聚类分析

- 聚类分析（cluster analysis）是无监督学习（unsupervised learning）中的一类重要方法，与监督学习（supervised learning）不同，由于缺乏样本标签，聚类分析只使用样本的特征信息进行分析。

- 聚类分析是一种探索性的分析方法，它基于某个特定标准将一个数据集划分为若干个不相交的子集，每个子集称为一个“簇”（cluster）或者“类”。

- 聚类分析的本质思想是将相似的数据对象归为一类，差异较大的数据对象划分到不同的类，从而将数据集划分为一系列具有不同模式的类别，即“物以类聚，人以群分”。

What is similarity?

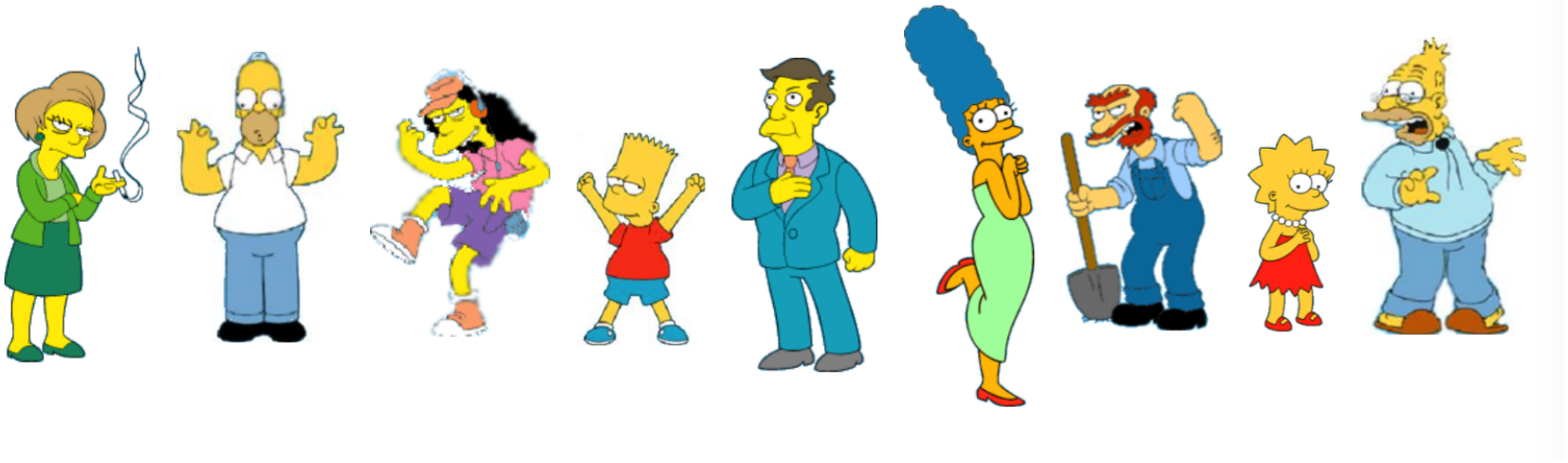
The quality or state of being similar; likeness; resemblance; as, a similarity of features.

Webster's Dictionary

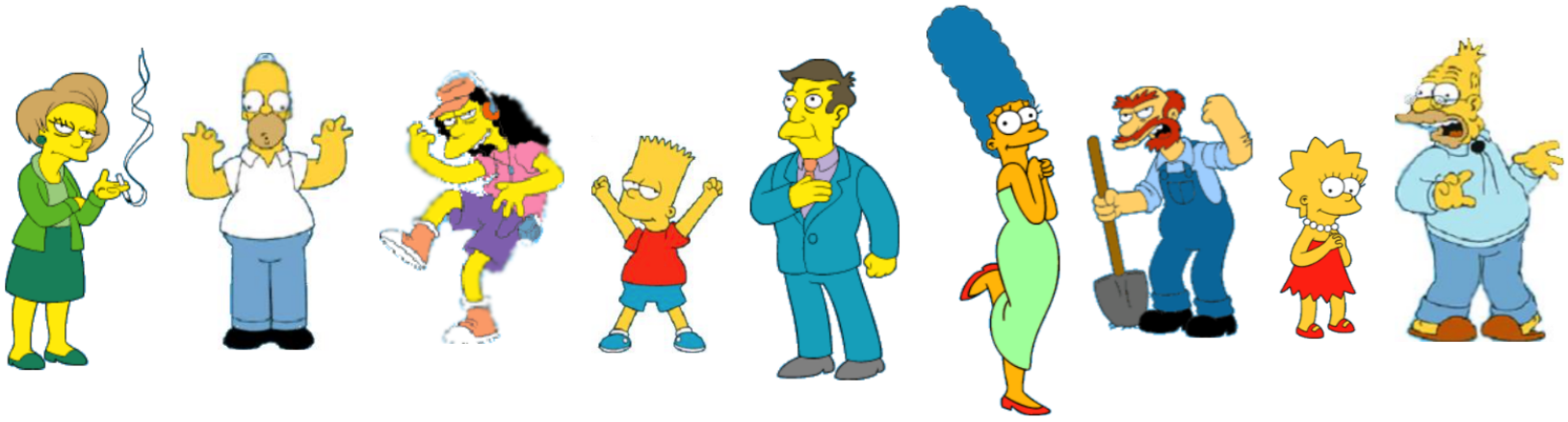
Similarity is hard to define, but…
"*We know it when we see it*"

The real meaning of similarity is a philosophical question. We will take a more pragmatic approach.

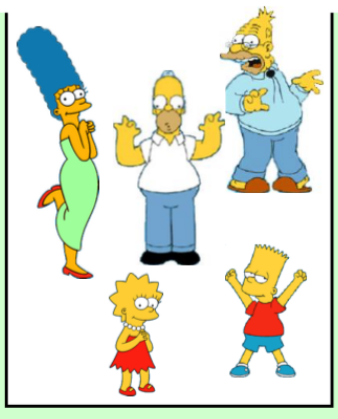
What is a natural grouping among these objects?



What is a natural grouping among these objects?



Clustering is subjective



Simpson's Family

School Employees

Females

Males

聚类分析

例 12-1 现收集到 20 种不同品牌啤酒的数据，它包括 20 个品牌的观测值和 4 个特征变量（热量、含钠量、酒精量和成本），如表 12-1 所示。现在需要对这 20 个品牌进行聚类，从而探究不同类别的品牌具有的特征。

表 12-1 20 种啤酒品牌的数据

品牌	**热量**	**含钠量**	**酒精量**	**成本**
Budweiser	144	15	4.7	0.43
Schlitz	151	19	4.9	0.43
Lowenbrau	157	15	0.9	0.48
Schlitz	170	7	5.2	0.73
Schlitz	152	11	5.0	0.77
...	...	...	...	...

聚类分析

- 假设有 n 个观测样本，每个样本可以观测到 p 个变量，则样本集合以矩阵 $\mathbf{X}$ 表示为

$$\mathbf{X} = [x_{ij}]_{n\times p} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

- 其中，矩阵的第 i 行表示第 i 个样本，$i = 1, 2, \ldots, n$；矩阵的第 j 列表示第 j 个变量，$j = 1, 2, \ldots, p$；矩阵元素 x_{ij} 表示第 i 个样本在第 j 个变量上的观测值。

聚类分析

- 本章将按照图 12-1 的示意图对不同类别聚类方法进行介绍

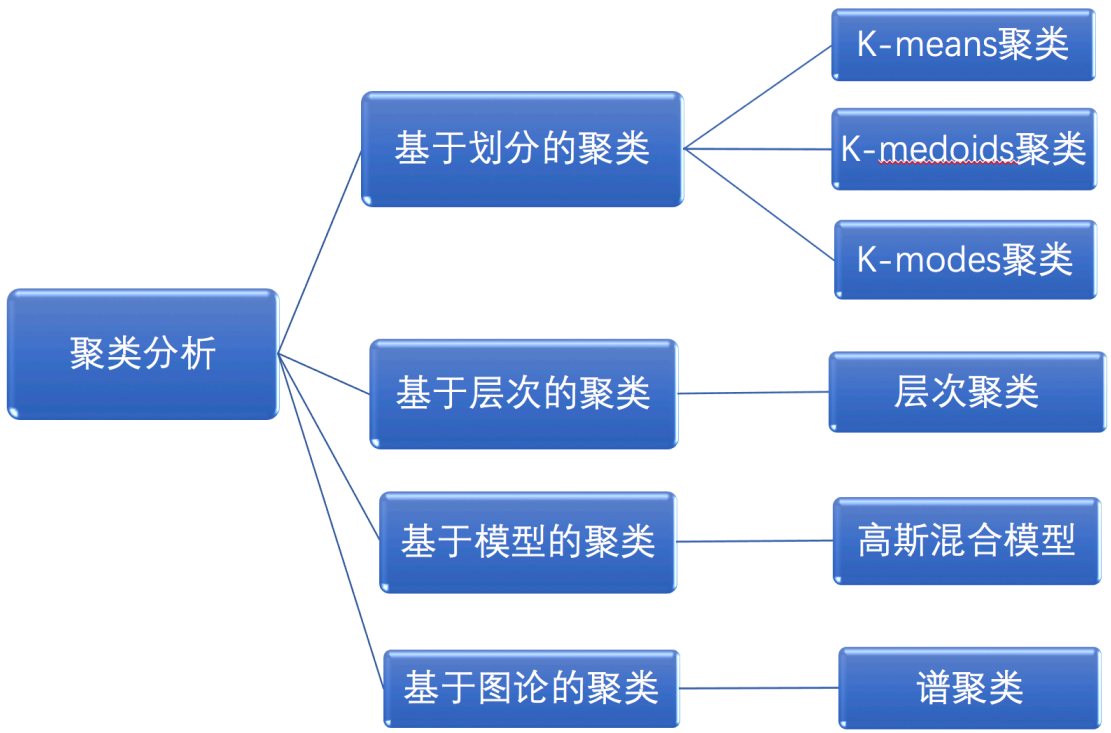


图 12-1　聚类分析中不同聚类方法的结构示意图

基于划分的聚类

基于划分（*Partioning-based*）的聚类方法

1. 将样本随机分割形成多个类别，作为初始的类别结构；这些类别需要满足两个基本条件：
   - 每个类必须包含至少一个样本
   - 每个样本必须属于其中一个类
2. 基于每个样本到各类中心的距离，重新确定新的分组。
3. 重复步骤 1 和 2 直至收敛。

差异性（*dissimilarity*）

在此类方法中，需要事先定义一个衡量样本之间差异性（距离）的指标，如果两个样本之间差异性较小，则倾向于归为一类，反之则倾向于归为不同的类。下面将根据变量类型的不同给出相应的“差异性”（“距离”）定义。

基于划分的聚类

连续型变量（*continuous variables*）

令 $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^\top$ 和 $\mathbf{x}_j = (x_{j1}, \ldots, x_{jp})^\top$ 分别表示样本 i 和 j，且样本中的每个元素都是连续型变量的观测值。令 d_{ij} 表示两者间的差异性（距离），那么可采用以下几种“距离”对观测样本之间的相似性进行度量：

- 明考夫斯基距离（Minkowski distance）

$$d_{ij} = \left(\sum_{k=1}^{p} |x_{ik} - x_{jk}|^q \right)^{1/q}$$

明考夫斯基距离简称明氏距离，根据 q 的不同取值可以分成：绝对距离（$q = 1$，Absolute distance）、欧式距离（$q = 2$，Euclidean distance）和切比雪夫距离（$q = \infty$，Chebyshev distance）。欧氏距离是常用的距离，但是有一些缺陷。一是它没有考虑到总体的变异对“距离”远近的影响，显然一个变异程度大的总体可能与更多样本近些，即使它们的欧氏距离不一定最近；另外，欧氏距离受变量的量纲影响较大。

基于划分的聚类

- 马氏距离（Mahalanobis distance）

$$d_{ij} = \left[(\mathbf{x}_i - \mathbf{x}_j)^\top \Sigma^{-1} (\mathbf{x}_i - \mathbf{x}_j) \right]^{1/2}$$

其中，Σ 为样本 $\mathbf{X}$ 对应的协方差矩阵。马氏距离又称为广义欧氏距离，它与上述几种距离的主要不同在于它考虑了观测变量之间的相关性。如果各变量之间相互独立，即观测变量的协方差矩阵是对角矩阵，则马氏距离就退化为用各个观测指标的标准差的倒数作为权数的加权欧氏距离。马氏距离考虑了观测变量之间的变异性，且不再受各指标量纲的影响。将原始数据作线性变换后，马氏距离不变。

基于划分的聚类

- 余弦距离（Cosine distance）

$$d_{ij} = \left(\sum_{k=1}^{p} x_{ik}x_{jk}\right) \Big/ \left(\sum_{k=1}^{p} x_{ik}^2 \sum_{k=1}^{p} x_{jk}^2\right)^{1/2}$$

余弦距离是用向量空间中两个向量夹角的余弦值来衡量两个个体间的差异。向量是多维空间中有方向的线段，如果两个向量的方向趋于一致，则夹角接近 0。该距离实际上是向量 $\mathbf{x}_i$ 和 $\mathbf{x}_j$ 的夹角的余弦值，即关注的是个体方向上的差异，对绝对数值不敏感，所以可以解决例如不同个体间存在的度量标准不统一的问题。

基于划分的聚类

有序型变量（*ordinal variables*）

有序型变量变量往往具有次序前后或等级高低之分，例如课程的成绩（A,B,C,D,F），服务的满意度（非常不满意，较为不满意，一般，较为满意，非常满意）等。假如一个有序变量有 M 个等级，则通常将其赋值为 $1,\ldots,M$ 的连续整数，并进行如下转换

$$x_{ik}^{*}=\frac{(x_{ik}-1/2)}{M}$$

基于划分的聚类

类别型变量（*categorical variables*）

类别型变量的每个取值表示一个类别，并且类别之间无次序先后之分，例如地图的颜色可能有五种：红色、黄色、绿色、粉红色和蓝色，每种颜色仅仅代表一个类别。此类变量包含的不同类别可以用字母、符号或者一组整数表示，这些整数只是用于数据处理，并不代表任何特定的顺序。对于样本 $\mathbf{x}_i$ 和 $\mathbf{x}_j$，它们之间的相似性可以用简单匹配方法来计算，即

$$d_{ij} = p - m$$

其中，m 是匹配的数目，即 $\mathbf{x}_i$ 和 $\mathbf{x}_j$ 取值相同类别的数目。

基于划分的聚类

K-means 聚类

- K-means 聚类是迭代下降聚类算法中最著名的算法之一，它适用于所有变量均为数值型变量的数据集，并且采用欧氏距离的平方作为样本间的距离
- K-means 聚类的基本想法是，一个好的聚类结果应该表现为组内的样本间差异尽量小，而不同组之间的样本差异尽量大，这两类差异分别用组内距离和组间距离来刻画，由于组内距离与组间距离之和为总距离，因此最小化组内距离等价于最大化组间距离，下面将给出组内距离的定义。

