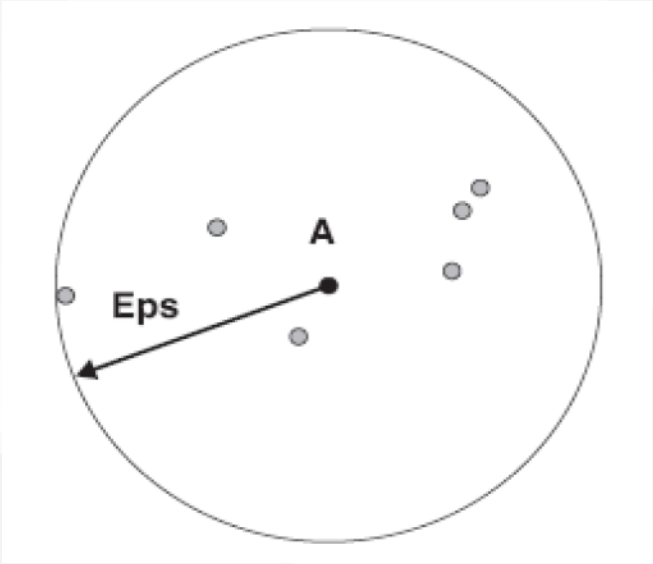
- 贪心: 一旦簇被合并或者拆分，过程不可逆
- 没有优化一个全局的目标函数
- 不同方法存在一个或多个以下问题:
  - 对噪声和离群点敏感
  - 比较难处理不同尺寸的簇和凸的簇
  - 成链, 误把大簇分裂

大纲

- 简介
- 距离函数
- 评价指标
- 聚类算法
  - K均值聚类
  - 混合高斯模型和EM算法
  - 层次聚类
  - 基于密度的聚类

DBSCAN

- DBSCAN (density-based spatial clustering of application with Noise)[1]
- **密度(Density)** = 给定半径 (ε)内点的个数

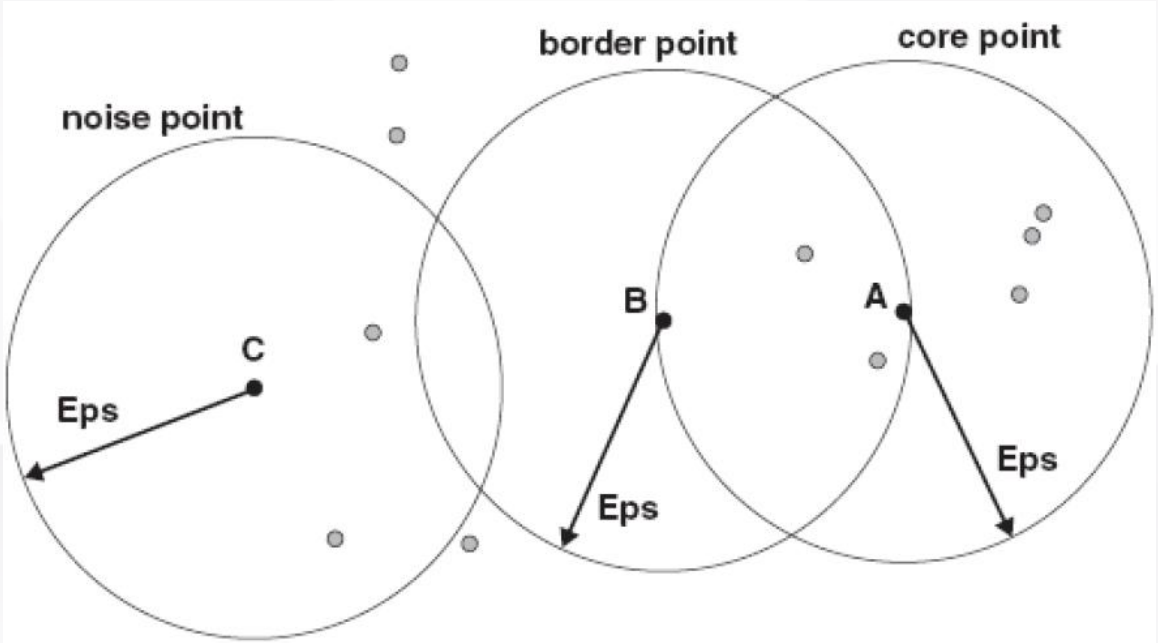


Density = 7 points

1Ester et al. A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD). 1996.

