聚类分析

张伟平

zwp@ustc.edu.cn

Office: 东区管理科研楼 1006

Phone: 63600565

课件 http://staff.ustc.edu.cn/~zwp/

论坛 http://fisher.stat.ustc.edu.cn

简介

1.1 简介 1
1.2 距离与相异性度量 6
1.3 聚类方法 11
1.3.1 系统聚类法 12
1.3.2 K-means 19
1.3.3 谱聚类 24
1.4 确定类的数目 30
1.5 聚类质量的评价 38

1.1 简介

- 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。
- 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组构成，因而是一种无监督的学习。
- 同义词：data segmentation（数据挖掘领域)、class discovery (机器学习领域)。
- 应用领域包括经济领域，生物领域，数据挖掘等等
- 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

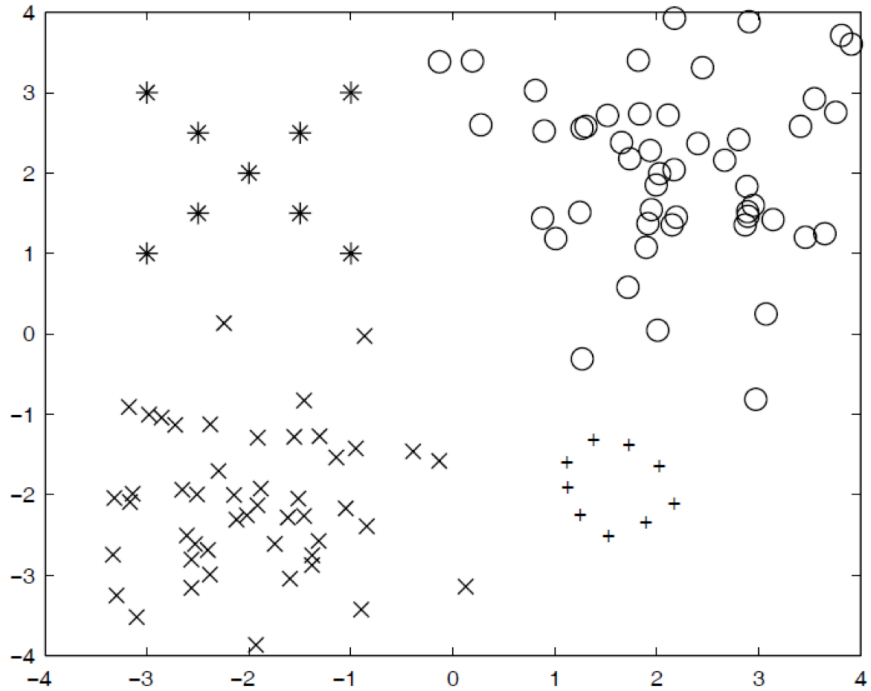
相关的研究领域

- 数据挖掘: 各种各种复杂形状类的识别，高维聚类等
- 统计学: 主要集中在基于距离的聚类分析，发现球状类
- 机器学习: 无指导学习（聚类不依赖预先定义的类)
- 其他领域: 空间数据技术, 生物学, 市场营销学

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，不同的聚类方法多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”。

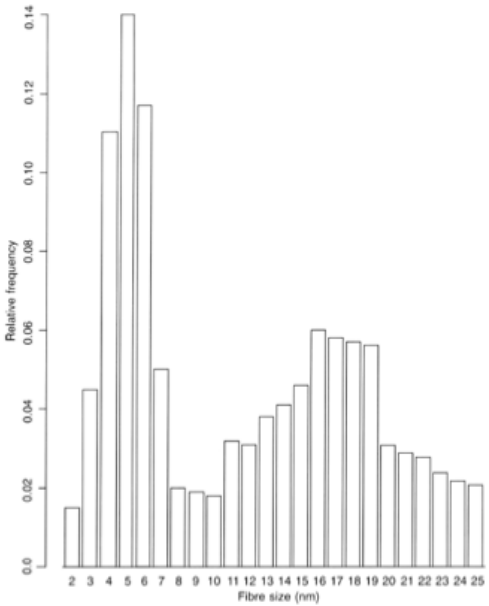
- 在 2D 或 3D 散点图中，我们很容易的发现数据中的类。
- 对发现的类我们经常赋予我们认为“应该”会存在的结构或者意义。
- 必须注意：**“类”可能仅仅是一个聚类方法的结果**
- 一个“类”依赖于如何定义它以及应用背景

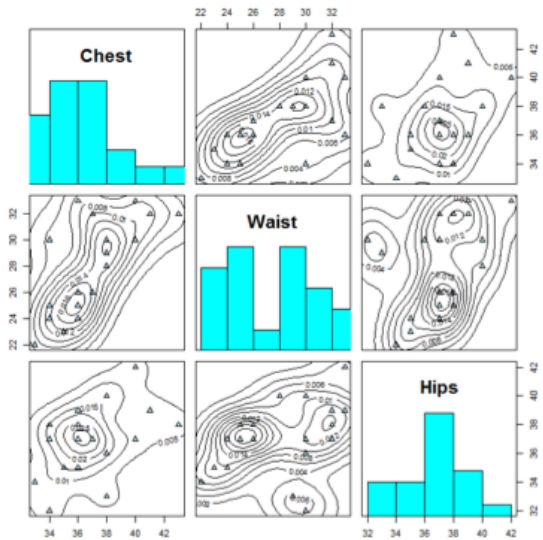


聚类与分类(clustering and classification)

- 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
  - 无训练样本，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

- **使用可视化工具探测类**
  - 多峰性是不同类存在的标志
  - 多种可视化技术可以使用：PCA, FA, MDS, Manifest learning, SOM 等等.