基于划分的聚类

组内距离（*within-cluster sum of squares; WCSS*）

- 给定一组含有 p 个变量的样本量为 n 的样本矩阵 $\mathbf{X}$，事先设定类别个数 K，使得每个样本仅属于其中一类，第 k 个类别含有的样本子集定义为 $C_k \subseteq \{1,\ldots,n\}$，且 $\cup_{k=1}^{K} C_k = \{1,\ldots,n\}$，令 $C = \{C_1,\ldots,C_K\}$ 则组内距离定义为：

$$\mathsf{WCSS}(C) = \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,j \in C_k} \sum_{l=1}^{p} d_{ij,l}^2 \qquad (12.1)$$

- 其中，n_k 是第 k 类中的样本个数，$d_{ij,l}$ 是样本 $\mathbf{x}_i$ 和 $\mathbf{x}_j$ 在变量 l 上的欧式距离。直接最小化目标 (12.1) 是极为困难的，因为一共有 K^n 种聚类方式，直接利用穷举法费时费力，因此接下来介绍 K-means 聚类一种经典的算法

基于划分的聚类

- 式子 (12.1) 可以重写为

$$\mathsf{WCSS}(C) = \sum_{k=1}^{K} \sum_{i \in C_k} ||\mathbf{x}_i - \overline{\mathbf{x}}_k||_2^2 \tag{12.2}$$

- 其中，$\overline{\mathbf{x}}_k = (\overline{x}_{k,1}, \ldots, \overline{x}_{k,p})^\top$，且 $\overline{x}_{k,l}$ 表示第 k 类中的样本在第 l 个变量上的均值，最小化式子 (12.2) 意味着将 n 个样本分配到 K 个类中使得每个类的点到该类中心的平均距离最小。基于式子 (12.2)，一个经典的迭代下降算法可以通过求解下面的增广优化问题：

$$\min_{C, \{\boldsymbol{\mu}_k\}_{k=1}^K} \sum_{k=1}^{K} \sum_{i \in C_k} ||\mathbf{x}_i - \boldsymbol{\mu}_k||_2^2 \tag{12.3}$$

基于划分的聚类

- 最小化 (12.3) 可以通过交替优化 C 和 $\{\boldsymbol{\mu}_k\}_{k=1}^K$ 得到。在给定聚类结构 C 时，最优的中心 $\boldsymbol{\mu}_k$ 恰好为第 k 类所有样本的均值。
- 在给定聚类结构 C 时，最优的中心 $\boldsymbol{\mu}_k$ 恰好为第 k 类所有样本的均值。
- 在给定每个类别的中心时，，最优的聚类结构为将每个点归到离其最近的中心所代表的的类上。

算法 12.1 K-means 聚类算法

1. 给定类数 K，为每个观测随机分配一个从 1 到 K 的数字，这些数字即表示这些观测的初始类。
2. 重复以下步骤，直至类的分配完成为止：

   1）分别计算 K 个类的类中心。第 k 个类的类中心是该类中所有 p 维观测向量的均值向量；

   2）计算每个观测与各个类中心的距离，将其重新分配到与其距离最小的类中。

图 12-2　K-means 聚类算法

基于划分的聚类

例 12-2 给定如下 5 个样本构成的矩阵

$$\mathbf{X} = \begin{bmatrix} 0 & 2 & 3 & 4 & 5 \\ 1 & 1 & 2 & 4 & 3 \\ 2 & 0 & 1 & 3 & 5 \end{bmatrix}^{\top}$$

试着 K-means 聚类算法将该样本划分为 2 类。

解：按照算法 12.1，聚类过程如下所示：

(1) 初始分配为 2 类，其中 $C_1^{(0)} = \{1, 2\}$，$C_2^{(0)} = \{3, 4, 5\}$；

(2) 计算 $C_1^{(0)}$ 的样本中心为 $\boldsymbol{\mu}_1^{(0)} = (1, 1, 1)^{\top}$，$C_2^{(0)}$ 的样本中心为 $\boldsymbol{\mu}_2^{(0)} = (4, 3, 3)^{\top}$；

基于划分的聚类

解:

(3) 计算 5 个样本点分别到两中心的距离:

- 对 $\mathbf{x}_1 = (0,1,2)^\top$, $d(\mathbf{x}_1, \boldsymbol{\mu}_1^{(0)}) = 2$, $d(\mathbf{x}_1, \boldsymbol{\mu}_2^{(0)}) = 21$, 将 $\mathbf{x}_1$ 分到类 $C_1^{(0)}$;
- 对 $\mathbf{x}_2 = (2,1,0)^\top$, $d(\mathbf{x}_2, \boldsymbol{\mu}_1^{(0)}) = 2$, $d(\mathbf{x}_2, \boldsymbol{\mu}_2^{(0)}) = 17$, 将 $\mathbf{x}_2$ 分到类 $C_1^{(0)}$;
- 对 $\mathbf{x}_3 = (3,2,1)^\top$, $d(\mathbf{x}_3, \boldsymbol{\mu}_1^{(0)}) = 5$, $d(\mathbf{x}_3, \boldsymbol{\mu}_2^{(0)}) = 6$, 将 $\mathbf{x}_3$ 分到类 $C_1^{(0)}$;
- 对 $\mathbf{x}_1 = (4,4,3)^\top$, $d(\mathbf{x}_4, \boldsymbol{\mu}_1^{(0)}) = 22$, $d(\mathbf{x}_4, \boldsymbol{\mu}_2^{(0)}) = 1$, 将 $\mathbf{x}_4$ 分到类 $C_2^{(0)}$;
- 对 $\mathbf{x}_1 = (5,3,5)^\top$, $d(\mathbf{x}_5, \boldsymbol{\mu}_1^{(0)}) = 36$, $d(\mathbf{x}_5, \boldsymbol{\mu}_2^{(0)}) = 5$, 将 $\mathbf{x}_5$ 分到类 $C_2^{(0)}$;

基于划分的聚类

解：

(4) 得到新的类 $C_1^{(1)} = \{1, 2, 3\}$，$C_2^{(1)} = \{4, 5\}$，计算新的类别中心：

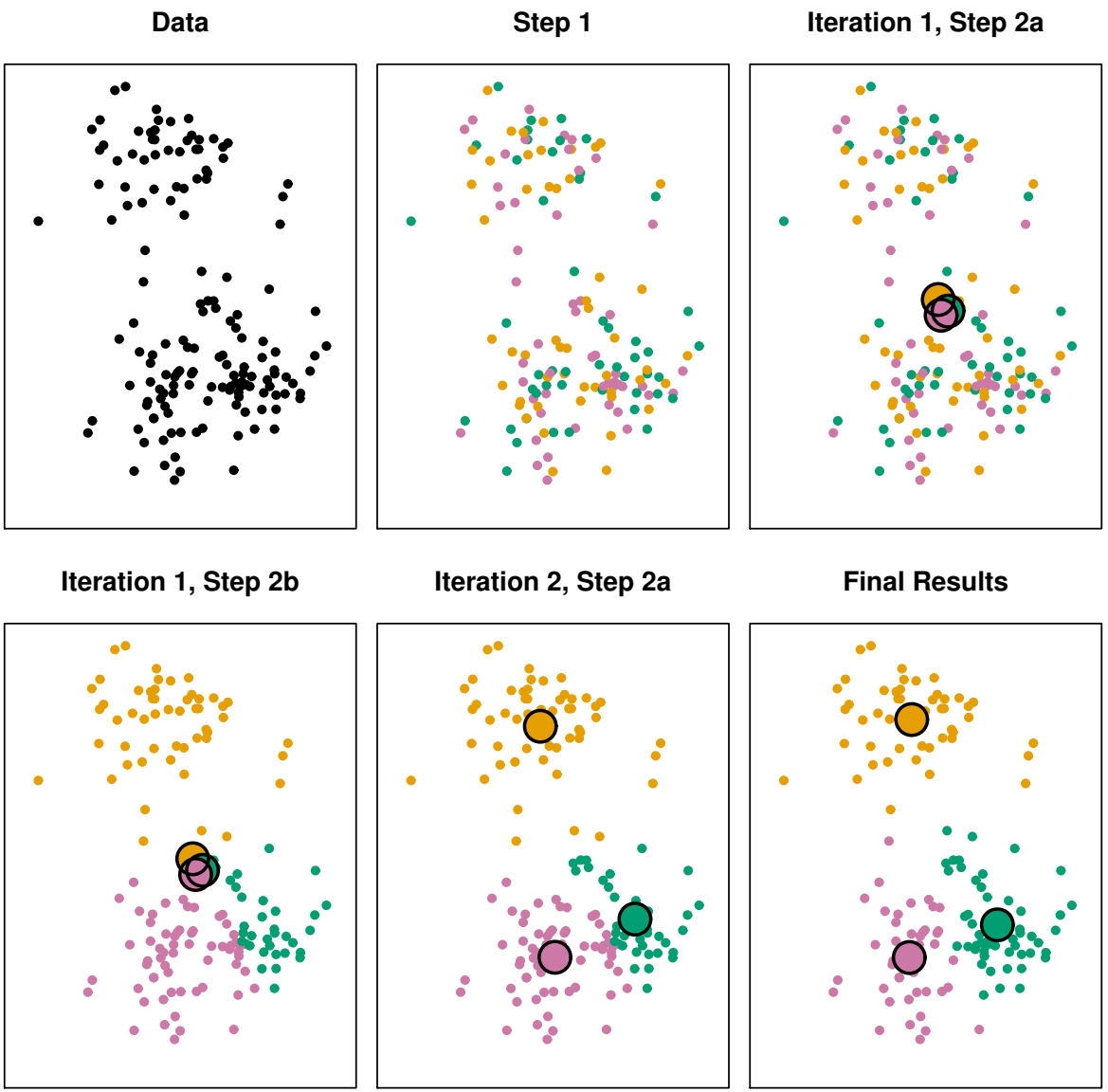
$$\boldsymbol{\mu}_1^{(1)} = (5/3, 4/3, 1)^\top, \quad \boldsymbol{\mu}_2^{(1)} = (4.5, 3.5, 4)^\top$$

(5) 重复步骤 (3) 和 (4)。将 $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ 划分到类 $C_1^{(1)}$，将 $\mathbf{x}_4, \mathbf{x}_5$ 划分到类 $C_2^{(1)}$，从而得到新的类 $C_1^{(2)} = \{1, 2, 3\}$，$C_2^{(2)} = \{4, 5\}$。

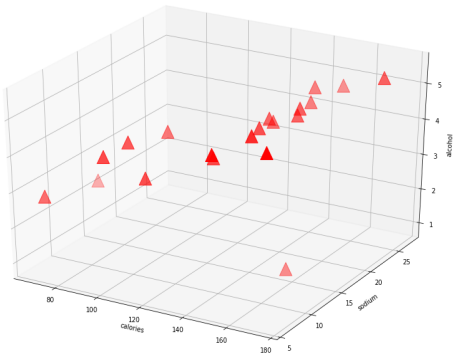
(6) 由于新的分类和上一步的类别一致，因此停止更新，最终聚类结果为：

$$C_1^* = \{1, 2, 3\}, C_2^* = \{4, 5\}$$

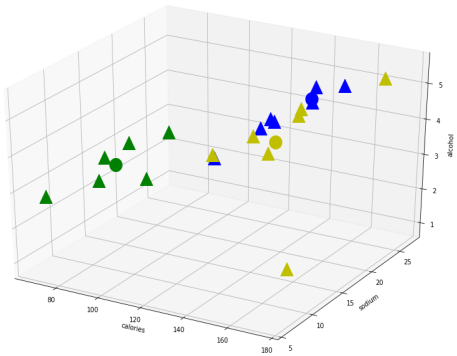
Example


Data
Step 1
Iteration 1, Step 2a
Iteration 1, Step 2b
Iteration 2, Step 2a
Final Results

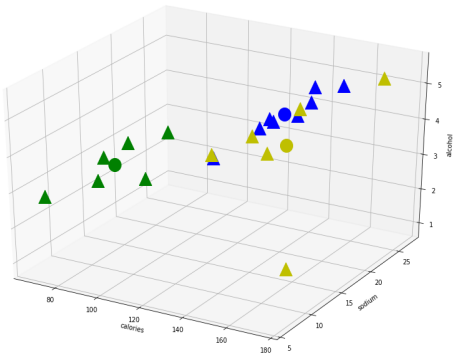
基于划分的聚类



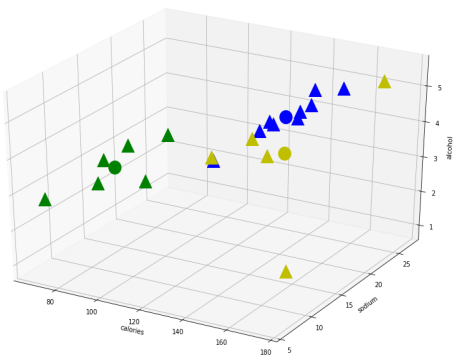
(a) 原始数据



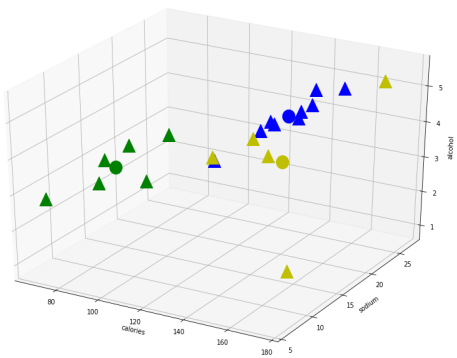
(b) 初始划分



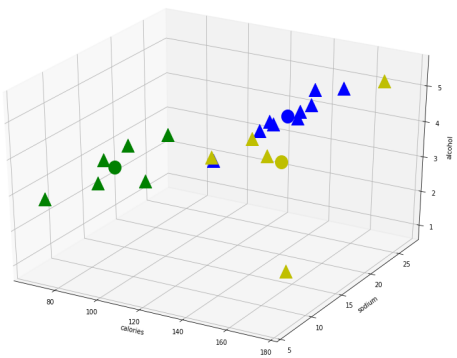
(c) 迭代 1



(d) 迭代 2



(e) 迭代 3



(f) 迭代 4

图 12-3　例 12-1 的聚类过程示意图

基于划分的聚类

K-means 聚类的优缺点

K-means 聚类由于其原理简单直观，运行速度较快，因此被广泛应用于实际分析，尤其是大规模数据集的挖掘中，但是缺点也很明显。

- 首先，必须事先给定一个类数 K，不合适的 K 值往往造成较差的聚类结果；
- 其次，K-means 算法找到的解不是全局最优解，而是局部最优解，因此聚类结果对初识聚类中心的设定较为敏感；
- 第三，K-means 算法采用的是欧式距离作为样本之间的“距离”度量，因此容易受到离群点的影响。