预备知识

- **核心点(Core point)**: 指定半径ε内多于指定数量MinPts个点
- **边界点(Border point)**:半径ε内有少于MinPts 个点，但是其在某个核心点的邻域内
- **噪声点 (Outliers)**: 核心点和边界点之外的点.



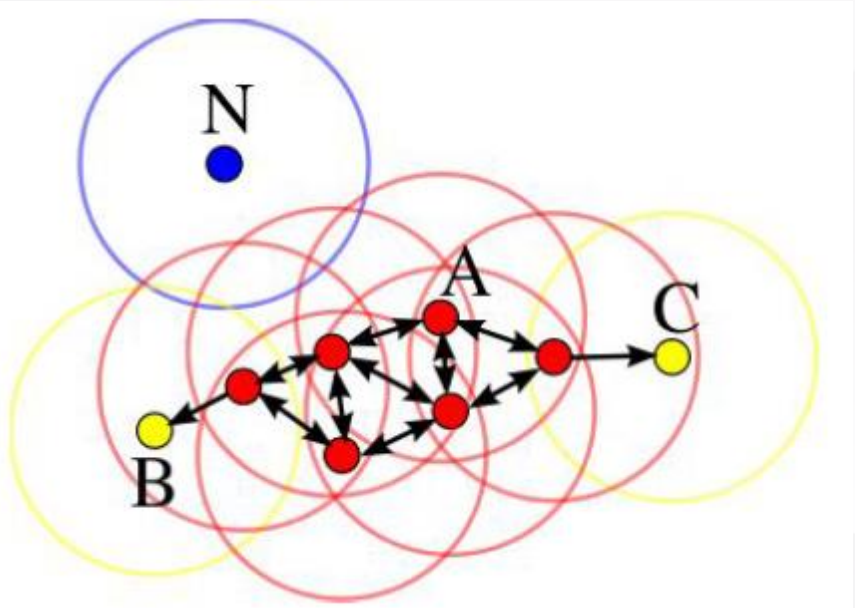
预备知识

- 点q 由点p 密度可达: 连接两个点的路径上所有的点都是核心点
  - 如果p 是核心点，那么由它密度可达的点形成一个簇
- 点q 和点p 是密度相连的，如果存在点o 从其密度可达点q 和点p
- 聚类的簇满足以下两个性质:
  - 连接性：簇内的任意两点点是密度相连的;
  - 最大性：如果一个点从一个簇中的任意一点密度可达，那么该点属于该簇

DBSCAN 算法

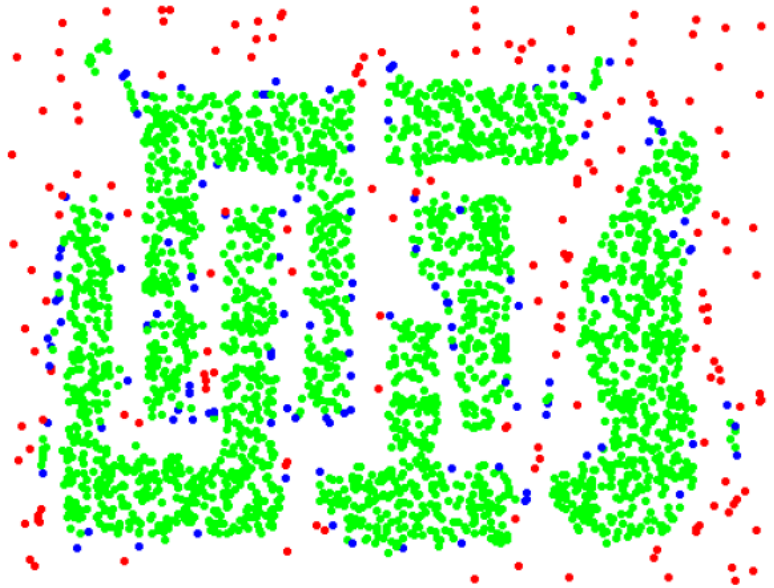
```
DBSCAN(D, eps, MinPts)
   C = 0
   for each unvisited point P in dataset D
      mark P as visited
      NeighborPts = regionQuery(P, eps)
      if sizeof(NeighborPts) < MinPts
         mark P as NOISE
      else
         C = next cluster
         expandCluster(P, NeighborPts, C, eps, MinPts)

expandCluster(P, NeighborPts, C, eps, MinPts)
   add P to cluster C
   for each point P' in NeighborPts
      if P' is not visited
         mark P' as visited
         NeighborPts' = regionQuery(P', eps)
         if sizeof(NeighborPts') >= MinPts
            NeighborPts = NeighborPts joined with NeighborPts'
      if P' is not yet member of any cluster
         add P' to cluster C
```