1.2 距离与相异性度量

- 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 $s(\cdot,\cdot)$ 一般满足

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

- 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

1. $d(\mathbf{x}, \mathbf{y}) \geq 0$, 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. $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y})$

metric dissimilarity

5. $d(\mathbf{x}, \mathbf{y}) \leq \max\{d(\mathbf{x}, \mathbf{z}), d(\mathbf{z}, \mathbf{y})\}$

如果还满足第 5 条, 则称 d 为**ultrametric dissimilarity**

- **样本点之间的相异性** 记 $\mathbf{x}, \mathbf{y} \in R^p$ 为两个样本点, 则距离的选择非常重要, 最好的距离准则往往要基于经验, 知识和运气等得到.
- 一般要根据数据的类型选择合适的相异性 (距离) 度量准则.
  - 比例尺度 (区间尺度) 下的样本数据点常用距离准则

Minkowski:

$$d_m(\mathbf{x},\mathbf{y}) = \Big[\sum_{k=1}^{p} |x_k - y_k|^m\Big]^{1/m} = \|\mathbf{x}-\mathbf{y}\|_m$$

Manhattan: (city-block distance, box-car distance)

$$d(\mathbf{x},\mathbf{y}) = \sum_{k=1}^{p} |x_k - y_k| = \|\mathbf{x}-\mathbf{y}\|_1$$

Euclidean:

$$d(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|_2$$

maximum:(Chebyshev distance)

$$d(\mathbf{x},\mathbf{y}) = \max |x_i - y_i| = \|\mathbf{x}-\mathbf{y}\|_\infty$$

Canberra:(非负量)

$$d(\mathbf{x},\mathbf{y}) = \sum_{k=1}^{p} \frac{|x_k - y_k|}{x_k + y_k}$$

– 0-1 型变量: 若 $\mathbf{x}, \mathbf{y}$ 的元素均非零即 1, 则

$\mathbf{y}$ \ $\mathbf{x}$	1	0	行和
1	a	b	a+b
0	c	d	c+d
列和	a+c	b+d	n=a+b+c+d

$$binary(Jaccard): \quad d(\mathbf{x}, \mathbf{y}) = \frac{b+c}{a+b+c} \leftarrow \textbf{no 0-0 match}$$

$$Czekanowski: \quad d(\mathbf{x}, \mathbf{y}) = \frac{b+c}{2a+b+c} \leftarrow \textbf{no 0-0 match, double 1-1 match}$$

其他见课本表 12.1.

– 属性变量: 若 $\mathbf{x}, \mathbf{y}$ 为属性变量, 各有 p 和 q 个不同的类别, 则度量两者之间的相似性常常基于列联表度量性检验

中的 χ^2 统计量进行.

$$\text{Coef. of contingency} \quad s_{ij} = \left(\frac{\chi^2}{\chi^2 + n}\right)^{1/2}$$

$$\text{Cramer's V contingency coef.} \quad s_{ij} = \left(\frac{\chi^2}{n \min\{p-1, q-1\}}\right)^{1/2}$$

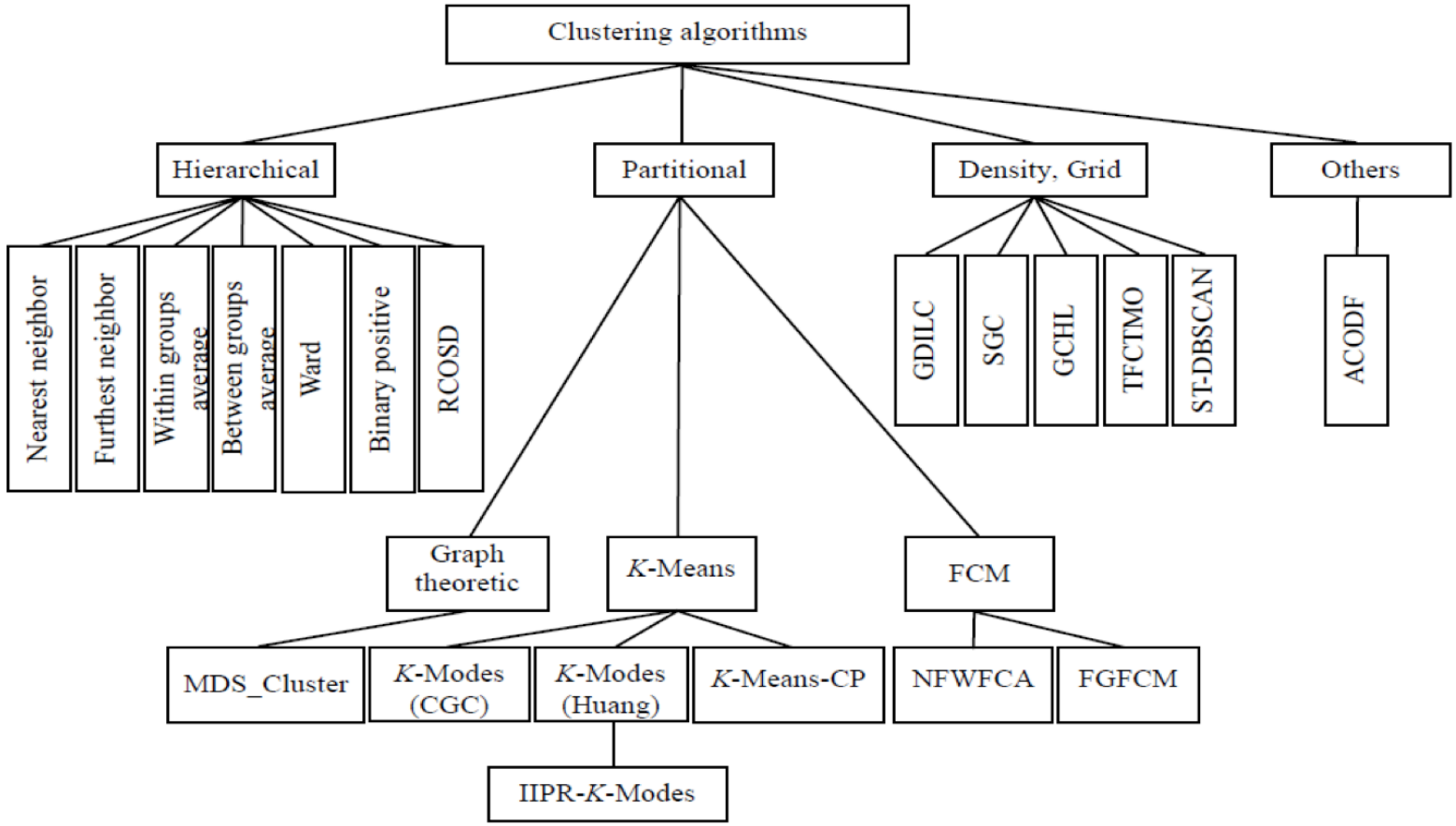
- **变量之间的相似性** 样本相关系数常常用来度量变量之间的相似性, 常常使用相关系数的绝对值度量变量之间的相似性. $\mathbf{x}$ 的第 i 个分量 X_i 和第 j 个分量 X_j 之间的相似性:

$$\text{样本相关系数 } r_{ij} = \frac{\sum_{k=1}^{n}(\mathbf{x}_{ik} - \bar{\mathbf{x}}_i)(\mathbf{x}_{jk} - \bar{\mathbf{x}}_j)}{\left[\sum_{k=1}^{n}(\mathbf{x}_{ik} - \bar{\mathbf{x}}_i)^2 \sum_{k=1}^{n}(\mathbf{x}_{jk} - \bar{\mathbf{x}}_j)^2\right]^{1/2}}$$

$$\text{夹角余弦 } \theta_{ij} = \frac{\sum_{k=1}^{n}\mathbf{x}_{ik}\mathbf{x}_{jk}}{\left[\sum_{k=1}^{n}\mathbf{x}_{ik}^2 \sum_{k=1}^{n}\mathbf{x}_{jk}^2\right]^{1/2}}$$

1.3 聚类方法

常见的聚类方法包括



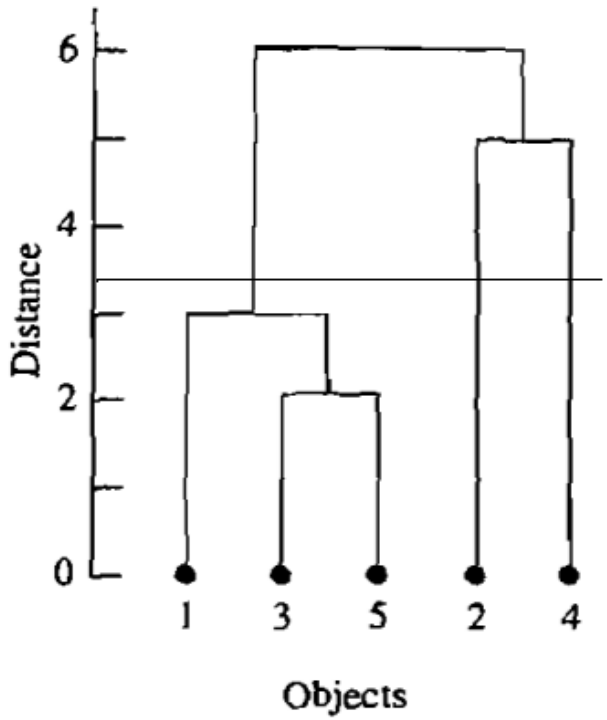
1.3.1 系统聚类法

- 系统聚类法 (Hierarchical clustering, 也称层次聚类法) 是最经典和常用的聚类方法之一.
- 系统聚类法需要度量样本点之间的距离 (dissimilarity) 和类与类之间的联接 (linkage) 程度
- 系统聚类法包括两种
  - **聚合方法**(agglomerative hierarchical method): (自下而上) 一开始将每个样本个体作为单独的一类，然后根据类间联接程度，合并相近的类，直到所有的类合并成一个类
  - **分裂方法**(divisive hierarchical method): (自上而下) 一开始将所有的样本个体置于一类，在迭代的每一步中，一个类不断地分为更小的类，直到每个样本个体单独为一个类.
- 我们主要介绍聚合聚类方法

树状图 (Dendrogram)

层次聚类的结果常常使用树状图 (dendrogram) 来表示.

- 每个节点表示一个类
- 每个叶子节点表示一个独点 (只含一个样本点的类).
- 根节点是包含了所有样本点的类
- 每个中间节点有两个子节点, 表示其通过合并这两个子类而来
- 当叶子节点调整到高度 0 时候, 则每个中间节点的高度与其两个子节点间的相异度大小成比例
- 在合适的高度上对树进行切割得到聚类结果



类间联系程度度量 (Linkage criteria)

- **single linkage**

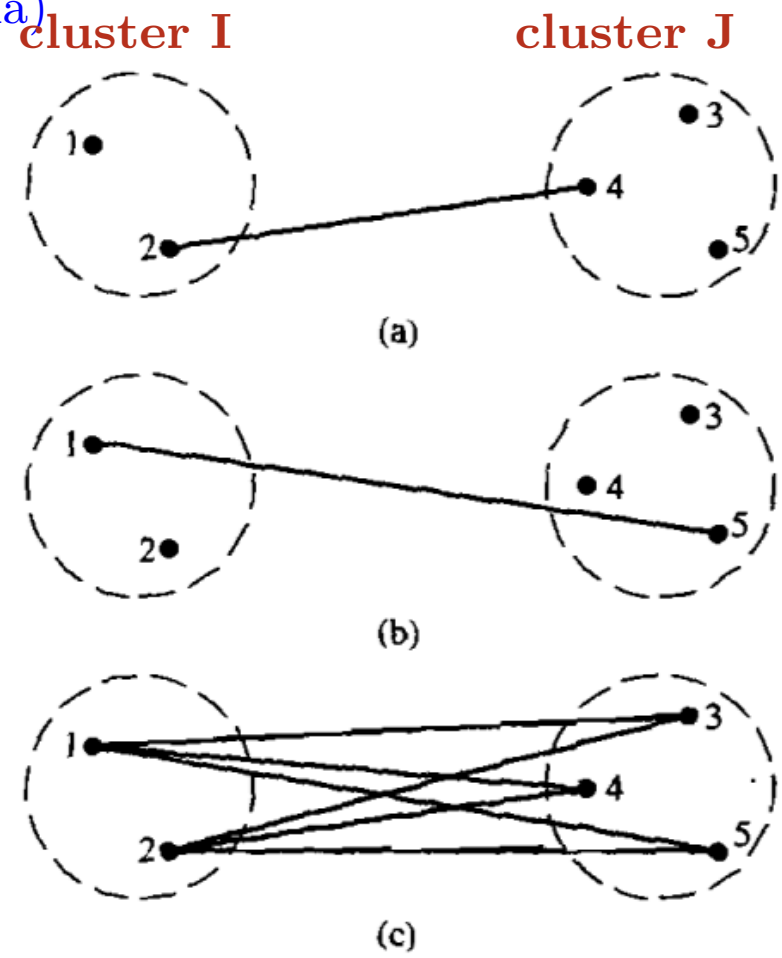
$$D(I,J) = \min_{i \in I, j \in J} \{d_{ij}\}$$