Example: different starting values

基于划分的聚类

K-medoids 聚类 *VS K-means* 聚类

- 一方面，K-medoids 聚类可以使用任意定义的“距离”（不再局限于欧式距离），因此对于分类型变量，可以采用相对应的距离指标进行聚类；
- K-means 聚类的中心点选取是基于类别中所有样本的均值，而 K-medoids 聚类将每一类中心点的选取限制在该类中已经存在的任意一点，因此极大地削弱了离群点的影响。

基于划分的聚类

K-medoids 聚类

- 给定当前的类别划分 C,，寻找的中心点需要最小化如下的目标函数：

$$i_k^* = \min_{i \in C_k} \sum_{j \in C_k} d_{ij} \tag{12.4}$$

- 其中，d_{ij} 表示任意距离度量下的样本 i 和样本 j 的距离，由 (12.4) 可以看出，通过遍历第 k 类中所有的点才能找到第 k 类的中心点。相比于直接计算第 k 类的样本均值，这种确定中心的方法计算较为复杂，时间复杂度从 $O(n_k)$ 提升到 $O(n_k^2)$。
- 在给定当前的中心之后，新的聚类分割则是通过最小化如下的目标函数：

$$C(i) = \min_{1 \le k \le K} d_{ii_k^*} \tag{12.5}$$

- 其中，$C(i)$ 表示样本 i 所属的类别。

基于划分的聚类

算法 12.2　K-medoids 聚类算法

1. 给定类数 K，从 n 个样本中随机挑选 K 个样本作为中心点，计算剩余的 $n-K$ 个点离各个中心点的距离，将其分配给最近的中心点对应的类。
2. 重复以下步骤，直至类的分配完成为止：

   1） 分别计算 K 个类的类中心。对于第 k 个类中除中心以外的所有点，依次将这些点作为中心点，分别计算这些点作为新的中心时（12.4）式的值，选择（12.4）式最小时对应的点作为最终确定的中心点；

   2）计算每个观测与各个类中心的距离，将其重新分配到与其距离最小的类中。

图 12-5　K-medoids 聚类算法

基于划分的聚类

K-modes 聚类

- 考虑到现实中存在的许多类别型变量，学者们将距离度量设置为简单匹配距离度量，类别中心定义为众数，从而提出 K-modes 聚类
- K-modes 聚类的目标函数与 (12.3) 类似，只是距离度量从欧式距离调整为简单匹配的距离度量。可以证明，通过给定的中心，将各个点归结到离其最近的中心所代表的的类上使得目标函数最小；通过给定的类别划分，每个类别的中心选为该类别各个变量属性的众数时目标函数最小。

基于划分的聚类

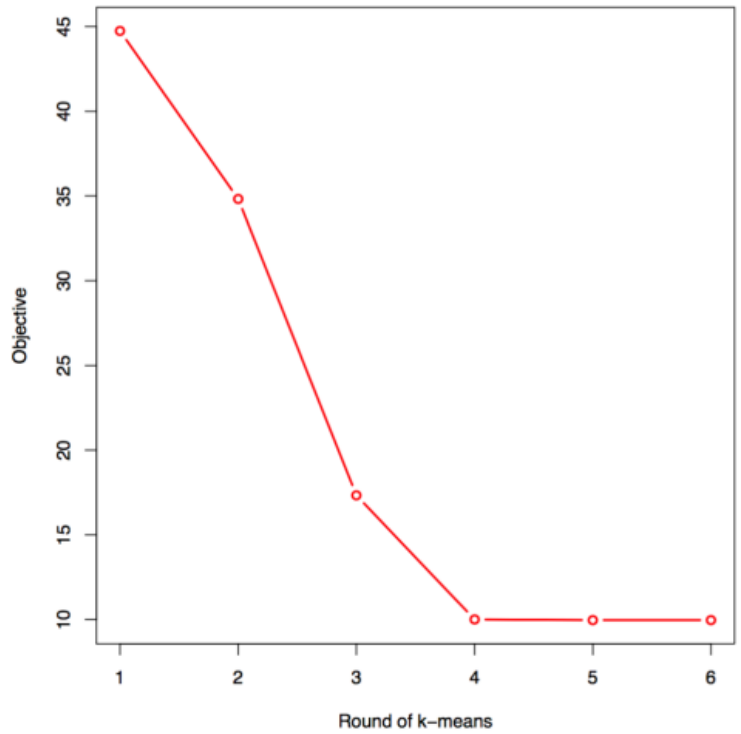
算法 12.3　K-modes 聚类算法

1. 给定类数 K，随机确定 K 个类别中心。
2. 重复以下步骤，直至类的分配完成为止：
   2）对于每个样本，依次计算它与 K 个类别中心的距离，将其分配到与其距离最小的类中；
   3）分别计算 K 个类的类中心。对于第 k 个类，则以该类中每个变量出现属性的众数构成变量向量作为新的类别中心。

图 12-6　K-modes 聚类算法

Choosing K

- Choosing K is a nagging problem in cluster analysis.
- Sometimes, the problem determines K. For example, clustering customers for K group in a business.
- Usually, we seek the natural clustering, but what does this mean?
- Plot the objective function VS K. Elbow finding.



基于层次的聚类

层次聚类

层次聚类，是将数据按照节点间的邻近程度进行分层排列，形成树状图的层次结构，其中的叶节点则表示各个数据样本

- 前文介绍了基于划分聚类的三种算法（K-means, K-medoids 和 K-modes），这些算法的聚类结果取决于事先设定的类别个数 K 和初识的类别划分。
- 与此相反，层次聚类无需这些初识设置，只需要用户基于两类中观测样本的成对距离来设定两类之间的距离度量方式，从而产生分层的聚类结果。
- 层次聚类又可细分为聚合型（agglomerative）层次聚类和分裂型（divisive）层次聚类两种方式。这里仅对聚合型层次聚类进行介绍。

基于层次的聚类

聚合型层次聚类

- 聚合型层次聚类采用的是一种自下而上的策略，开始时每个观测样本各自作为一类，随着层次的提升，在每个层次递归地将最近的两类合并为新的一类，这样每上升一个层次就会形成一个新的类，同时类别的总数减少一个，最终到顶层时所有样本聚为一类。
- 事实上，层次结构中的每个层都表示将数据分为不相交类别的一种分割方式，整个层次结构表示这种分组的有序序列，用户可以根据经验来决定哪个层次的聚类结果最有意义。

基于层次的聚类

聚合型层次聚类

- 层次聚类的层次结构可以通过二叉树来直观地展示，其中树的节点代表类，根节点表示整个数据集。
- 最底层所反映的终端的 n 个节点表示 n 个观测样本（自身为一类），每个非终端节点（“父”）具有两个子节点，这两个子节点代表两个类，这个形式显示两个类（“子”节点）聚成（合并）一个新的类（“父”节点）。
- 树形图（dendrogram）提供了对层次聚类具有高度可解释性的图形描述，这在后面的例子中将会进一步说明。

基于层次的聚类

- 在层次聚类法中，每上升一个层次，原先层次中距离较近的两个类会聚在一起，随着层次的提升，最终形成一个类。
- 由于这里需要度量类别之间的两两距离，所以需要将观测之间相似度的概念扩展到观测组的相似度上。在这里，同样采用“距离”来度量观测组的相似性，并且基于不同的“链接”方式定义不同的“距离”准则。

基于层次的聚类

- 表 12-2 列出了 4 种常用的链接形式，分别是：单链接法（singlelinkage）、完全链接法（complete linkage）、平均链接法（average linkage）以及中心链接法（centroid linkage）。

距离形式	描述
单链接法	计算 A 类和 B 类之间的所有观测样本间的距离，并记录其中最短的距离作为两类间的距离；
完全链接法	计算 A 类和 B 类之间的所有观测样本间的距离，并记录其中最长的距离作为两类间的距离；
平均链接法	计算 A 类和 B 类之间的所有观测样本间的距离，并记录这些距离的平均值作为两类间的距离；
中心链接法	计算 A 类和 B 类的类别中心，并计算两中心的距离作为两类间的距离。

图 12-7　4 种常用的链接方式

基于层次的聚类

聚合型层次聚类的工作流程

- 给定样本差异性的衡量指标和类别的链接方式，通过对类间距离最小的 2 类逐一进行合并，直到最终聚为 1 类。

算法 12.4 层次聚类算法（聚合型）

1. 将每个观测视为一类，共得到 n 个初始类。
2. 计算 n 个观测样本中，总共 $\binom{n}{2} = n(n-1)/2$ 对两两观测样本之间的距离。
3. 对 $K = n, n-1, \quad , 2$，重复以下步骤直至所有观测都属于一个类或者满足某个终止条件：

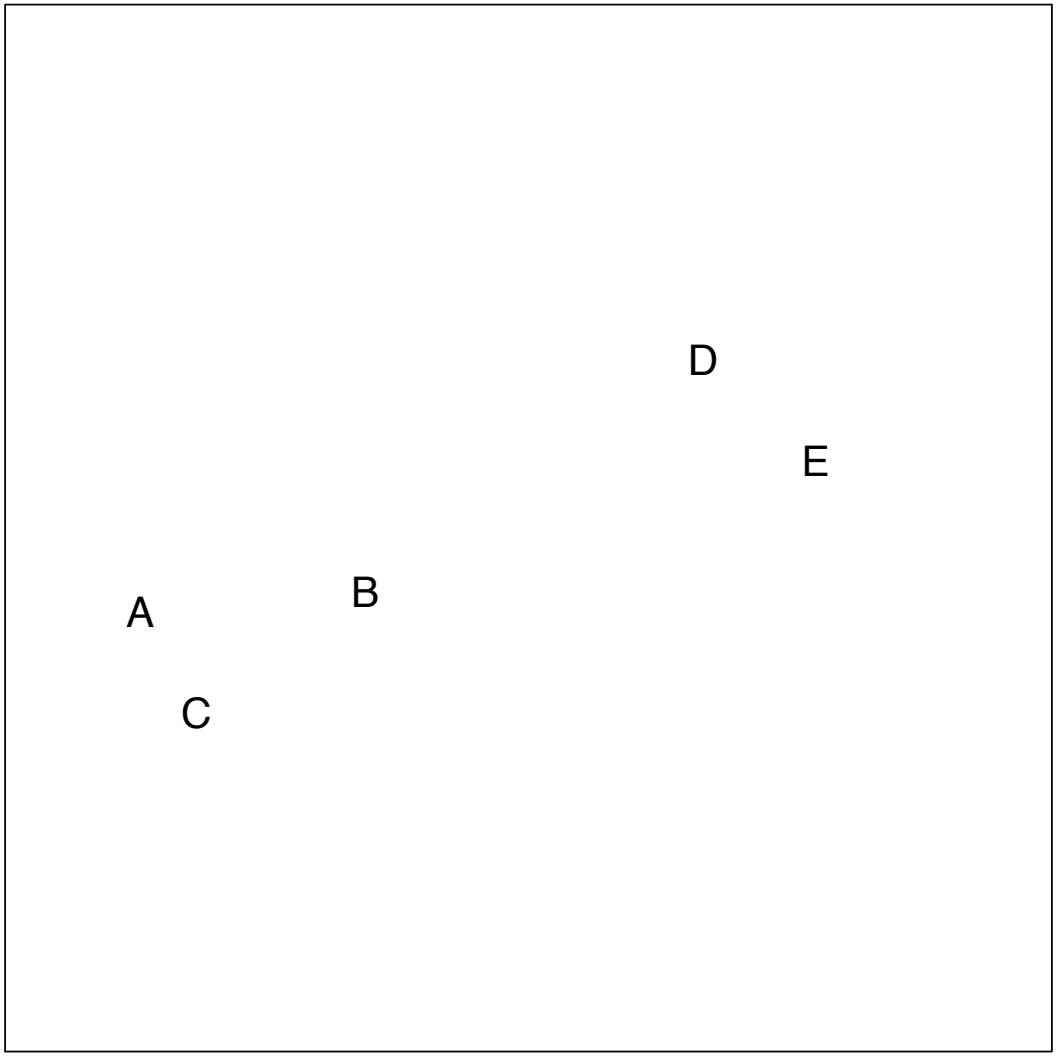
   1）在 K 个类中，比较任意两类间的距离（基于最短距离），将距离最小的（即最相似的）那两类结合起来，形成新的类；