DBSCAN: 核心点, 边界点和噪声点

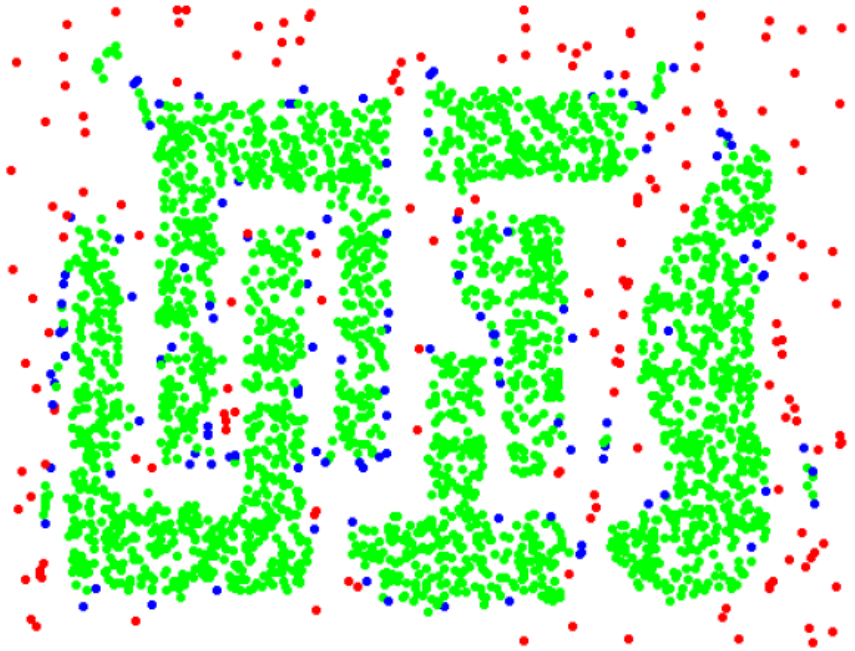
Original Points



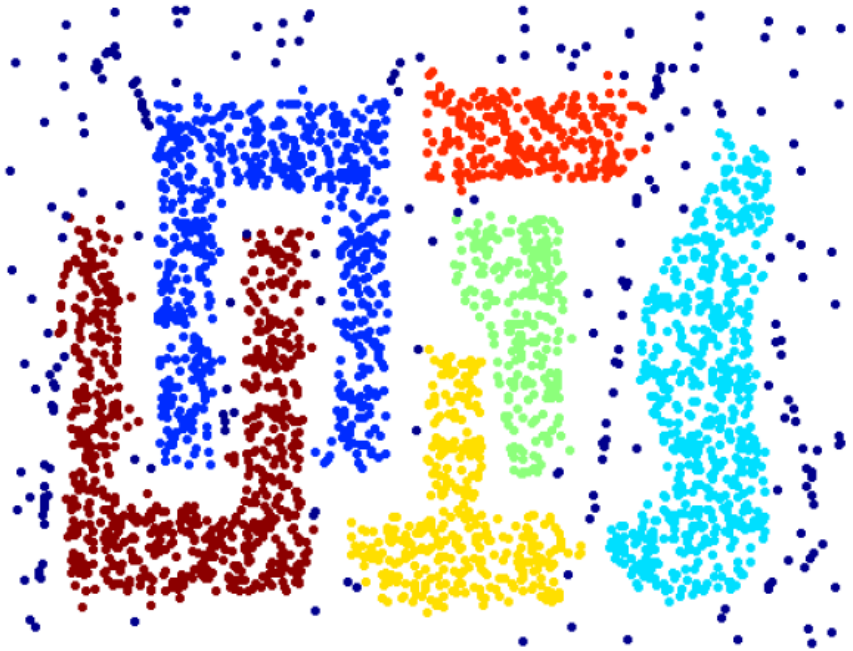
Point types: core, border and noise

Eps = 10, MinPts = 4

DBSCAN: 确定簇



Point types: core, border and noise