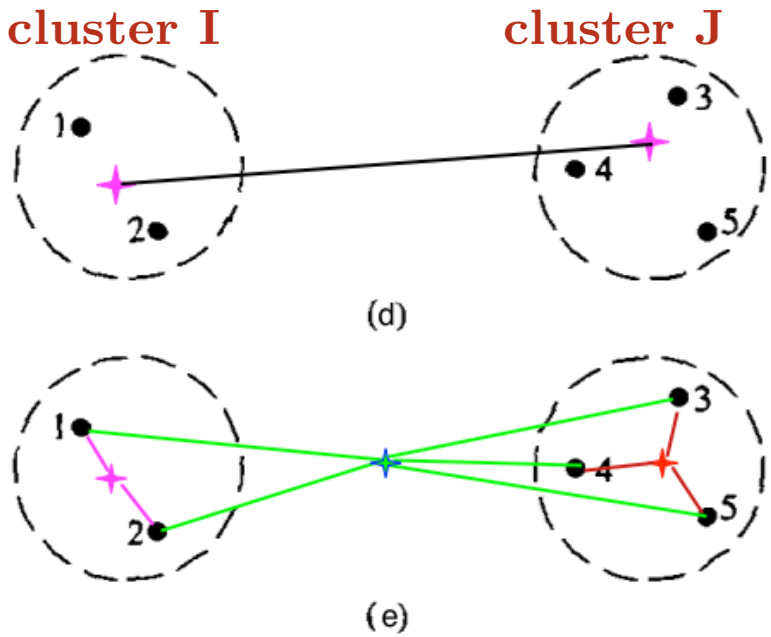
- **complete linkage**

$$D(I,J) = \max_{i \in I, j \in J} \{d_{ij}\}$$

- **average linkage**

$$D(I,J) = \frac{1}{n_I + n_J} \sum_{i \in I, j \in J} d_{ij}$$





- **centroid linkage**:

$$D(I, J) = \|\bar{\mathbf{x}}_I - \bar{\mathbf{x}}_J\|$$

- **ward method**: 分别计算类 I 和 J 内各点到其重心 (均值) 的平方欧式距离和 (称为离差平方和), 分别记为 W_I 和 W_J; 然后将所有点合并为一个类 M, 计算其离差平方和 W_M, 最后定义类 I 和 J 之间的平方距离为

$$D(I, J) = W_M - W_I - W_J$$

离差平方和法使得两个大的类倾向于有较大的距离，因而不易合并; 相反，两个小的类却因倾向于有较小的距离而易于合并。这往往符合我们对聚类的实际要求。

几种联系度量的特点:

- Single linkage 下的类有串链特点. 合并两个类时只看它们最近的点而不管其他点, 聚出的类呈现链状. 因此适合于条形甚至 S 形的类. 其聚类结果对相异度的单调变换不变.
- Complete linkage 避免出现链状类, 但是会导致过大的类. 因为其只考虑两类最相异 (远) 的点, 故受异常点 (距离别的类比所在类的点更近) 影响严重. 其聚类结果对相异度的单调变换不变.
- Average linkage 是前两者的权衡. 但是当相异度进行单调变换时候, 基于平均距离的聚类结果会发生变化.
- Centroid linkage 下的聚类树状图可能会出现逆连 (中间节点的高度在两个子节点高度之间), 切割树时难以确定类数目

聚合聚类算法

1. 输入所有样本点, 每个点为一个类
2. 计算相异度 (距离) 矩阵 (d_{ij})
3. 合并最小链接 (linkage) 的两个类
4. 计算新的类和其他所有类之间链接大小
5. 重复 3-4, 直至所有类合并为一个类或者满足停止条件
6. 输出树状图, 切割树状图得到聚类结果

聚合聚类方法较分裂式聚类方法更常用, 分裂方法由于难以指定合适分裂方法而不太常用. **cluster** 包里的 **diana**函数使用分裂式层次聚类方法 (Kaufman and Rousseeuw, 1990).

系统聚类方法的特点

- 无需事先指定类的数目
- 需要确定相异度和联接度量准则
- 运算量较大, 适用于不太大规模的数据
- 一旦一个步骤 (合并或分裂) 完成, 就不能撤销或修正, 因此产生了改进的层次聚类方法, 如 BRICH, BURE, ROCK, Chameleon 等.

1.3.2 K-means

- 流行和经典的聚类方法之一, 比层次聚类法运算量小, 适用于小到中大规模样本数据
- 需要事先指定聚类的数目 k
- **思想:** 随机选择 k 个样本点，每个样本点初始地代表一个类的平均值或中心, 对剩余每个样本点, 根据其到类中心的距离, 被划分到最近的类; 然后重新计算每个类的平均值. 不断重复这个过程, 直到所有的样本都不能再分配为止.
- 适用于发现球状类
- 不同的初始值, 结果可能不同
- 有些 K-means 算法的结果和数据输入顺序有关
- 可能会陷入局部解

K-means 算法

MacQueen (1967) 提出的经典算法. 核心想法是: 找出 k 个类, 使得每个数据点到与其最近的聚类中心的平方欧式距离和最小. 算法如下

1. 输入数据和聚类数目 k
2. 执行下面二者之一
   - 随机将数据分为 k 个类 $C_1, \ldots, C_k$, 计算每个类的中心 $\bar{\mathbf{x}}_i, i = 1, \ldots, k$
   - 指定 k 个类的中心 $\mathbf{x}_i, i = 1, \ldots, k$, 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

$$ESS = \sum_{i=1}^{k} \sum_{j \in C_i} \|\mathbf{x}_j - \bar{\mathbf{x}}_i\|^2$$

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 ESS 减少. 完成后重新计算各类的中心 $\bar{\mathbf{x}}_i, i = 1, \ldots, k$

5. 重复 3 和 4, 直至没有点需要进行调整 (ESS 不能减少)

K-medoids 方法是对 K-means 方法的推广: 其类似于 K-means 算法, 区别在于

- 每个类使用 "代表个体" 代替 K-means 中的类个体平均
- 可以使用相异度度量, 而不像 K-means 仅限于平方欧式距离
- 需要事先指定类的个数
- 相比 K-means 方法更加稳健 (对噪音和异常点)
- **R** base 的 **kmeans**函数使用 K-means 算法, 而 **cluster**包的 **pam**(partitioning around medoids) 函数使用 K-medoids 算法.