   2）计算剩下的 $K-1$ 个新类中每两个类间的距离。

图 12-8　聚合型层次聚类算法

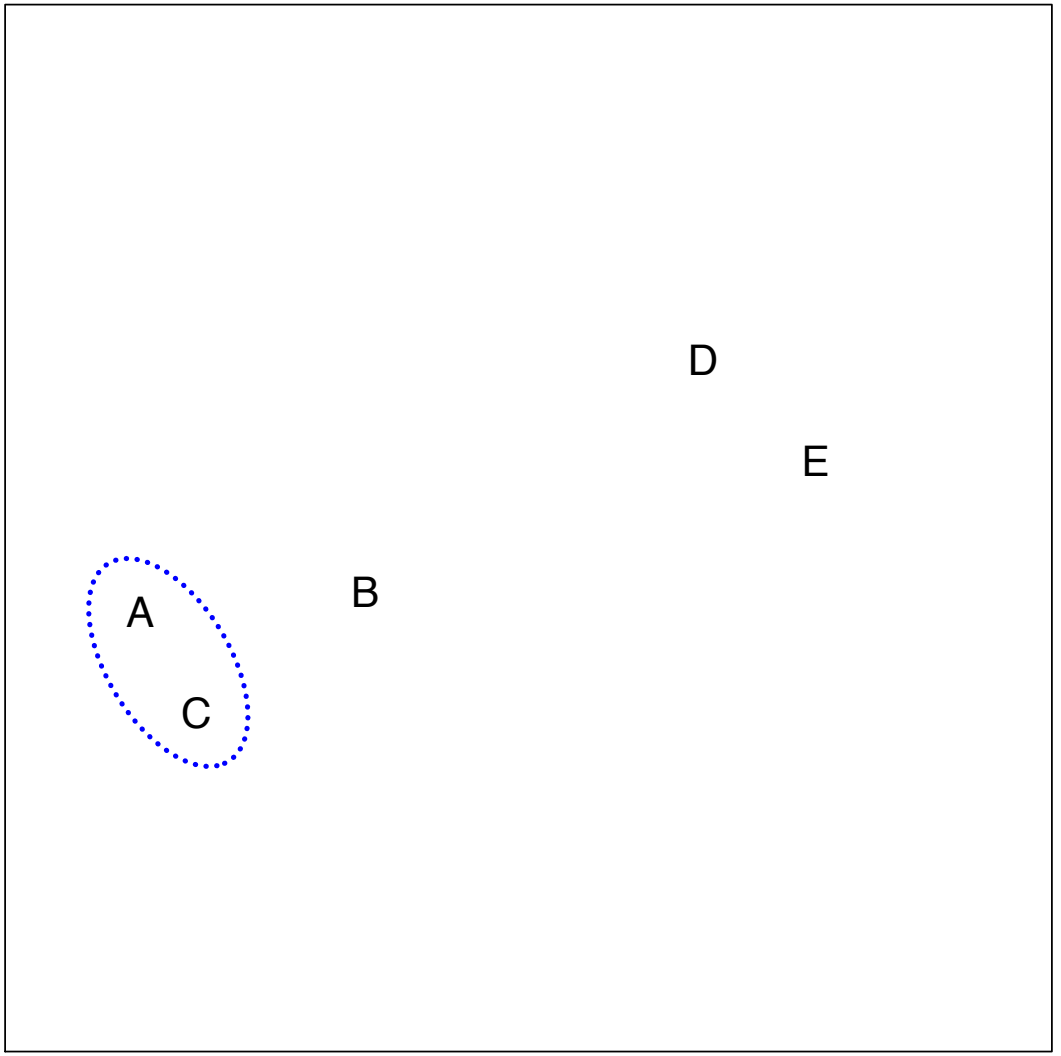
Hierarchical Clustering: the idea

Builds a hierarchy in a "bottom-up" fashion...



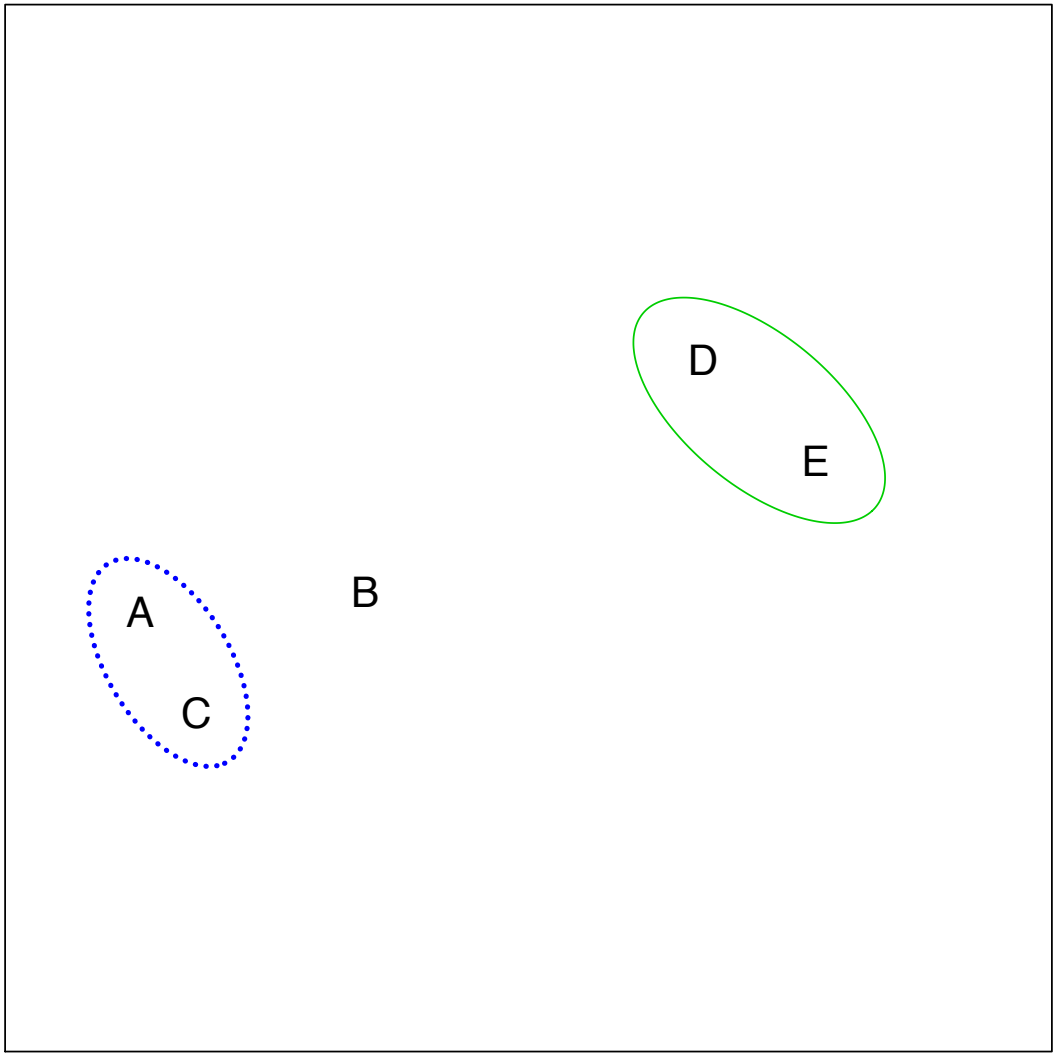
Hierarchical Clustering: the idea

Builds a hierarchy in a "bottom-up" fashion...



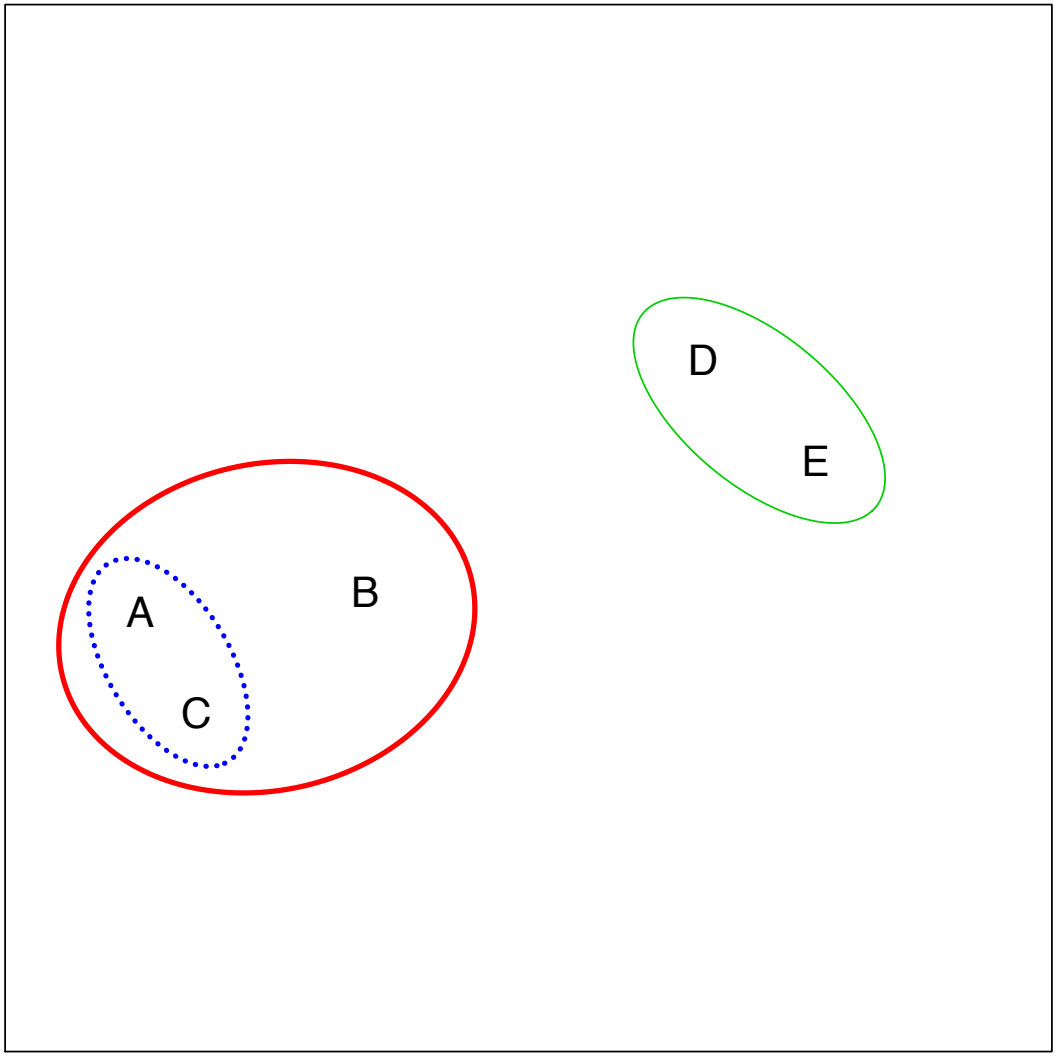
Hierarchical Clustering: the idea

Builds a hierarchy in a "bottom-up" fashion...



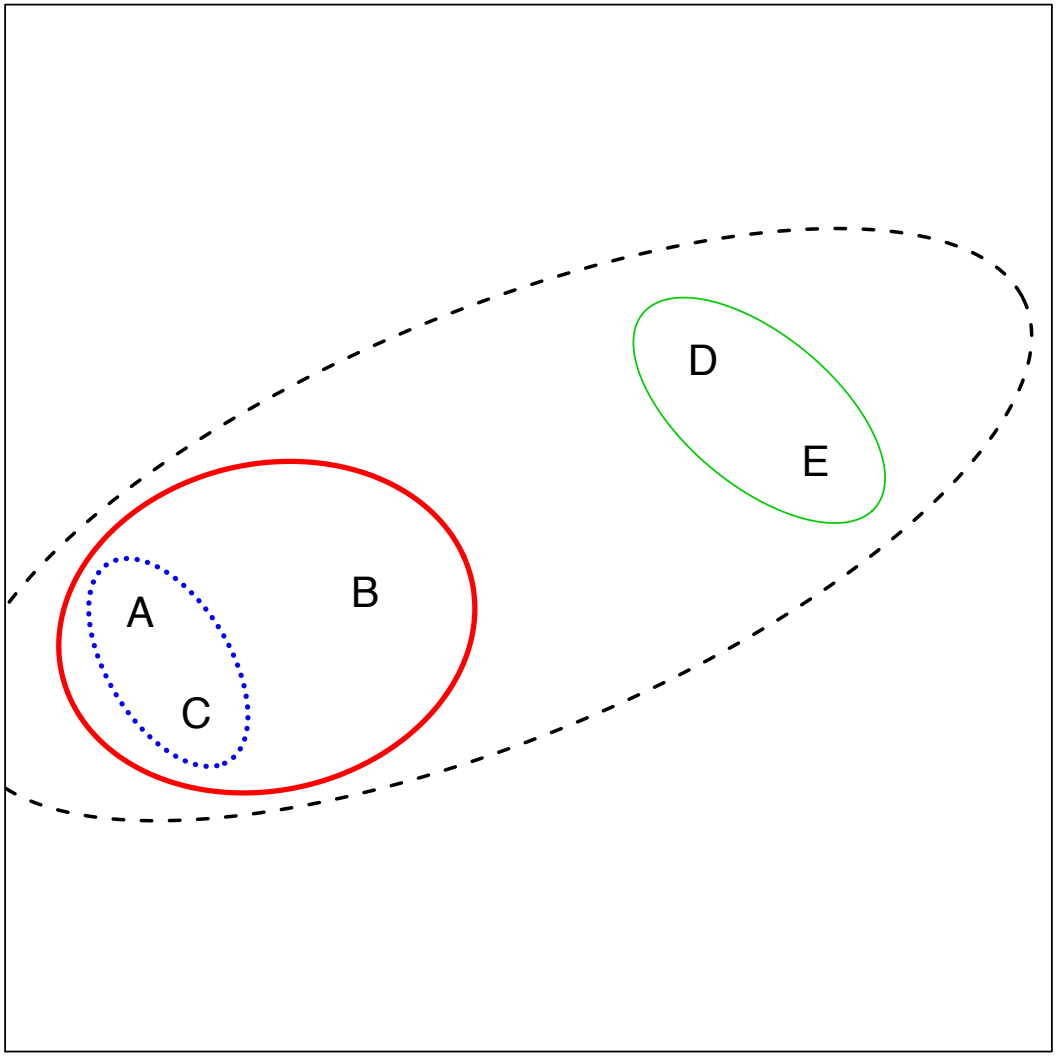
Hierarchical Clustering: the idea

Builds a hierarchy in a "bottom-up" fashion...