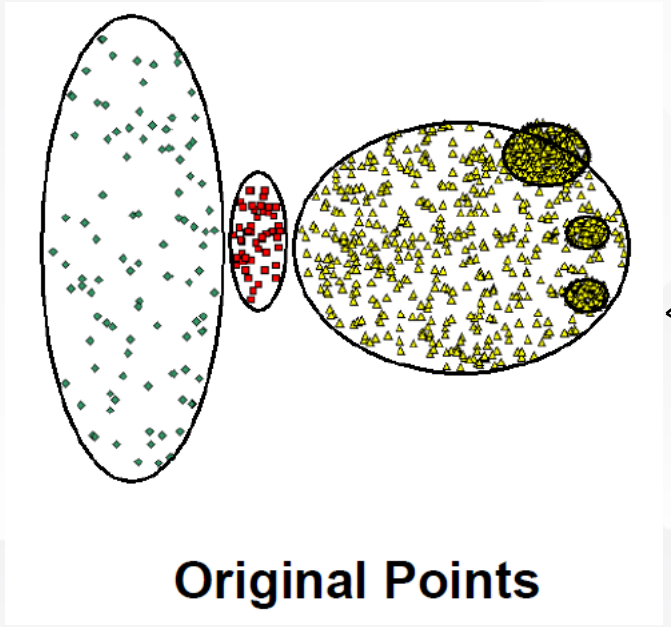
Clusters

对DBSCAN的分析

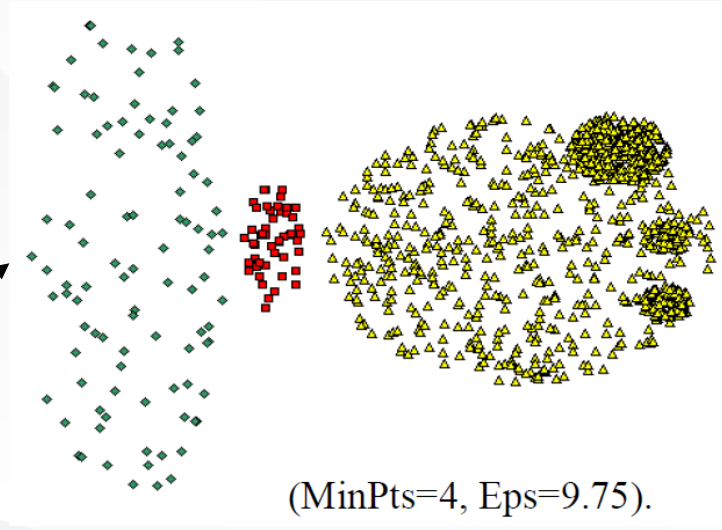
- 优势
  - 不需要明确簇的数量
  - 任性形状的簇
  - 对离群点(outliers)较为鲁棒
- 劣势
  - 参数选择比较困难(MinPts,ε)
  - 不适合密度差异较大的数据集