- K-means 和 K-medoids 算法的计算复杂度高, 适于小规模数据.
- 大规模数据可以使用 CLARA (Clustering LARge Applications, Kaufman and Rousseeuw 1990) 算法: 随机选择一部分的样本点, 使用 K-medoids 算法
- 以及改进的 CLARANS (Clustering Large Application based upon RANdomized Search, Ng and Han 1994) 算法等.

K-medoids 算法

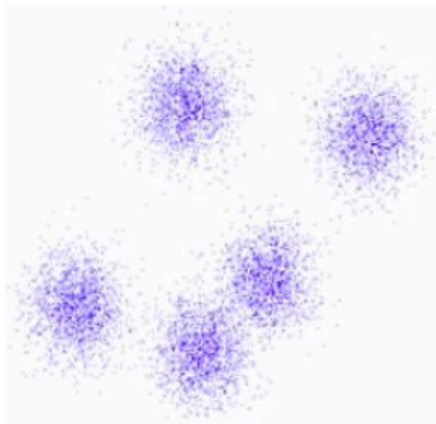
1. 输入相异度矩阵 $D = (d_{ij})$ 和聚类数目 k
2. 随机选择 k 个样本点作为类的中心点 (medoids)
3. 将每个样本点关联到和其相异度最小的中心点 (即所有样本点划分成 k 个类)
4. 对第 l 个类 $(l = 1, \ldots, k)$, 寻找类内具有最小平均相异度的点 i_0:

$$i_0 = \arg\min_{i \in C_l} \sum_{j \in C_l} d_{ij}$$

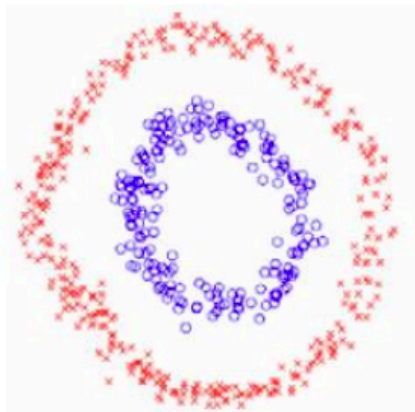
   从而得到第 l 个类的新的中心点
5. 重复 3 和 4 直至没有个体能够调整

1.3.3 谱聚类

- 两种不同的准则
  - 紧性 (compatness): k-means, hierarchical clustering
  - 相连性 (connectivity): spectral clustering



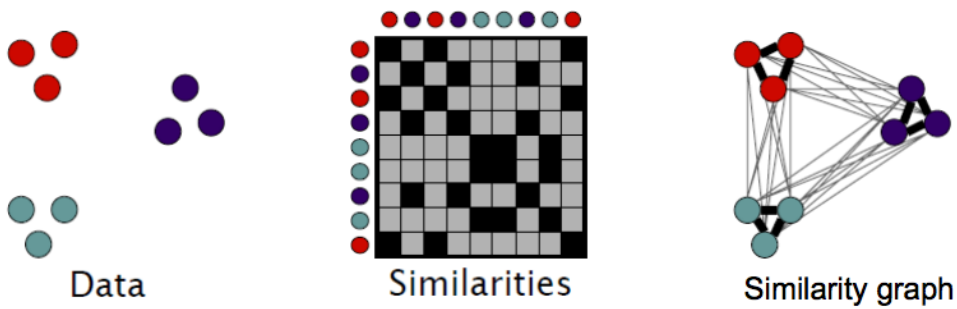
Compactness



Connectivity

谱聚类(Spectral clustering) 是现代流行聚类算法之一，较传统的 K-means 和层次聚类要好。

- 基于数据之间的相似度矩阵，使用特征向量 (谱).
- 在低维空间表达原始数据后, 在低维空间使用 K-means 进行聚类.
- 常使用无向图有权图来描述
- 给定样本点 $V = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, 计算其相似度 (图分析中称连接阵) 矩阵 W, 将数据划分为 k 个组, 使得组内的点相似, 而组间的点不相似



– 对无向图中顶点之间的权重, 常用 Gaussian kernel 建立:

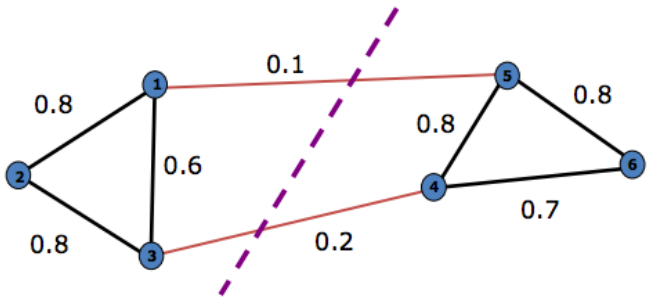
$$W_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{2\sigma^2}}, \ \sigma^2\text{控制相邻程度}$$

– 目标: 使得组内具有较高权重, 区间具有较小权重

- 记 $W(A,B) = \sum_{i\in A, j\in B} W_{ij}$ 表示两个顶点集 A 和 B 之间的相似程度. 从而寻找一个最优的划分集 $A_1, \ldots, A_k$, 使得下式最小化

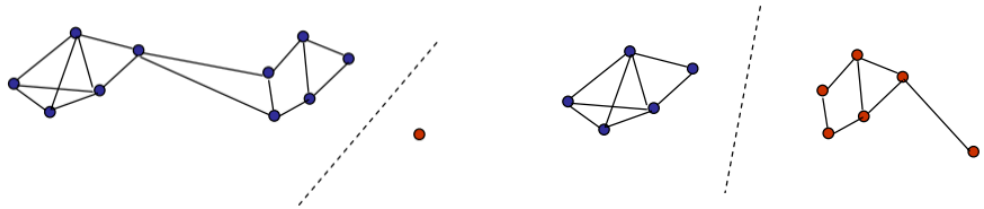
$$cut(A_1, \ldots, A_k) = \frac{1}{2}\sum_{i=1}^{k} W(A_i, \bar{A}_i)$$

其中 $\bar{A}$ 表示 A 的补集.