Hierarchical Clustering: the idea

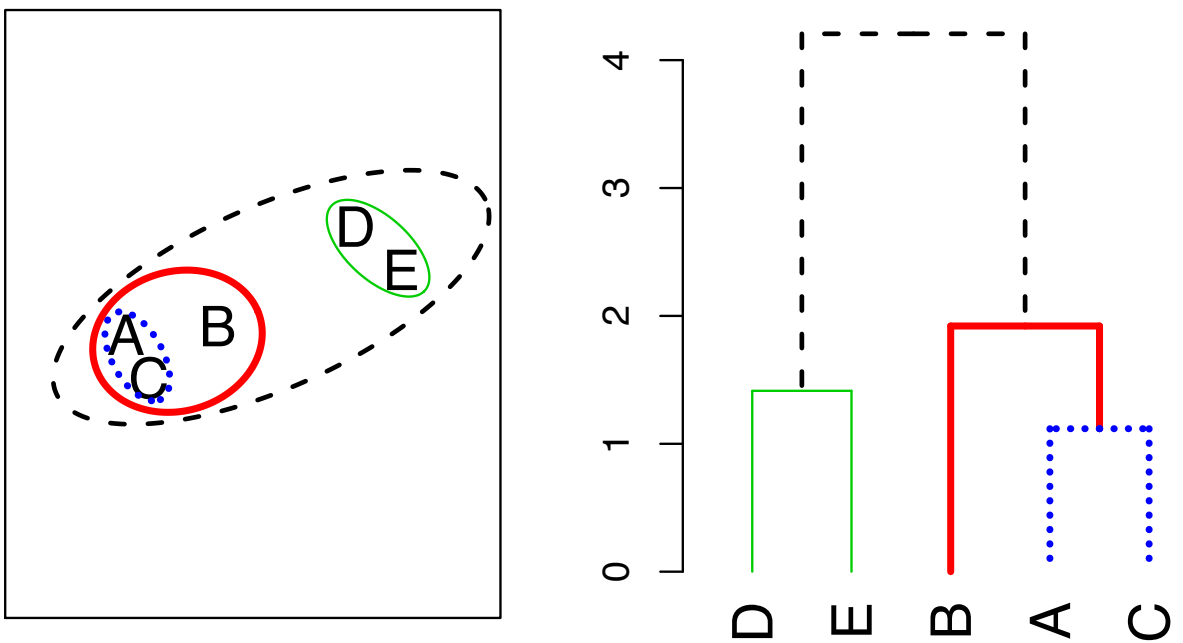
Builds a hierarchy in a "bottom-up" fashion...



Hierarchical Clustering Algorithm

The approach in words:

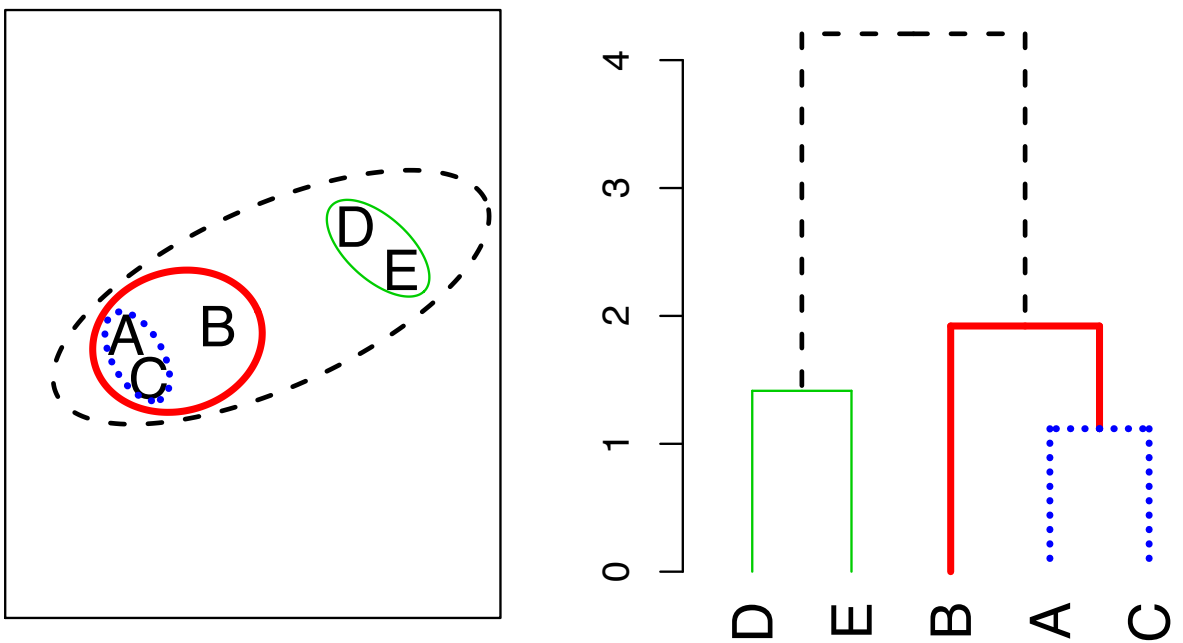
- Start with each point in its own cluster.
- Identify the closest two clusters and merge them.
- Repeat.
- Ends when all points are in a single cluster.



Hierarchical Clustering Algorithm

The approach in words:

- Start with each point in its own cluster.
- Identify the closest two clusters and merge them.
- Repeat.
- Ends when all points are in a single cluster.



基于层次的聚类

例 12-3 在例 12-2 的设定下，样本间距离采用欧氏距离的平方，类间链接方式选择单链接法，试用聚合型层次聚类算法对样本进行聚类

解：基于例 12-2 的矩阵 $\mathbf{X}$，计算样本间的距离矩阵 $\mathbf{D}$

$$\mathbf{D} = \begin{bmatrix} 0 & 8 & 11 & 26 & 38 \\ 8 & 0 & 3 & 22 & 38 \\ 11 & 3 & 0 & 9 & 21 \\ 26 & 22 & 9 & 0 & 6 \\ 38 & 38 & 21 & 6 & 0 \end{bmatrix}^{\top}$$

(1) 基于 5 个样本构建 5 个初始类，其中 $C_i = \{i\}, i = 1, \ldots, 5$，由于此时每个类别仅含 1 个样本，因此类间距离等于样本间距离；

(2) 根据矩阵 $\mathbf{D}$，$d_{23} = d_{32} = 3$ 最小，因此将 C_2 和 C_3 合并为一个新类 $C_6 = \{2, 3\}$；

基于划分的聚类

解：

(3) 分别计算 C_6 和 C_1, C_4, C_5 间的距离：

- 由于 $d_{21} = 8, d_{31} = 11$，因此 $d_{61} = 8$;
- 由于 $d_{24} = 22, d_{34} = 9$，因此 $d_{64} = 9$;
- 由于 $d_{25} = 38, d_{35} = 21$，因此 $d_{65} = 21$;
- 其余两类间的距离分别为：$d_{14} = 26, d_{15} = 38, d_{45} = 6$;

(4) 由于 $d_{45} = 6$ 最小，因此将 C_4, C_5 合并为一个新类 $C_7 = \{4, 5\}$

(5) 分别计算 C_6, C_7 和 C_1 间的距离：

- 由于 $d_{24} = 22, d_{25} = 38, d_{34} = 9, d_{35} = 21$，因此 $d_{67} = 9$;
- 由于 $d_{21} = 8, d_{31} = 11$，因此 $d_{61} = 8$;
- 由于 $d_{41} = 26, d_{51} = 38$，因此 $d_{71} = 26$;

基于划分的聚类

解：

(6) 由于 $d_{61}=8$ 最小，因此将 C_6 和 C_1 合并为一个新类 $C_8=\{1,2,3\}$；

(7) 将 C_7 和 C_8 合并为一个新类 $C_9=\{1,2,3,4,5\}$，至此所有样本归为 1 类，聚类终止。

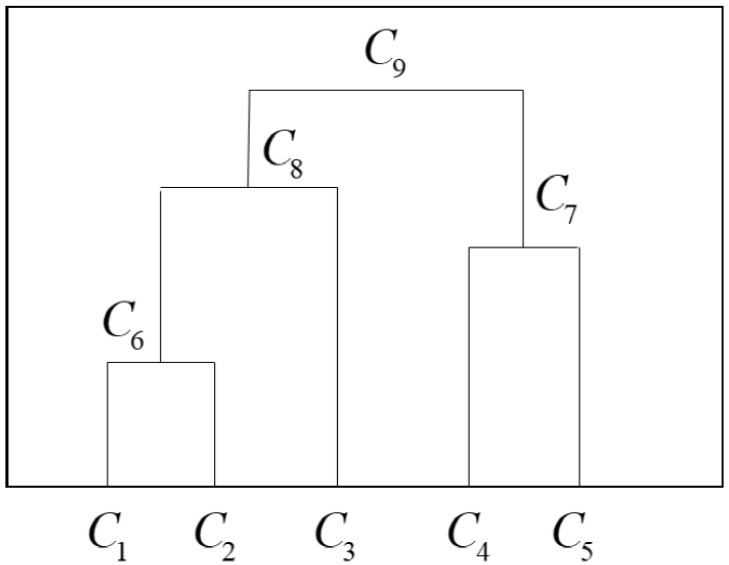


图 12-9　例 12-3 的聚合型层次聚类树状图

Hierachical Clustering

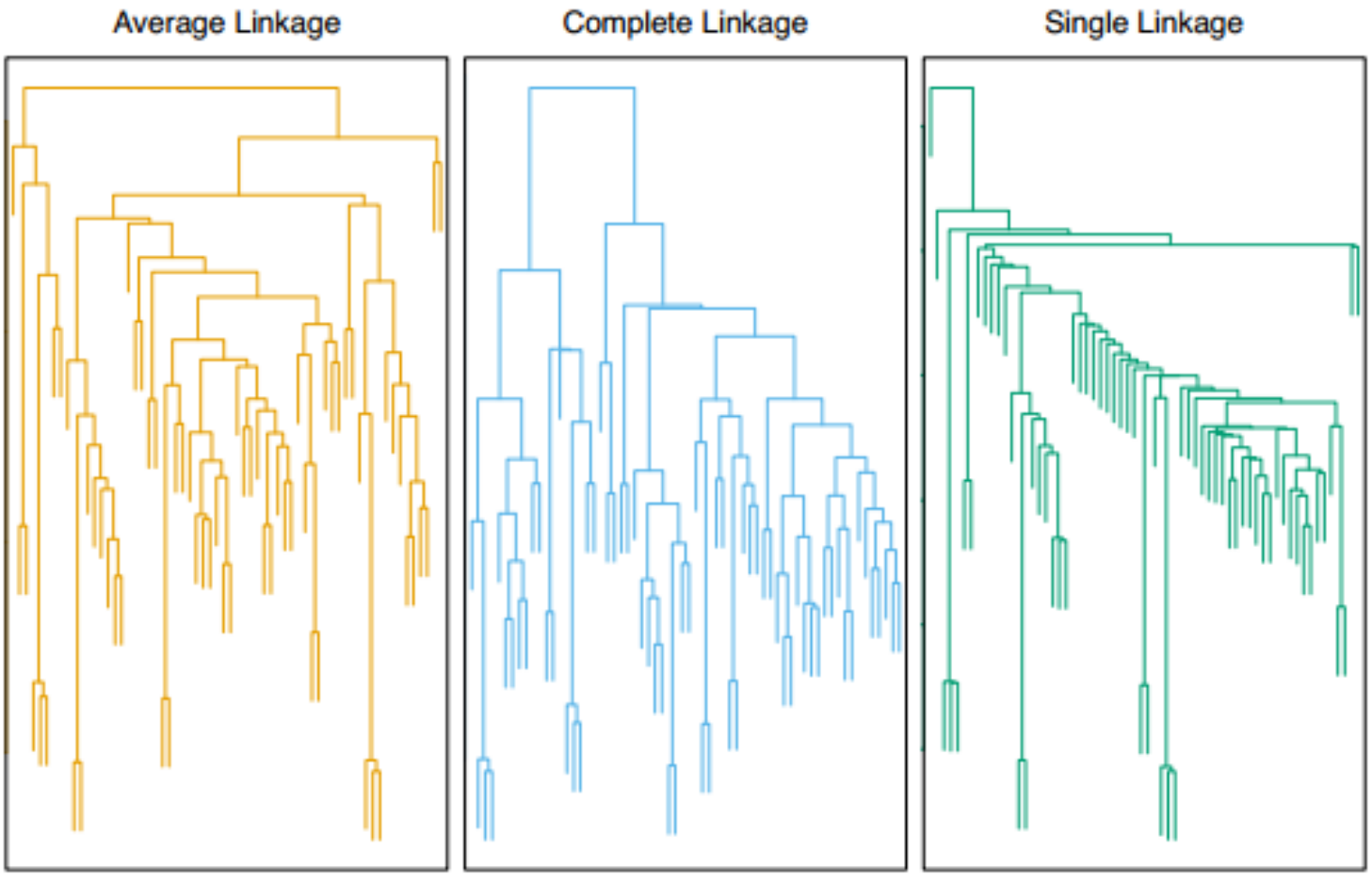
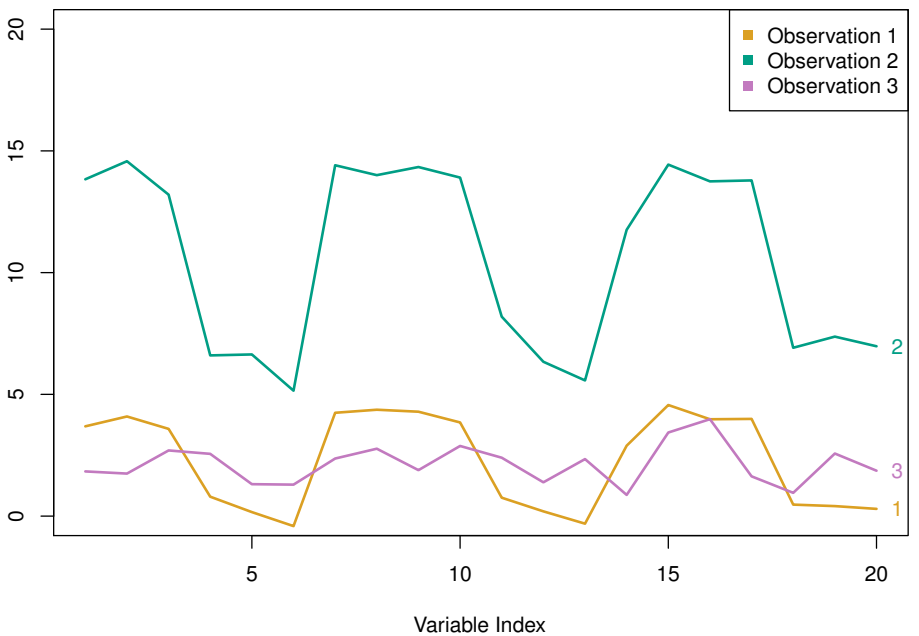