DBSCAN什么时候表现不好

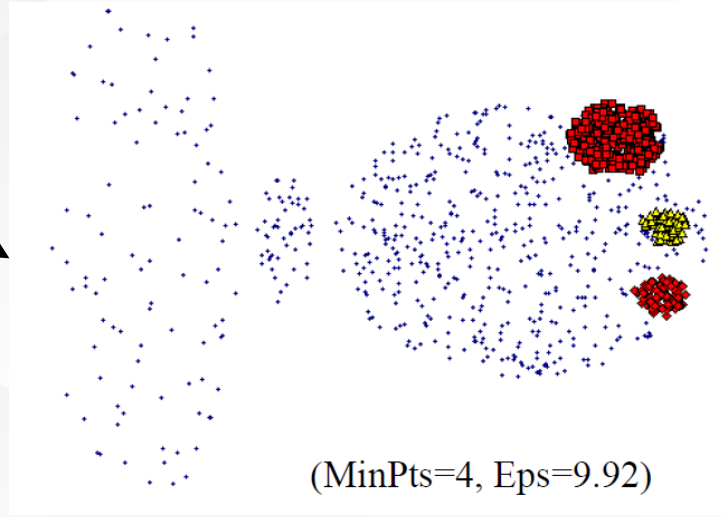
- 变化的密度
- 高维的数据



Original Points



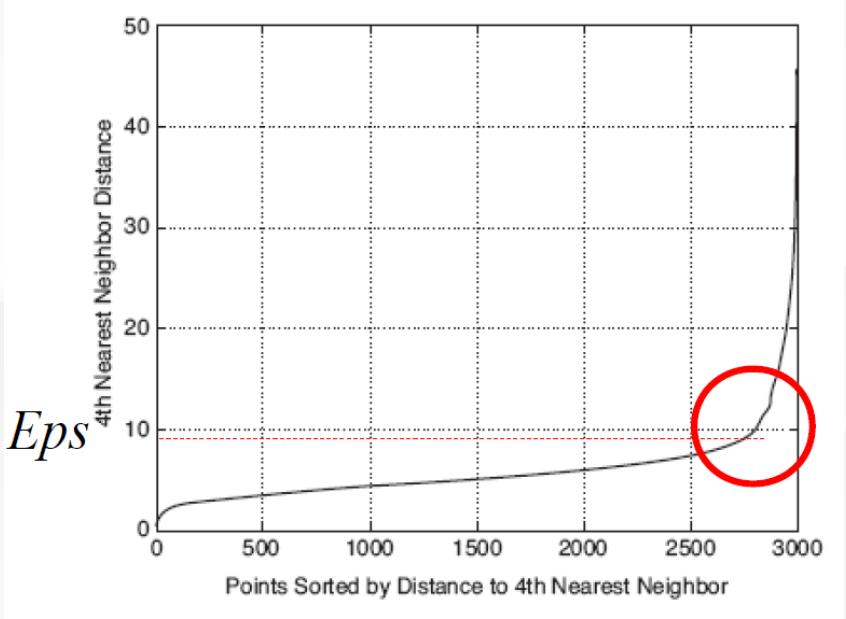
(MinPts=4, Eps=9.75).



(MinPts=4, Eps=9.92)

DBSCAN: 如何确定 EPS 和 MinPts

- 直观想法：同一个簇内的点，它们第*k*个最近邻大约相同的距离。
- 噪声点到其第*k*最近邻距离较远
- 方法：画出每个点到其第*k*最近邻的距离



MinPts = k

一些其他的聚类算法

- 基于中心的聚类
  - Fuzzy c-means
  - PAM (Partitioning Around Medoids)
- 层次化
  - CURE (Clustering Using Representatives): shrinks points toward center
  - BIRCH (balanced iterative reducing and clustering using hierarchies)
- 基于图的聚类
  - Graph partitioning on a sparsified proximity graph
  - Shared nearest-neighbor (SNN graph)
- 谱聚类
- 子空间聚类
- 数据流聚类
- 协同聚类

参考书

- 周志华，《机器学习》
- Christopher Bishop 《Pattern Recognition and Machine Learning》
- Introduction to Data Mining by Pang-Ning Tan, Michael Steinbach and Vipin Kumar

The End

A Recent Clustering Algorithm

- Rodriguez, Alex, and Alessandro Laio. Clustering by fast search and find of density peaks. Science 344.6191 (2014): 1492-1496.
- Assumption: cluster centers are surrounded by neighbors with lower local density and they are at a relatively large distance from any points with a higher local density.