Minimize weight of between-group connections

- 但是这样的目标函数对异常点敏感:



What we get　　What we want

- 其中一种解决方法 (Ncut, Shi and Malik, 2000):

$$Ncut(A_1, \ldots, A_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{vol(A_i)}$$

其中 $vol(A) = \sum_{i \in A} d_i = \sum_{i \in A} \sum_{j=1}^{n} W_{ij}$.

- 识别最小的 $Ncut(A_1, \ldots, A_k)$ 是 NP-hard .
- 有一些基于线性代数的有效逼近算法
- 基于 Laplacian 矩阵, 或 Graph Laplacian: $L = D - W$, $D = diag(d_1, \ldots, d_n)$.
- 对指定的 $k(k > 2)$, 记 $h_{ij} = \frac{1}{\sqrt{vol(A_j)}}$, 如果 $\mathbf{x}_i \in A_j$, 否则为 0. $H = (h_{ij})$, 则:

$$\min_{A \subset V} Ncut(A_1, \ldots, A_k) \Longleftrightarrow \begin{cases} \min\limits_{A_1, \ldots, A_k} Trace(H'LH) \\ st. \qquad H'DH = I_k \end{cases}$$

替换$H = D^{-1/2}T$
Relaxation!!!

$$\Longrightarrow \begin{cases} \min\limits_{T \in R^{n \times k}} Trace(T'D^{-1/2}LD^{-1/2}T), \\ st. \qquad T'T = I_k \end{cases}$$

- 最优的 T_{opt} 为矩阵 $L_{sym} = D^{-1/2}LD^{-1/2}$ 的前 k 个特征向量.

– 最终的解 $H_{opt} = D^{-1/2}T_{opt}$ 为矩阵 $L_{rw} = D^{-1}L$ 的前 k 个特征根.

谱聚类算法(Shi and Malik, 2000)

给定 n 个点 $\mathbf{x}_1, \ldots, \mathbf{x}_n$, 以及聚类数目 k:

- 构建图: 计算连接矩阵 W
- 计算 Laplacian 阵 $L = D - W$
- 计算广义特征根方程 $Lu = \lambda Du$ 的前 k 个特征向量, 记为 $U = [u_1, \ldots, u_k]$
- 记 $\mathbf{y}_i$ 为 U 的第 i 行,$i = 1, \ldots, n$, 使用 K-means 算法对 $\mathbf{y}_1, \ldots, \mathbf{y}_n$ 聚类得到类 $C_1, \ldots, C_k$
- 输出类 $A_1, \ldots, A_k$, 其中 $A_i = \{\mathbf{x}_j | \mathbf{y}_j \in C_i\}$

Ng, Jordan, and Weiss (2002) 给出了另外一种类似的算法.

1.4 确定类的数目

- 有时候提前指定聚类的数目没有问题, 例如将一个客户数据根据 k 个销售员聚成 k 个类等
- 大多数情况下, 聚类的确切数目是未知的. 此时确定聚类的数目是很困难的一个问题 (高维数据难以检查, 或者难以解释类数目的合理性).
- 确定类的数目也是非常重要的, 比如确定某种癌症有两种类型还是三种类型是差异非常大的

Silhouette(侧影) 法

- Kaufman and Rousseeuw (1990) 提出的一种既可以评价每个点应该在当前所属类还是其他类, 也可以评价整体的聚类结果效果.

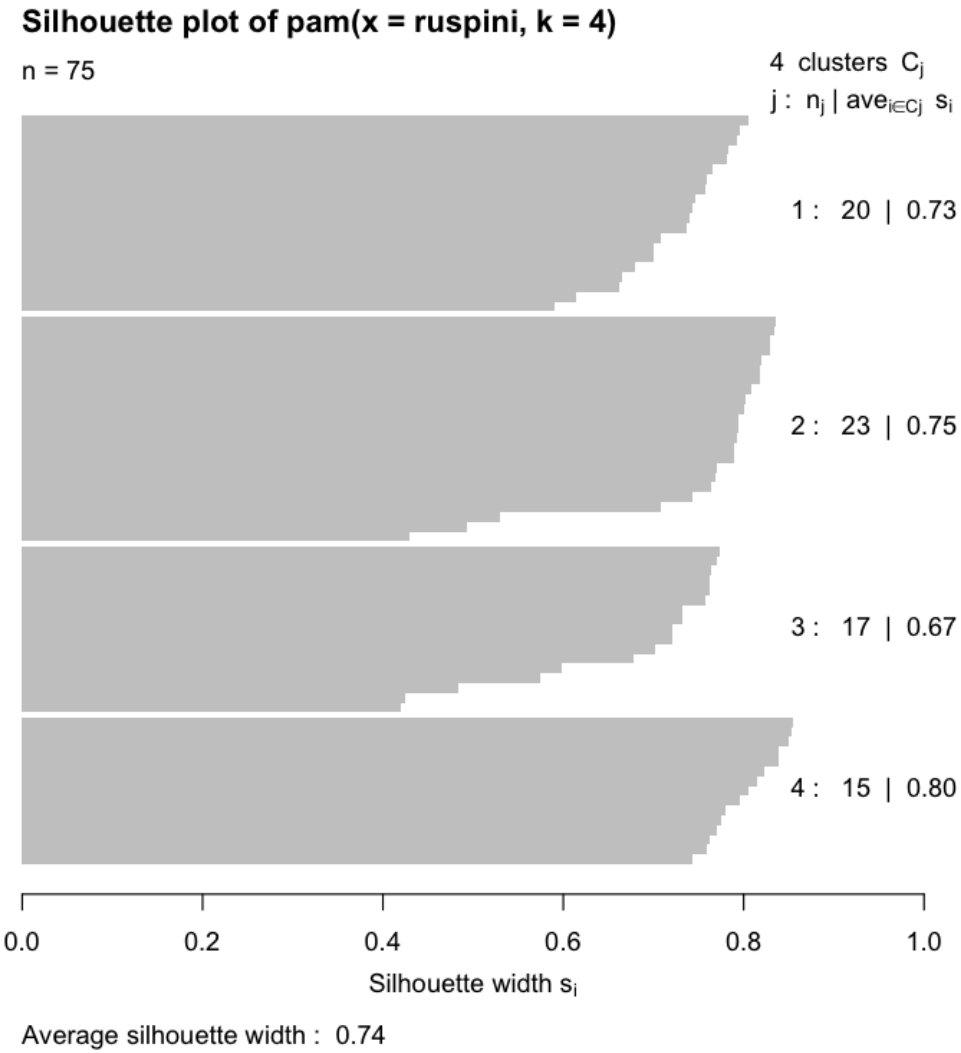
- 给定观测点 i, 记
  - $a(i)$ 为点 i 与其所属类 C_i 中其他点之间的平均相异度值
  - $\bar{d}(i, C)$ 表示点 i 到其他类 $C(C \neq C_i)$ 内的所有点之间的平均相异度值
  - $b(i)$ 表示所有 $\bar{d}(i, C)$ 中的最小值
- 第 i 个观测点的 Silhouette 值定义为