FIGURE 10.12. *Average, complete, and single linkage applied to an example data set. Average and complete linkage tend to yield more balanced clusters.*

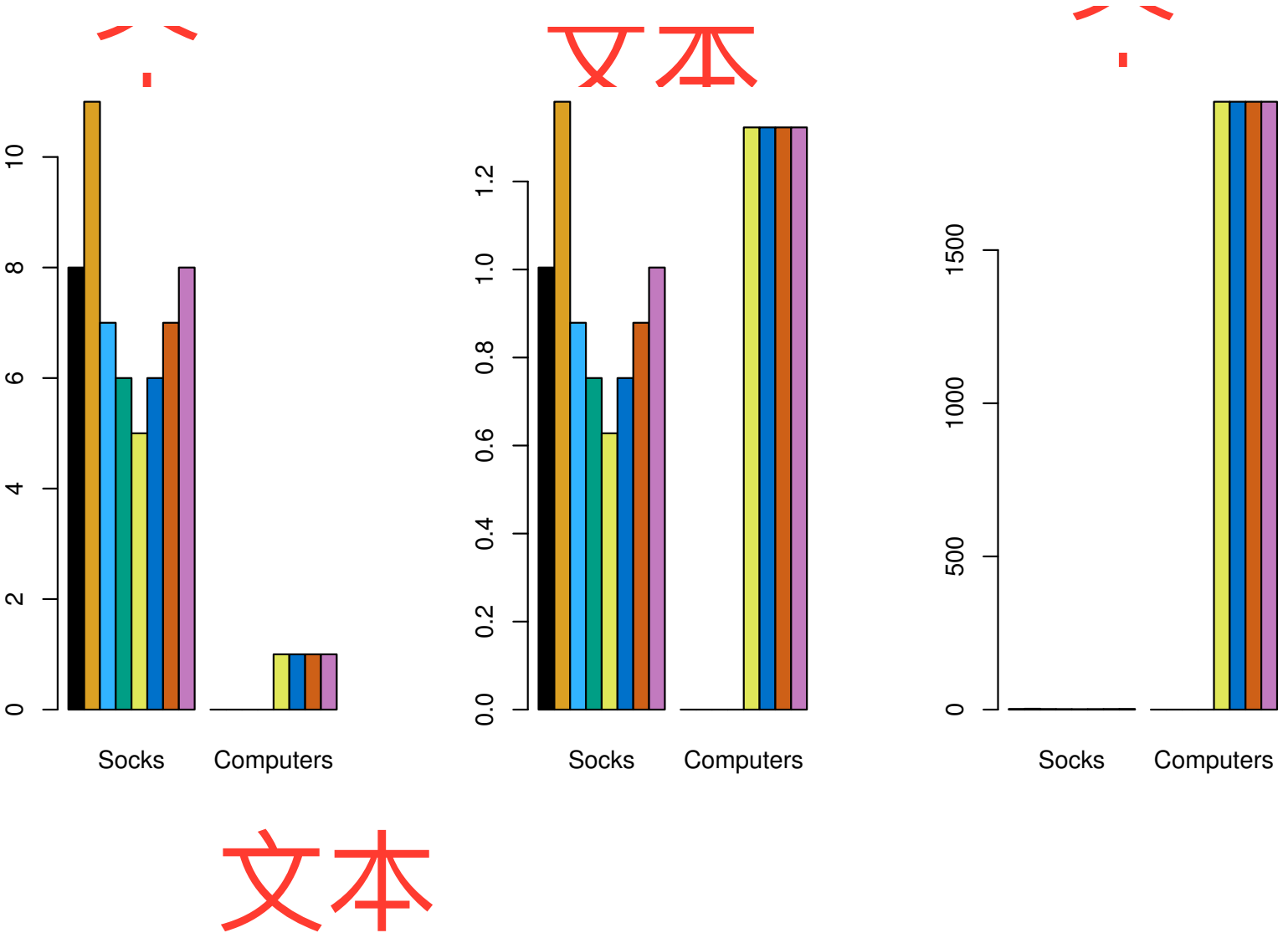
Choice of Dissimilarity Measure

- So far have used Euclidean distance.
- An alternative is *correlation-based distance* which considers two observations to be similar if their features are highly correlated.
- This is an unusual use of correlation, which is normally computed between variables; here it is computed between the observation profiles for each pair of observations.

文本
文本



Scaling of the variables matters



基于层次的聚类

基于层次聚类和基于划分聚类的异同

- 两者本质上都是以距离的远近亲疏作为标准进行聚类的；
- 基于划分的聚类法只能产生指定类数的聚类结果，具体类数的确定依赖于经验的积累，而层次聚类法可直接产生一系列的聚类结果；
- 层次聚类法不具有很好的可延展性，由于它在合并类时需要检查和估算大量的对象或类，因此当样本量 n 很大时不是很适用；而基于划分的聚类算法中，K-means 和 K-modes 则是具有较高的可延展性，因为它们的复杂度近乎线性；

基于模型的聚类

基于模型的聚类

- 基于划分的聚类和基于层次的聚类都是以距离度量作为基础。
- 本节介绍的基于模型的聚类则是以事先定义的数学模型作为聚类的基础。该聚类假设数据是通过已定义的模型来生成，借助优化手段使得给定的数据尽量拟合已定义的模型，从而得到模型的相应参数，既而确定各个类别的特点。
- 目前基于模型的聚类主要有两类方法，一类是基于概率统计的模型聚类，另一类则是基于神经网络的模型聚类。本节主要介绍基于概率统计的模型聚类

基于模型的聚类

基于概率统计的模型聚类

基于概率统计的模型聚类一般假设总体服从一个有限混合模型(finite mixture model)，该有限混合模型（总体分布）由各个混合成分（子分布）混合而成，每个混合成分代表其中一个类别的样本分布特征，通过概率的相关理论从而推出每个混合成分的特征（例如均值和方差等），并且将每个观测样本归属于后验概率最大的混合成分对应的类别。

基于模型的聚类

有限混合模型

- 给定一个含有 n 个样本 $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathbb{R}^p$ 的数据集，目的是将这些样本聚为 K 类。假设这些观测样本是随机向量 $\mathbf{x} \in \mathbb{R}^p$ 的独立实现值，并且假设样本的类别标签 $\{z_1, \ldots, z_n\} \in \mathbb{R}$ 是随机变量 $z \in \{1, \ldots, K\}$ 的独立实现值，那么我们可以用 $\{(\mathbf{x}_i, z_i)\}_{i=1}^n$ 来表示这个完整的数据集。这里称该数据集为完整数据集，是因为同时考虑了观测到的样本以及无法观测到的样本类别标签。
- 将 $\mathbf{x}$ 的概率密度函数用 $g(\mathbf{x})$ 来表示，则有限混合模型如下：

$$g(\mathbf{x}) = \sum_{k=1}^{K} \pi_k f_k(\mathbf{x}) \quad s.t. \sum_{k=1}^{K} \pi_k = 1 \tag{12.6}$$

- 其中，π_k 表示总体分布中第 k 个子分布所占的比例，$f_k(\mathbf{x})$ 表示第 k 个子分布的条件概率密度函数。

基于模型的聚类

有限混合模型

- 这里一般假设不同的子分布具有相同的分布形式，只是代表分布特征的参数不同，因此 (12.6) 可以重新写成下式：

$$g(\mathbf{x}) = \sum_{k=1}^{K} \pi_k f(\mathbf{x}; \boldsymbol{\theta}_k) \tag{12.7}$$

- 其中，$\boldsymbol{\theta}_k$ 代表第 k 个子分布的参数向量。对于这样一个数据集 $\mathbf{x} = \{x_1, \ldots, x_n\}$，可以写成这个混合模型的对数似然函数：

$$\ell(\boldsymbol{\theta}; \mathbf{x}) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k f(\mathbf{x}_i; \boldsymbol{\theta}_k) \right) \tag{12.8}$$

基于模型的聚类

有限混合模型

- 由于样本的类别标签未知，这里引入完整的对数似然函数：

$$\ell_c(\boldsymbol{\theta}; \mathbf{x}, \mathbf{z}) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \log \Big(\pi_k f(\mathbf{x}_i; \boldsymbol{\theta}_k) \Big) \tag{12.9}$$

- 其中，如果第 i 个样本属于第 k 类则 $z_{ik} = 1$，反之则为 0。

基于模型的聚类

EM 算法

- EM 算法是目前解决有限混合模型最常用的方法之一，它通过不断迭代来最大化完整对数似然的条件期望来得到最终的结果，完整对数似然的条件期望定义如下：

$$\mathrm{E}[\ell_c(\boldsymbol{\theta};\mathbf{x},\mathbf{z})|\boldsymbol{\theta}^*] = \sum_{k=1}^{K}\sum_{i=1}^{n} t_{ik}\log\Big(\pi_k f(\mathbf{x}_i;\boldsymbol{\theta}_k)\Big) \qquad (12.10)$$

- 其中，$t_{ik} = \mathrm{E}[z = k|\mathbf{x}_i, \boldsymbol{\theta}^*]$，$\boldsymbol{\theta}^*$ 是给定的混合参数集合。
- 通过给定一个初始解 $\boldsymbol{\theta}^{(0)}$，EM 算法交替执行 E 步和 M 步。

(1) E 步：对完整的对数似然函数求条件期望，即在现有的参数集合 $\boldsymbol{\theta}^{(q)}$ 下求得 $\mathrm{E}[\ell_c(\boldsymbol{\theta};\mathbf{x},\mathbf{z})|\boldsymbol{\theta}^{(q)}]$

(2) M 步：最大化 $\mathrm{E}[\ell_c(\boldsymbol{\theta};\mathbf{x},\mathbf{z})|\boldsymbol{\theta}^{(q)}]$，即

$$\boldsymbol{\theta}^{(q+1)} = \max_{\boldsymbol{\theta}} \ \mathrm{E}[\ell_c(\boldsymbol{\theta};\mathbf{x},\mathbf{z})|\boldsymbol{\theta}^{(q)}] \qquad (12.11)$$

基于模型的聚类

EM 算法

- 通过给定一个初始解 $\boldsymbol{\theta}^{(0)}$，EM 算法交替执行 E 步和 M 步。

  (1) E 步：对完整的对数似然函数求条件期望，即在现有的参数集合 $\boldsymbol{\theta}^{(q)}$ 下求得 $\mathrm{E}[\ell_c(\boldsymbol{\theta};\mathbf{x},\mathbf{z})|\boldsymbol{\theta}^{(q)}]$
  (2) M 步：最大化 $\mathrm{E}[\ell_c(\boldsymbol{\theta};\mathbf{x},\mathbf{z})|\boldsymbol{\theta}^{(q)}]$，即

$$\boldsymbol{\theta}^{(q+1)} = \max_{\boldsymbol{\theta}} \ \mathrm{E}[\ell_c(\boldsymbol{\theta};\mathbf{x},\mathbf{z})|\boldsymbol{\theta}^{(q)}] \qquad (12.11)$$

- 以上两个步骤交替进行，直到停止条件得到满足，一般我们采用的停止条件为：$|\ell(\boldsymbol{\theta}^{(q+1)};\mathbf{x}) - \ell(\boldsymbol{\theta}^{(q)};\mathbf{x})| < \epsilon$，这里 ϵ 是一个很小的事先给定的正数。
- EM 算法的一个优良性质是，该算法可以保证似然函数在每次迭代过程都得到提升，因此可以证明该算法可以收敛到局部最优解。

基于模型的聚类

高斯混合模型

- 有限混合模型中的混合成分可选择多种概率分布形式，例如高斯分布、偏正态分布、非对称拉普拉斯分布、t 分布等。
- 在这些分布当中，高斯分布由于理论上和计算上的优势，从而成为最广泛使用的分布形式。在高斯分布的模型假设下，每个混合成分的概率密度函数 $f(\mathbf{x};\boldsymbol{\theta}_k)$ 一般设定为一个多维高斯密度函数 $\phi(\mathbf{x};\boldsymbol{\theta}_k)$，该密度函数的特征参数为均值 $\boldsymbol{\mu}_k$ 和协方差矩阵 $\boldsymbol{\Sigma}_k$。
- 因此 $\mathbf{x}$ 的概率密度函数可以写成：

$$g(\mathbf{x};\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \phi(\mathbf{x};\boldsymbol{\theta}_k) \tag{12.12}$$

$$\phi(\mathbf{x};\boldsymbol{\theta}_k) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}_k|^{1/2}} \exp\left\{ -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\top}\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\} \tag{12.13}$$

基于模型的聚类

高斯混合模型

基于高斯假设下的该模型称为高斯混合模型 (GMM)，基于这个模型假设，上一节介绍的 EM 算法有如下的具体形式：

(1) E 步：给定当前的参数集合 $\boldsymbol{\theta}^{(q-1)}$，计算完整对数似然函数的期望。实际上，这个过程退化为计算

$$t_{ik}^{(q)} = \mathsf{E}[z_{ik}|\mathbf{x}_i, \boldsymbol{\theta}^{(q-1)}] = \mathsf{P}(z_i = k|\boldsymbol{\theta}^{(q-1)}) \tag{12.14}$$

- 其中，$\mathrm{P}(z_i = k|\boldsymbol{\theta}^{(q-1)})$ 表示给定参数集合 $\boldsymbol{\theta}^{(q-1)}$，观测样本 $\mathbf{x}_i$ 属于高斯混合分布中第 k 个成分的后验概率。
- 其中，$\mathrm{P}(z_i = k|\boldsymbol{\theta}^{(q-1)})$ 表示给定参数集合 $\boldsymbol{\theta}^{(q-1)}$，观测样本 $\mathbf{x}_i$ 属于高斯混合分布中第 k 个成分的后验概率。根据贝叶斯定理，我们计算该后验概率 $t_{ik}^{(q)}$，$i = 1, \ldots, n, k = 1, \ldots, K$

$$t_{ik}^{(q)} = \frac{\pi_k^{(q-1)} \phi(\mathbf{x}_i, \boldsymbol{\theta}_k^{(q-1)})}{\sum_{l=1}^{K} \pi_l^{(q-1)} \phi(\mathbf{x}_i, \boldsymbol{\theta}_l^{(q-1)})} \tag{12.15}$$