The Algorithm

- Local density of point i:

$$\rho_i = \sum_j \chi(d_{ij} - d_c)$$

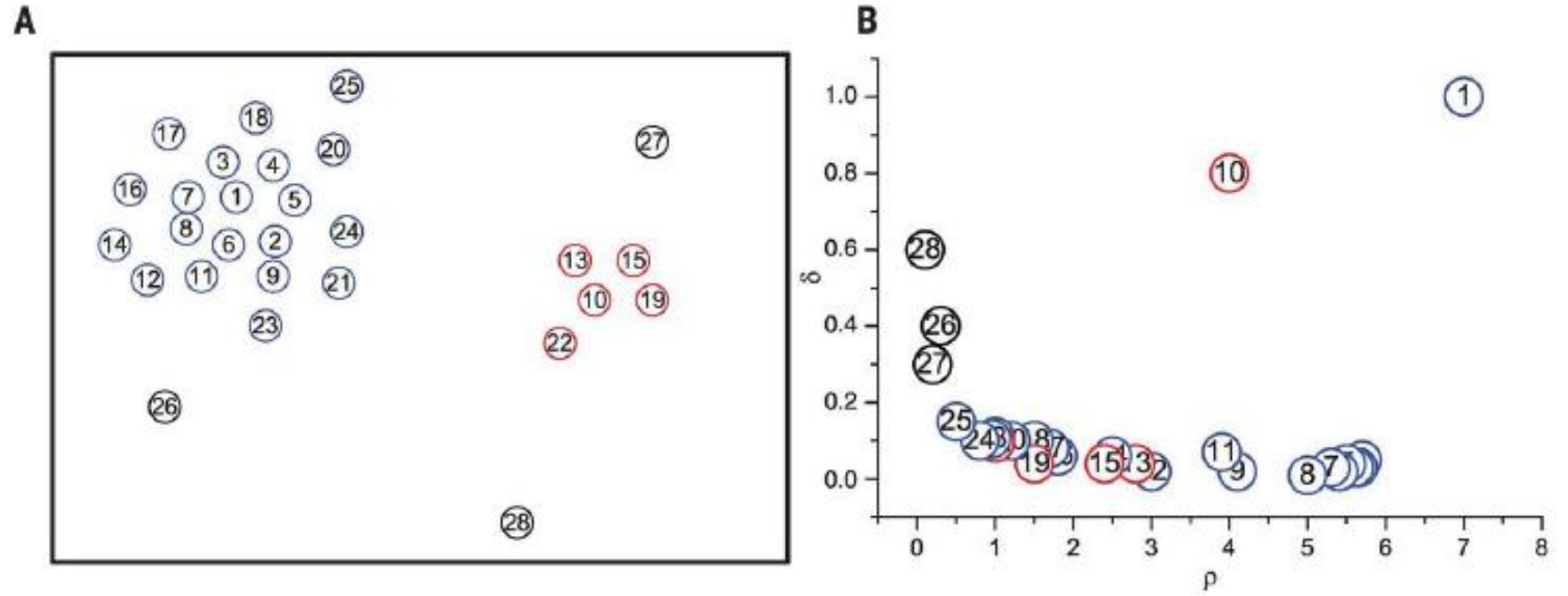
  - χ(x)=1 if x<0 and χ(x)=0 otherwise;
  - d_c is a cutoff distance (robust for large data set)
  - ρ_i is equal to the number of points closer than d_c to points i