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

- 平均的 Silhouetter 值为

$$\bar{s} = \frac{1}{n} \sum_{i=1}^{n} sw_i$$

- 显然, $-1 \leq s(i) \leq 1$, $s(i)$ 靠近 1 需要 $a(i) \ll b(i)$. 由 $a(i)$ 的定义知小的 $a(i)$ 意味着点 i 匹配该类非常好, 而大的 $b(i)$ 意味着点 i 匹配其他类很差, 从而 $s(i)$ 靠近 1 表明点 i 的聚类合适.
- $s(i)$ 靠近 -1 表明点 i 被聚类到相邻类中更合适
- $s(i)$ 靠近 0 表明点 i 在两个类的交集处, 即该点属于当前类或者相邻类均合适
- Kaufman and Rousseeuw 建议使用评价的轮廓宽来估计数据中的类数目:
  - $\bar{s} > 0.5$ 表明数据聚类合适
  - $\bar{s} < 0.2$ 表明数据不存在聚类特征
- **R**的包 **cluster**里的 *silhouette*函数将每个类中各点的 $s(i)$ 按从小到大以水平线画出.


Silhouette plot of pam(x = ruspini, k = 4)
n = 75
4 clusters C_j
j : n_j | $ave_{i \in Cj}$ s_i
1 : 20 | 0.73
2 : 23 | 0.75
3 : 17 | 0.67
4 : 15 | 0.80
0.0
0.2
0.4
0.6
0.8
1.0
Silhouette width s_i
Average silhouette width : 0.74

CH index

我们以 K-means 方法为例, 大部分想法可以推广到其他方法里.

- 给定类数目 k, K-means 算法最小化类内波动 (within-cluster variation):

$$W(k) = \sum_{i=1}^{k} \sum_{C(j)=i} \|\mathbf{x}_j - \bar{\mathbf{x}}_i\|^2$$

- 显然 $W(k)$ 随 k 增加而减少, $W(k)$ 越小表明类越紧凑.
- 类间波动性 (Between-cluster variation) 度量类与类之间远离的程度:

$$B(k) = \sum_{i=1}^{k} n_i \|\bar{\mathbf{x}}_i - \bar{\mathbf{x}}\|^2$$

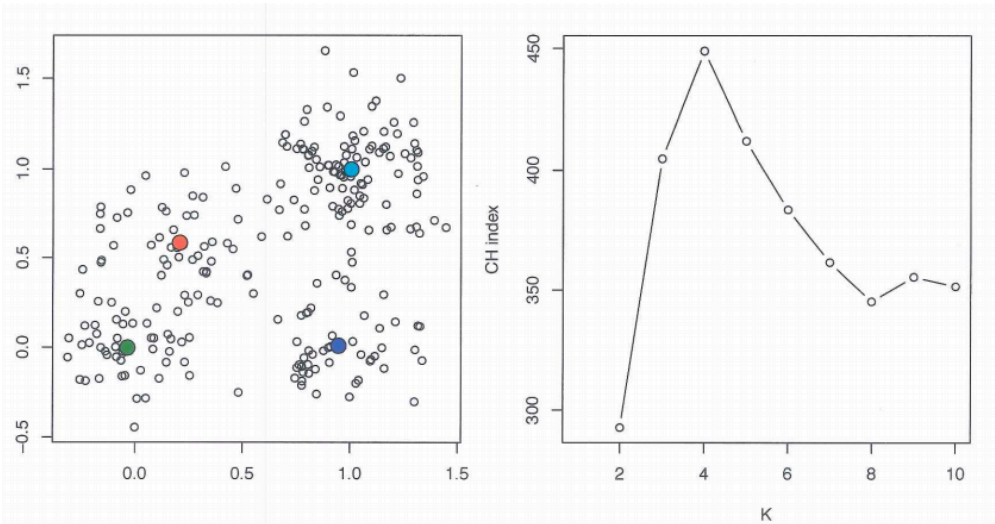
- 显然 $B(k)$ 随 k 增加而增加, $B(k)$ 越大表明类与类之间界限越清晰

- 单独使用 $W(k)$ 或 $B(k)$ 都是有缺点的, 因此自然地使用

$$CH(k) = \frac{B(k)/(k-1)}{W(k)/(n-k)}$$

选择合适的 k:

$$\hat{k} = \arg \max_{2 \le k \le K_{max}} CH(k)$$



Gap Statistic

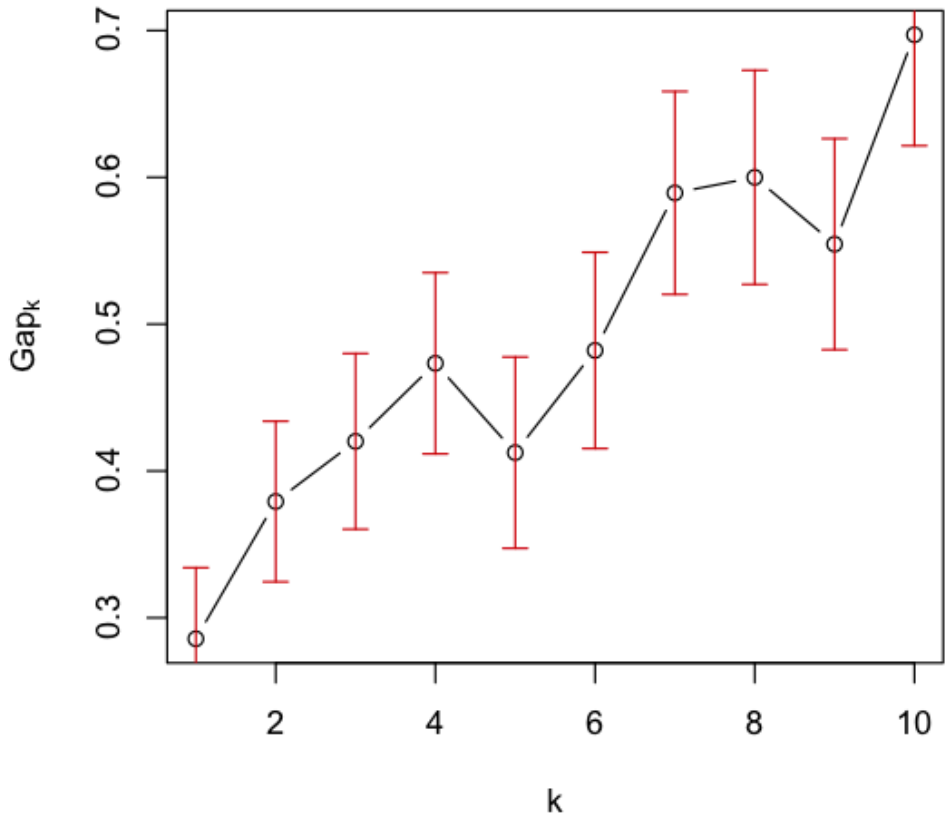
- $W(k)$ 随 k 增加而减少, 但是对每个 k 的减少量应该是有信息的!
- Tibshirani et al. (2001) 提出的Gap Statistic基于想法: 比较观察到的类内波动性 $W(k)$ 和 $W^*(k)$, 这里 $W^*(k)$ 是在零假设均匀分布 (数据为一个类, 且服从一个区域上的均匀分布) 下得到类内波动性, 通过随机模拟计算.
- 使用 Bootstrap 方法, 计算

$$\begin{aligned} Gap_n(k) &= E_n^* log(W^*(k)) - log(W(k)) \\ &\approx \frac{1}{B}\sum_b log(W_b^*(k)) - log(W(k)) \end{aligned}$$

以及 $sd_k = \frac{1}{B}\sum_b (W_b^*(k) - \bar{W}^*(k))^2$, 其中 $\bar{W}^*(k) = \frac{1}{B}\sum_b W_b^*(k)$.

- 令 $s_k = sd_k\sqrt{1+1/B}$, 最后选择类的数目为

$$\hat{k} = \inf\{k : Gap(k) \geq Gap(k+1) - s_{k+1}\}$$



1.5 聚类质量的评价

- 聚类性能评价方法通常分为外部评价法 (external criterion) 和内部评价法 (internal criterion).
- 外部评价法分析聚类结果与另一参考结果 (reference, 比如另一种聚类方法的结果或者真实的类别) 有多么相近. 例如
  - Purity, F-measure, Rand Statistics, Entropy 等
- 内部评价法来分析聚类的本质特点. 内部评价法常又分为绝对和相对评价法. 常用的方法有
  - 绝对评价法: Davis-Bouldin, Dunn, Expected Density ρ 等
  - 相对评价法: Elbow criterion, GAP statistics 等

外部评价法

- 设有 n 个样本点, 参考 (Reference) 类别为 $C^* = \{C_1^*, \ldots, C_l^*\}$.
- 某种聚类结果为 $C = \{C_1, \ldots, C_k\}$

F-measure:

- 精度 (Precision): $|C_j \cap C_i^*|/|C_j|$.
- 查全率 (Recall): $|C_j \cap C_i^*|/|C_i^*|$
- 计算 C_j 相对于 C_i^* 的 F-measure:$F_{ij}(\alpha) = \frac{1+\alpha}{\frac{1}{precision} + \frac{\alpha}{recall}}$, (加权调和平均), α 常取 1.

从而总的 F-measure 定义为

$$\sum_{i=1}^{l} \frac{|C_i^*|}{n} \max_{j=1,\ldots,k} \{F_{ij}\}$$

F-measure 较大时说明聚类结果满意.

- Entropy: 聚类 C 相对于 C^* 的 Entropy 定义为

$$H(C) = \sum_{C_j \in C} \frac{|C_j|}{n} \Big[- \sum_{C_j \cap C_i^* \neq \varnothing} \frac{|C_j \cap C_i^*|}{|C_j|} log_2 \frac{|C_j \cap C_i^*|}{|C_j|} \Big]$$

- Rand Index 考虑样本点中的点对, 记

$$n_{11} = \sharp\{(\mathbf{x}_i, \mathbf{x}_j) | \mathbf{x}_i, \mathbf{x}_j \in C_i; \mathbf{x}_i, \mathbf{x}_j \in C_k^*\}$$

$$n_{00} = \sharp\{(\mathbf{x}_i, \mathbf{x}_j) | \mathbf{x}_i \in C_{i_1}, \mathbf{x}_j \in C_{i_2}; \mathbf{x}_i \in C_{k_1}^*, \mathbf{x}_j \in C_{k_2}^*\}$$

$$n_{10} = \sharp\{(\mathbf{x}_i, \mathbf{x}_j) | \mathbf{x}_i, \mathbf{x}_j \in C_i; \mathbf{x}_i \in C_{k_1}^*, \mathbf{x}_j \in C_{k_2}^*\}$$

$$n_{01} = \sharp\{(\mathbf{x}_i, \mathbf{x}_j) | \mathbf{x}_i \in C_{i_1}, \mathbf{x}_j \in C_{i_2}; \mathbf{x}_i, \mathbf{x}_j \in C_{k_2}^*\}$$

Rand index 定义为

$$R = \frac{n_{11} + n_{00}}{\binom{n}{2}}$$

$R = 0$ 表明两种聚类没有重叠, $R = 1$ 表示两个聚类完全相同. 其严重依赖于聚类数目.

- Adjusted Rand Index 两种不相关随机划分的 Rand index 期望值并不为常数 (零), Hubert and Arabie (1985) 对 Rand index 进行了调整 (假设超几何分布: 固定列联表的边际), 使得对两种独立的聚类值为 0, 两种完全相同的聚类值为 1. Meila (2003) 指出 Ajusted Rand index 值可取负值.