基于模型的聚类

高斯混合模型

(2) M 步：最大化 $\mathrm{E}[\ell_c(\boldsymbol{\theta};\mathbf{x},\mathbf{z})|\boldsymbol{\theta}^{(q-1)}]$，将样本 $\mathbf{x}_i$ 归为后验概率最大的类，从而形成混合比例 π_k，均值 $\boldsymbol{\mu}_k$ 和协方差 $\boldsymbol{\Sigma}_k$ 的一个更新，如下所示，对 $k=1,\ldots,K$

$$n_k^{(q)}=\sum_{i=1}^{n} z_{ik}^{(q)} \quad \widehat{\pi}_k^{(q)}=\frac{n_k^{(q)}}{n}$$

$$\widehat{\boldsymbol{\mu}}_k^{(q)}=\frac{1}{n_k^{(q)}}\sum_{i=1}^{n} z_{ik}^{(q)}\mathbf{x}_i$$

$$\widehat{\boldsymbol{\Sigma}}_k^{(q)}=\frac{1}{n_k^{(q)}}\sum_{i=1}^{n} z_{ik}^{(q)}(\mathbf{x}_i-\widehat{\boldsymbol{\mu}}_k^{(q)})(\mathbf{x}_i-\widehat{\boldsymbol{\mu}}_k^{(q)})^{\top}$$

基于图论的聚类

谱聚类 *(spectral clustering)*

- 通过前面的叙述，我们依次介绍了基于划分的聚类、基于层次的聚类以及基于模型的聚类三种不同的聚类方法。其中，基于划分的聚类和基于层次的聚类的基本思想是通过“距离”来刻画类别的相似性，而基于模型的聚类则是用分布来拟合数据，从而根据所属分布的后验概率和分布的中心，来确定所属的类别和类别的位置特征。这三种聚类方法原则上对凸形状的类具有较好的聚类效果，但是现实中存在许多非凸形状的类别，此时应用以上三类方法往往得到较差结果。
- 现在我们将介绍谱聚类 (spectral clustering)，该方法建立在图论的基础之上，将数据样本间的关系以图形的方式呈现，并对图形进行分割，从而得到适合样本空间中任意形状的子图（即子类）。

基于图论的聚类

图形表示

- 给定一个含有 n 个样本 $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathbb{R}^p$ 的数据集，将其以无向图进行表示，标记为 $G = (V, E)$。
- 顶点集合为 $V = \{v_1, \ldots, v_n\}$，每个顶点代表一个样本。
- 边的集合为 $E = \{e_{ij}; i, j = 1, \ldots, n\}$，边 e_{ij} 表示顶点 i 和顶点 j 相连，并赋予其权重 $w_{ij} \geq 0$，进而构筑邻接矩阵 (adjacency matrix)$\mathbf{W} = \{w_{ij}\}_{i,j=1,\ldots,n}$，其中 $w_{ij} = 0$ 意味着顶点 i 和 j 不相连，基于无向图的需要，这里规定 $w_{ij} = w_{ji}$。

基于图论的聚类

拉普拉斯矩阵的性质

- 对于节点 $v_i \in V, i = 1, \ldots, n$，定义它的度 (degree) 为：$d_i = \sum_{j=1}^{n} w_{ij}$，从而得到度矩阵 (degree matrix) $\mathbf{D} = \mathrm{diag}(d_1, \ldots, d_n)$。
- 定义如下的拉普拉斯矩阵 (Laplacian matrix)：$\mathbf{L} = \mathbf{D} - \mathbf{W}$。
- 性质 12-1($\mathbf{L}$ 的基本性质)

  (1) 对于任意的向量 $\boldsymbol{f} \in \mathbb{R}^n$，可以证明

$$\boldsymbol{f}^\top \mathbf{L} \boldsymbol{f} = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2$$

  (2) $\mathbf{L}$ 是对称的半正定矩阵。

  (3) $\mathbf{L}$ 的最小特征值为 0，且其对应的特征向量为 $\mathbf{1}_n$。

基于图论的聚类

证明：

(1) 根据 d_i 的定义，我们有

$$\begin{aligned}\boldsymbol{f}^\top \mathbf{L}\boldsymbol{f} = \boldsymbol{f}^\top \mathbf{D}\boldsymbol{f} - \boldsymbol{f}^\top \mathbf{W}\boldsymbol{f} &= \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} f_i f_j w_{ij} \\ &= \frac{1}{2}\Big(\sum_{i=1}^{n} d_i f_i^2 - 2\sum_{i,j=1}^{n} f_i f_j w_{ij} + \sum_{j=1}^{n} d_j f_j^2\Big) \\ &= \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2\end{aligned}$$

(2) $\mathbf{L}$ 的对称性是根据 $\mathbf{D}$ 和 $\mathbf{W}$ 的对称性得到，至于 $\mathbf{L}$ 的半正定性则由于 (1) 中的式子非负。(3) 和 (4) 可由 (1) 和 (2) 的结果直接得到。

基于图论的聚类

标准化拉普拉斯矩阵

- 在谱聚类中，一般需要对拉普拉斯矩阵 $\mathbf{L}$ 进行标准化处理，得到标准化的拉普拉斯矩阵。
- 根据标准化的方式可以分为：随机游走标准化拉普拉斯矩阵 (random walk normalized graph Laplacian matrix) $\mathbf{L}_{\mathrm{rw}}$ 和对称标准化拉普拉斯矩阵 (symmetric normalized graph Laplacian matrix) $\mathbf{L}_{\mathrm{sym}}$

$$\mathbf{L}_{\mathsf{rw}} = \mathbf{D}^{-1}\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}$$

$$\mathbf{L}_{\mathsf{sym}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$$

基于图论的聚类

谱聚类的直观原理

- 当数据样本呈现的无向图具有 K 个子图，并且每个子图内部的顶点互相连接，即 $w_{ij} > 0$，而任意两个不同子图中的顶点互不相连，即 $w_{ij} = 0$，此时时拉普拉斯矩阵经过排列之后，必然是一个块对角矩阵。
- 其中，第 k 个块对角矩阵对应第 k 个子图：$G_k = (V_k, E_k)$，从而有 1 个 0 特征值，且其对应的特征向量为 $\mathbf{1}_{n_k}$，n_k 为第 k 个子图含有的顶点个数。因此原拉普拉斯矩阵有 K 个个 0 特征值，且其对应的特征向量为各个子图的示性向量。

基于图论的聚类

邻接矩阵的 *4* 种构建方法

- 全连接图 (fully-connected graph)
  在全连接图中，所有的顶点互相连接，并且每条边都赋予一个正的权重，例如使用高斯核函数：

$$w_{ij} = \exp\left(\frac{-||\mathbf{x}_i - \mathbf{x}_j||_2^2}{\sigma^2}\right)$$

  其中，$\sigma > 0$ 是一个给定的参数。
- k 最近邻图 (k-nearest neighbor graph)
  在 k 最近邻图中，当顶点 v_i 属于 v_j 的 k 最近邻或顶点 v_j 属于 v_i 的 k 最近邻时，顶点 v_i 与 v_j 相连，即 $w_{ij} = 1$。

基于图论的聚类

邻接矩阵的 *4* 种构建方法

- 互为 k 最近邻图 (mutual k-nearest neighbor graph)
  在互为 k 最近邻图中，当顶点 v_i 属于 v_j 的 k 最近邻且顶点 v_j 属于 v_i 的 k 最近邻时，顶点 v_i 与 v_j 相连，即 $w_{ij} = 1$。
- ϵ 邻域图 (ϵ-neighbor graph)
  在 ϵ 邻域图中，当顶点 v_i 与 v_j 之间的距离小于 ϵ 时，顶点 v_i 与 v_j 相连，即 $w_{ij} = 1$

基于图论的聚类

谱聚类算法

- 经典的谱聚类算法包括三种：非标准化谱聚类 (算法 12.5)、基于随机游走标准化拉普拉斯矩阵的谱聚类算法 (Shi and Malik, 2000) 和基于对称标准化拉普拉斯矩的谱聚类算法 (Ng, Jordan and Weiss, 2002)

算法 12.5 非标准化谱聚类算法

1. 根据 n 个样本，选择合适的方法构造邻接矩阵 $\mathbf{W}$，并计算拉普拉斯矩阵 $\mathbf{L}$。
2. 计算 $\mathbf{L}$ 的前 K 个特征值对应的特征向量 $u_1,\ldots u_K$。
3. 构建一个 $n\times K$ 的数据矩阵 $\mathbf{U}$，该矩阵的第 j 列对应第二步的特征向量 u_j。
4. 将矩阵 $\mathbf{U}$ 的第 i 行视为第 i 个样本，因此总共得到 n 个样本，对这些样本运用 K-means 聚类算法进行聚类，最终得到 K 个子类。

图 12-10　非标准化谱聚类算法

基于图论的聚类

谱聚类的工作原理

- 通过观察谱聚类的算法，其基本流程需要先对拉普拉斯矩阵进行特征分解，得到特征向量，进而构造新的数据矩阵，从而将原样本空间映射到新的空间，并应用 K-means 聚类算法进行聚类。那么为何这样的方式可以准确地将不同类别的样本进行划分？这里参考 Luxburg(2007) 进而探究谱聚类背后的数学原理。
- 对于邻接矩阵 $\mathbf{W}$，令 $\overline{A}$ 表示 A 的补集，且定义

$$W(A,B) := \sum_{i \in A, j \in B} w_{ij}$$

- 给定聚类个数 K，那么对于这个无向图，我们需要找到一个合理的分割，使得某种损失最小

基于图论的聚类

谱聚类的工作原理

- 两种最常用的分割损失分别是 RatioCut 和 Ncut，定义如下：

$$\mathsf{Cut}(A_1, \ldots, A_K) := \frac{1}{2} \sum_{k=1}^{K} W(A_k, \overline{A}_k)$$

$$\mathsf{RatioCut}(A_1, \ldots, A_K) := \frac{1}{2} \sum_{k=1}^{K} \frac{W(A_k, \overline{A}_k)}{|A_k|} = \sum_{k=1}^{K} \frac{\mathsf{Cut}(A_k, \overline{A}_k)}{|A_k|}$$

$$\mathsf{Ncut}(A_1, \ldots, A_K) := \frac{1}{2} \sum_{k=1}^{K} \frac{W(A_k, \overline{A}_k)}{\mathsf{vol}(A_k)} = \sum_{k=1}^{K} \frac{\mathsf{Cut}(A_k, \overline{A}_k)}{\mathsf{vol}(A_k)}$$

- 其中，$|A_k|$ 表示集合 A_k 中的顶点个数，$\mathrm{vol}(A_k) = \sum_{i \in A_k} d_i$。

基于图论的聚类

谱聚类的工作原理

接下来将以 RatioCut 作为损失函数，分别推导聚类个数在 $K = 2$ 和 $K > 2$ 的情况下，对该损失函数如何优化求解。

(1) RatioCut 的二分类近似

- 此时目标函数如下所示：

$$\min_{A \subset V} \quad \mathsf{RatioCut}(A, \overline{A}) \tag{12.16}$$

- 对于集合 $A \subset V$，定义向量 $\boldsymbol{f} = (f_1, \ldots, f_n)^\top \in \mathbb{R}^n$，其中第 i 个元素定义为

$$f_i = \begin{cases} \sqrt{|\overline{A}|/|A|}, & v_i \in A \\ -\sqrt{|A|/|\overline{A}|}, & v_i \in \overline{A} \end{cases} \tag{12.17}$$

基于图论的聚类

谱聚类的工作原理

- 结合前面定义的非标准化拉普拉斯矩阵，可得：

$$
\begin{aligned}
\boldsymbol{f}^{\top}\mathbf{L}\boldsymbol{f} &= \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2 \\
&= \frac{1}{2}\sum_{i\in A, j\in\overline{A}} w_{ij}\left(\sqrt{\frac{|\overline{A}|}{|A|}} + \sqrt{\frac{|A|}{|\overline{A}|}}\right)^2 \\
&\quad + \frac{1}{2}\sum_{i\in\overline{A}, j\in A} w_{ij}\left(-\sqrt{\frac{|\overline{A}|}{|A|}} - \sqrt{\frac{|A|}{|\overline{A}|}}\right)^2 \\
&= \mathsf{Cut}(A,\overline{A})\left(\frac{|\overline{A}|}{|A|} + \frac{|A|}{|\overline{A}|} + 2\right) \\
&= \mathsf{Cut}(A,\overline{A})\left(\frac{|\overline{A}|+|A|}{|A|} + \frac{|A|+|\overline{A}|}{|\overline{A}|}\right) \\
&= |V|\mathsf{RatioCut}(A,\overline{A})
\end{aligned}
$$

基于图论的聚类

谱聚类的工作原理

- 除此之外，我们可以得到如下关系：

$$\sum_{i=1}^{n} f_i = \sum_{i\in A}\sqrt{|\overline{A}|/|A|} - \sum_{i\in\overline{A}}\sqrt{|A|/|\overline{A}|}$$

$$= |A|\sqrt{|\overline{A}|/|A|} - |\overline{A}|\sqrt{|A|/|\overline{A}|} = 0$$

- 所以按照向量 f 的定义，它与向量 $\mathbf{1}_n$ 正交，另外可以观察到

$$||\boldsymbol{f}||_2^2 = \sum_{i=1}^{n} f_i^2 = |A|\frac{|\overline{A}|}{|A|} + |\overline{A}|\frac{|A|}{|\overline{A}|} = n$$

基于图论的聚类

谱聚类的工作原理

- 因此目标函数 (12.6) 等价于

$$\min_{A \subset U} \boldsymbol{f}^\top \mathbf{L} \boldsymbol{f} \quad s.t. \quad \boldsymbol{f} \perp \mathbf{1},\ f_i\text{定义如 (12.7)},\ ||\boldsymbol{f}||_2 = \sqrt{n} \tag{12.18}$$

- 由于目标函数 (12.18) 是个 NP 困难问题，因此我们通常对上面离散的约束进行放松，最后得到下面的目标函数

$$\min_{A \subset U} \boldsymbol{f}^\top \mathbf{L} \boldsymbol{f} \quad s.t. \quad \boldsymbol{f} \perp \mathbf{1},\ ||\boldsymbol{f}||_2 = \sqrt{n} \tag{12.19}$$

- 根据 Rayleigh-Ritz 定理，目标函数 (12.19) 的解为 $\mathbf{L}$ 第二小特征值对应的特征向量 (因为最小特征值为 0，对应的特征向量为 $\mathbf{1}_n$，不具有区分度。)