- Its distance δ_i from points of higher density

$$\delta_i = \min_{j:\rho_j>\rho_i} d_{ij}$$

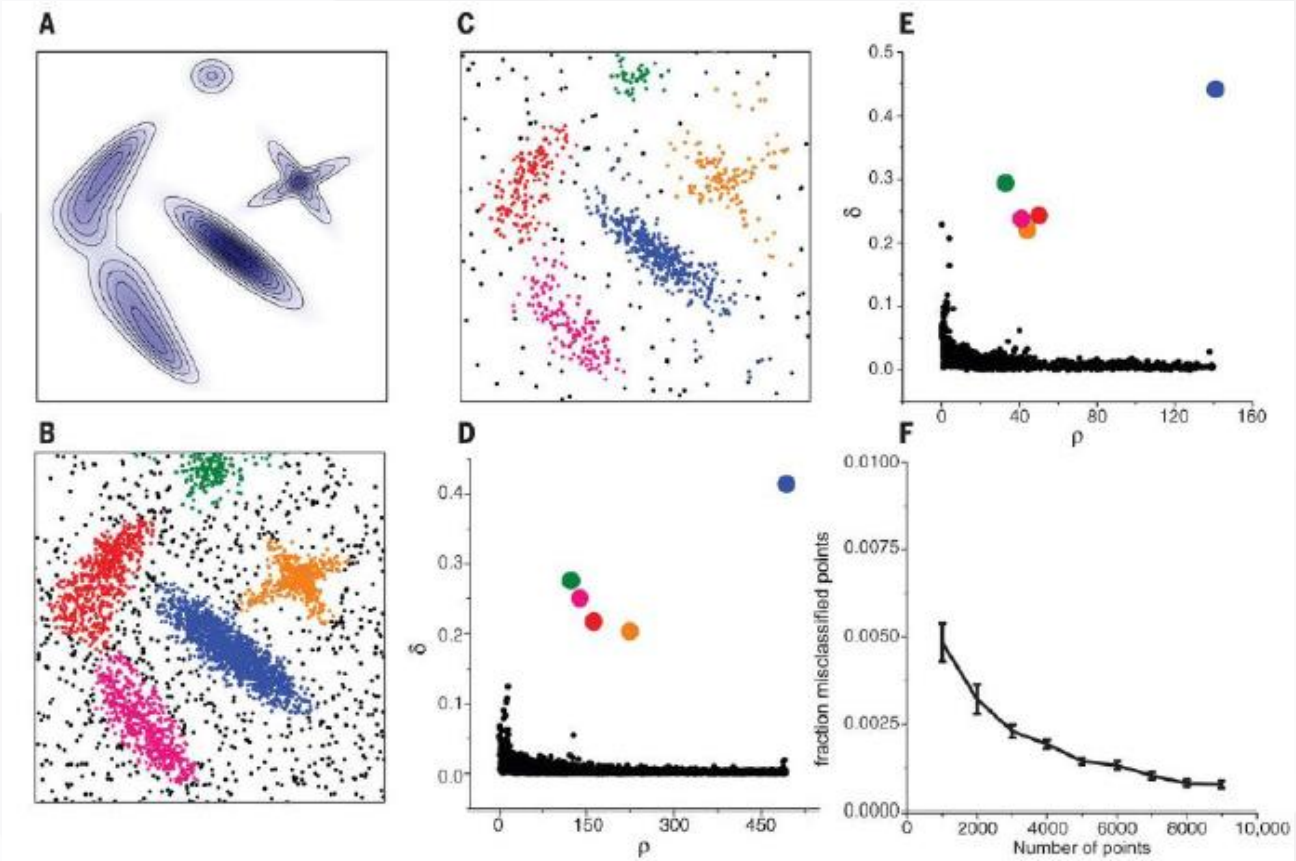
  - For the point with highest density, $\delta_i = \max_j d_{ij}$
  - δ_i is much larger than the typical nearest neighbor distance only for point with local or global maximum density
  - Cluster centers recognized as points for which the value of δ_i is anomalously large

Algorithm Illustration



- (A) point distribution. Data points are ranked in order of decreasing density.
  (B) Decision graph. Different colors corresponds to difference clusters.

Experimental Results



(A) probability distribution; (B) and (C) sample distributions for 4000 and 1000 points; (D) and (E) decision graphs; (F) the fraction of points assigned to incorrect cluster as a function of the sample dimension

Experimental Results