基于图论的聚类

谱聚类的工作原理

(2) RatioCut 的多分类近似

- 类似前文的二分类近似，假如要将顶点集合 V 划分为 K 个子集 $A_1, \ldots, A_K$，我们首先定义 K 个示性向量 $\boldsymbol{h}_k = (h_{1,k}, \ldots, h_{n,k})^\top$，其中

$$h_{i,k} = \begin{cases} 1/\sqrt{|A_k|}, & v_i \in A_k \\ 0, & v_i \notin A \end{cases} \quad (i = 1, \ldots, n; k = 1, \ldots, K) \tag{12.20}$$

- 基于此，我们可以构造一个 $n \times K$ 的矩阵 $\mathbf{H}$,，该矩阵的第 k 列对应示性向量 $\boldsymbol{h}_k$。可以发现矩阵 $\mathbf{H}$ 中的列互相正交，即 $\mathbf{H}^\top \mathbf{H} = \mathbf{I}$。类似上一小节的推导，我们可以得到以下两个等式

$$\mathbf{h}_k^\top \mathbf{L} h_k = \frac{\mathsf{Cut}(A_k, \overline{A}_k)}{|A_k|}, \quad \mathbf{h}_k^\top \mathbf{L} \mathbf{h}_k = (\mathbf{H}^\top \mathbf{L} \mathbf{H})_{kk}$$

基于图论的聚类

谱聚类的工作原理

- 因此可以得到

$$\begin{aligned}\mathsf{RatioCut}(A_1,\ldots,A_K) &= \sum_{k=1}^{K}\mathbf{h}_k^\top\mathbf{L}\mathbf{h}_k = \sum_{k=1}^{K}(\mathbf{H}^\top\mathbf{L}\mathbf{H})_{kk} \\ &= \mathsf{Tr}(\mathbf{H}^\top\mathbf{L}\mathbf{H})\end{aligned}$$

- 其中，$\mathrm{Tr}(\dot{})$ 表示矩阵的迹运算。综上所述，最小化 $\mathrm{RatioCut}(A_1,\ldots,A_K)$ 问题等价于如下问题

$$\min_{A_1,\ldots,A_K} \quad \mathsf{Tr}(\mathbf{H}^\top\mathbf{L}\mathbf{H}) \quad s.t.\ \mathbf{H}^\top\mathbf{H}=\mathbf{I},\ \mathbf{H}\text{ 定义为 (12.20)} \tag{12.21}$$

基于图论的聚类

谱聚类的工作原理

- 类似上一小节的条件放宽，最终的目标函数如下：

$$\min_{\mathbf{H}\in\mathbb{R}^{n\times K}} \mathrm{Tr}(\mathbf{H}^\top \mathbf{L}\mathbf{H}) \quad s.t.\ \mathbf{H}^\top\mathbf{H} = \mathbf{I} \tag{12.22}$$

- 目标函数 (12.22) 是一个标准的迹最小化问题，同根据 Rayleigh-Ritz 定理，该问题的解是矩阵 $\mathbf{L}$ 的前 K 个特征值对应的特征向量，其中第 k 个特征向量填充矩阵 $\mathbf{H}$ 的第 k 列。

Convex Clustering; CC

- K-means clustering and Hierarchical clustering are two widely used clustering methods.
- Due to the instability of these two methods, Lindsten et al. (2011) and Hocking et al. (2011) used a convex penalty, such as L_1 norm, to replace the hidden L_0 norm in these two methods.
- Because of the convex relaxation for these two methods, we call such new clustering method, convex clustering.

Convex Clustering; CC

- Chi and Lange (2015) consider the following objective function for convex clustering.

$$\underset{\mathbf{A}}{\operatorname{argmin}} \frac{1}{2}||\mathbf{X}-\mathbf{A}||_2^2 + \gamma \sum_{i<j} w_{i,j}||\mathbf{A}_{i\cdot} - \mathbf{A}_{j\cdot}||_q$$

- where $\mathbf{A}$ is the approximated matrix which consists of constant structures. $\mathbf{A}_{i\cdot}$ denotes the ith row of $\mathbf{A}$ and $||\cdot||_q$ is the L_q norm of a vector. $w_{i,j}$ is the non-negative weight between the ith row and the jth row.

Sparse Convex Clustering; SCC

- Wang et al. (2018) add a group sparsity-induced penalty to take variable selection and get clustering results simultaneously so that we can deal with high-dimensional data.

$$\underset{\mathbf{A}}{\operatorname{argmin}} \frac{1}{2}\sum_{j=1}^{p}||\mathbf{x}_j-\mathbf{a}_j||_2^2+\gamma_1\sum_{i<j}w_{i,j}||\mathbf{A}_{i\cdot}-\mathbf{A}_{j\cdot}||_2+\gamma_2\sum_{j=1}^{p}u_j||\mathbf{a}_j||_2$$

- where u_j is the non-negative weight on $\mathbf{a}_j$ and γ_2 is a tuning parameter to take variable selection.
- We can get important variables by shrinking the unimportant variables by the second penalty.

SCC with Simulated Data

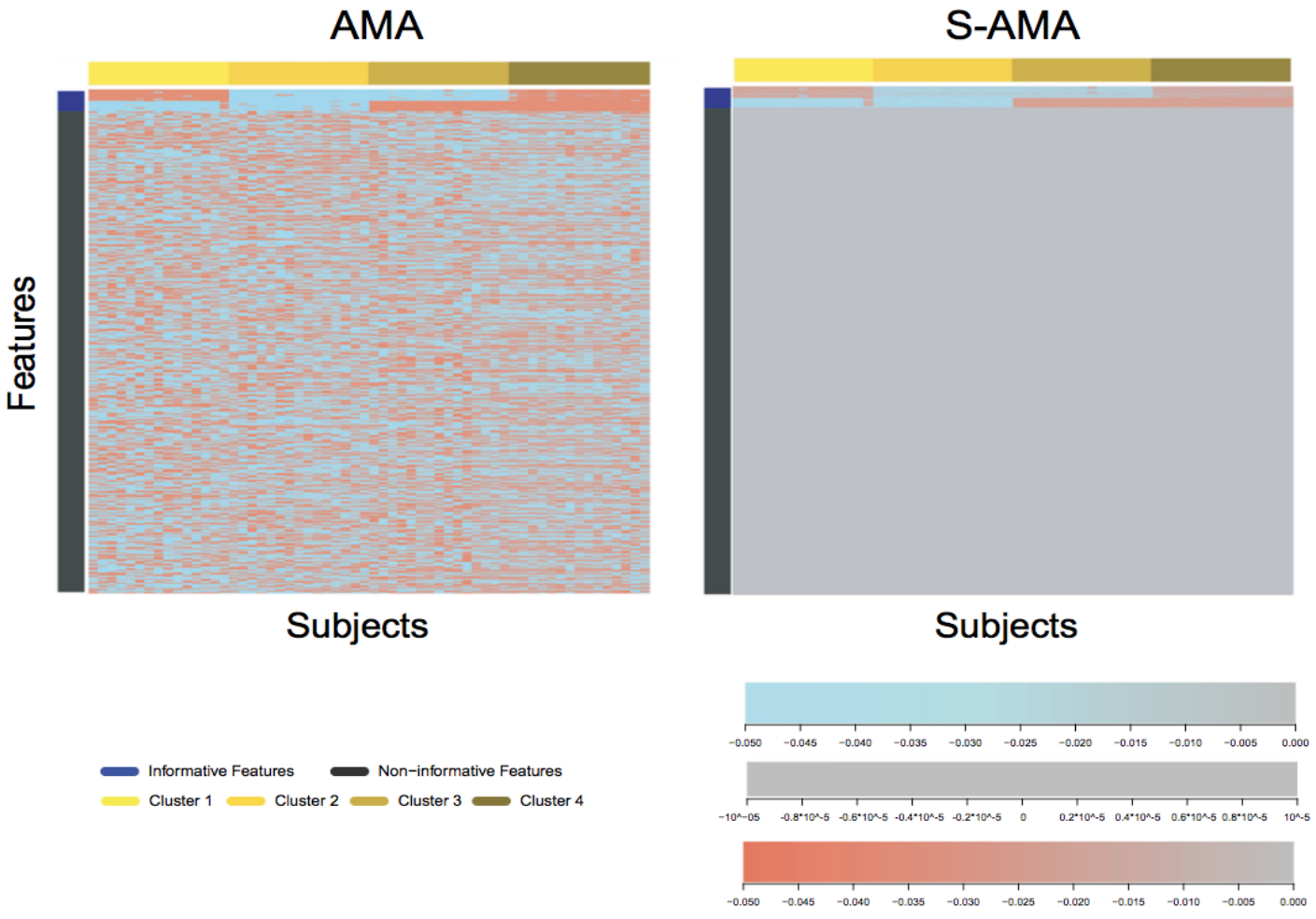


Figure 7: SCC with Simulated Data

Convex Biclustering; CBC

- The convex clustering can be extended to convex biclustering by adding the similar penalty on the pair-wise columns of the matrix.
- Different from SSVD, we want to penalize the difference value between different row vectors and the difference value between different column vectors simultaneously.
- Because we consider the simultaneous penalty on the rows and columns difference, we can finally get $K \times R$ constant submatrix.

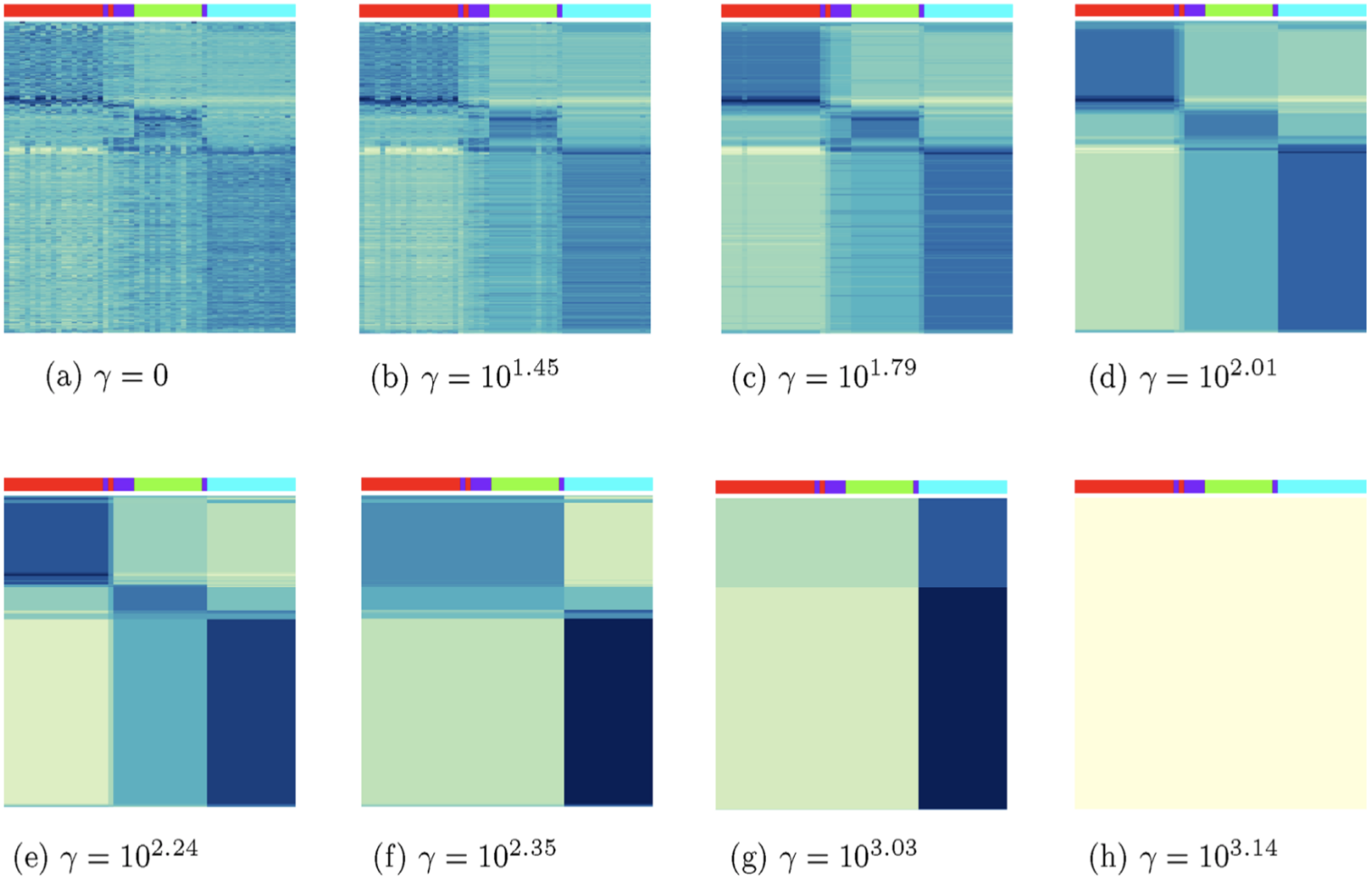
Convex Biclustering; CBC

- The detail objective function is as follows

$$F_\gamma(U) = \frac{1}{2}||X - U||_F^2 + \gamma[\Omega_W(U) + \Omega_W(U^\top)]$$

- Where $\Omega_W(U) = \sum_{i \leq j} w_{ij}||U_{.i} - U_{.j}||_2$, w_{ij} denotes the weights and $U_{.i}(U_{i.})$ denotes the ith column(row).
- If we delete one penalty from above two penalties, we may get a Convex Clustering problem.
- Authors propose to use Sparse Gaussian Kernel Weights as weights and use DLPA algorithm to turn the Convex Biclustering into Convex Clustering. Then we can use ADMM or AMA algorithm to get the final clusters.
- Note that, we can get some submatrixes of constant values.

CBC with Lung Cancer Data



(a) $\gamma = 0$ (b) $\gamma = 10^{1.45}$ (c) $\gamma = 10^{1.79}$ (d) $\gamma = 10^{2.01}$

(e) $\gamma = 10^{2.24}$ (f) $\gamma = 10^{2.35}$ (g) $\gamma = 10^{3.03}$ (h) $\gamma = 10^{3.14}$

Figure 8: CBC with Lung Cancer Data