第八章 非监督学习方法

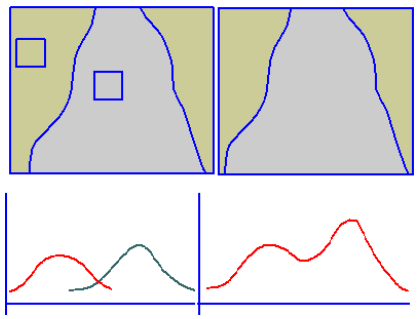
2009-12-8

8.0 引言

- **有监督学习** (supervised learning)：用已知类别的样本训练分类器，以求对训练集数据达到某种最优，并能推广到对新数据的分类。
- **非监督学习** (unsupervised learning)：样本数据类别未知，需要根据样本间的相似性对样本集进行分类 (聚类，clustering)。
  - 在一堆数据中寻找一种“自然分组”（C 组）。要求同组（类别）的样本较为相似，而不同组的样本间有明显不同。

8.0 引言

- **监督与非监督学习方法比较**



8.0 引言

- **监督与非监督学习方法比较**
  - 监督学习方法必须要有训练集与测试样本。在训练集中找规律，而对测试样本使用这种规律；而非监督学习只有一组数据，在该组数据集内寻找规律。
  - 监督学习方法的目的是识别事物，给待识别数据加上标号，因此训练样本集必须由带标号的样本组成。而非监督学习方法只有要分析的数据集本身，没有标号。如果发现数据集呈现某种聚集性，则可按自然的聚集性分类，但不以与某种预先的分类标号对上号为目的。

8.0 引言

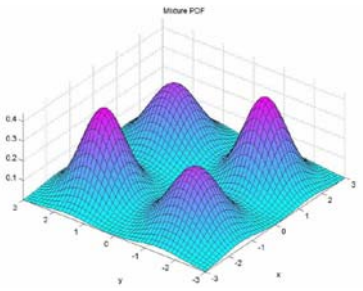
- **主要的非监督学习方法**
  - **基于概率密度函数估计的直接方法**：设法找到各类别在特征空间的分布参数再进行分类。比如直方图方法。
  - **基于样本间相似性度量的间接聚类方法**：设法定出不同类别的核心或初始类核，然后依据样本与各核心之间的相似性度量将样本聚集成不同类别。

8.0 引言

- 聚类是一个难以被严格定义的问题，因为“自然分组”本身就很抽象，且可能因人而异。
- 需要解决两个问题：
  - 如何度量样本之间的相似性？
  - 如何衡量某一种分组的好坏？（即目标函数）
- 寻找“最优分组”的计算复杂度太高，故一般的聚类算法都是近似算法。

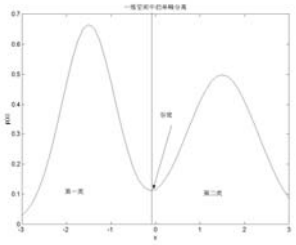
8.1 单峰子集的分离方法

- **基本思想**：把特征空间分为若干个区域，在每个区域上混合概率密度函数是单峰的，每个单峰区域对应一个类别。
- 例：如下图所示，x=0 和 y=0 这两个超平面可以把 (x,y) 平面分成四个单峰区域。



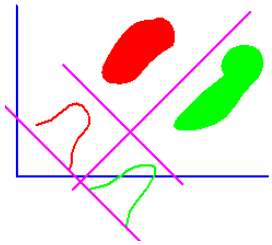
8.1.1 投影方法

- **一维空间中的单峰分离**
  - 对样本集 $K_N=\{x_i\}$ 应用直方图方法估计概率密度函数，找到概率密度函数的峰以及峰之间的谷底，以谷底为阈值对数据进行分割。



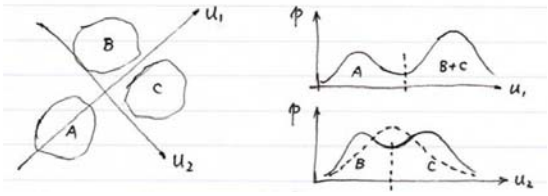
8.1.1 投影方法

- 多维空间 $\boldsymbol{y}$ 中直接划分成单峰区域比较困难，把它投影到一维空间 $\boldsymbol{x}$ 中简化问题（降维）。
- **基本原理**：考查样本某一分量的统计值，其分布往往呈现多峰形式；找到低谷，将峰值分别划分于不同的区域，每个区域只有一个高峰，从而把聚在同一高峰下的样本划分为一类。



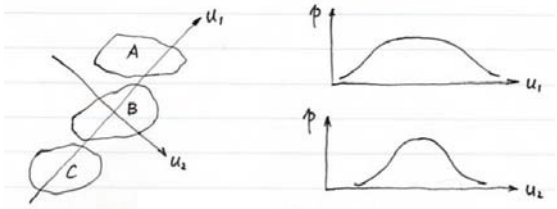
8.1.1 投影方法

- **确定合适的坐标系统 u**
  - 启发式的方法 — 使投影 $\{\mathbf{u}^T\mathbf{y}\}$ 的方差最大：方差越大，类之间分离的程度也可能越大；
  - 满足这样要求的 **u** 是样本协方差矩阵的最大特征值对应的特征向量。



8.1.1 投影方法

- **确定合适的坐标系统 u**
  - 存在问题：这样选择的 **u** 有时并不能产生多峰的边缘密度函数。



8.1.1 投影方法

- **算法步骤**

1. 计算样本 **y** 的混合协方差矩阵的最大特征值对应的特征向量**u**，把样本数据投影到**u**上 $(\mathbf{u}^T\mathbf{y})$;
2. 对投影后的数据，用直方图法求边缘概率密度函数;
3. 找到边缘概率密度函数的各谷点，在这些谷点上作垂直于**u**的超平面把数据划分成几个子集;
4. 如没有谷点，则用下一个最大的特征值代替;
5. 对所得到的各个子集进行同样的过程，直至每个子集都是单峰为止。

8.1.2 单峰子集分离的迭代算法

- 假设数据集 S 有一个划分

$$S=\bigcup_{i=1}^{c}\Gamma_i,$$

其中 Γ_i 互不相交，且 $|\Gamma_i|=N_i$, $\sum_i N_i=N=|S|$;

- 估计各类的加权类条件概率密度函数:

$$f(\mathbf{y}\,|\,\Gamma_i)=\frac{N_i}{N}p(\mathbf{y}\,|\,\Gamma_i);$$

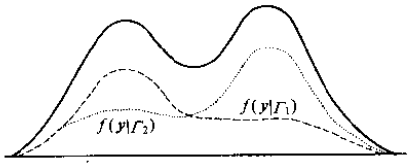
可用 Parzen 方法估计类条件概率密度

$$p(\mathbf{y}\,|\,\Gamma_i)=\frac{1}{N_i}\sum_{j=1}^{N_i}K(\mathbf{y},\mathbf{y}_i),\quad \mathbf{y}_i\in\Gamma_i.$$

8.1.2 单峰子集分离的迭代算法

- 考虑两个子集（类）的类条件概率密度函数加权估计值之间的“距离”

$$\int\left[f(\mathbf{y}\,|\,\Gamma_i)-f(\mathbf{y}\,|\,\Gamma_j)\right]^2 p(\mathbf{y})d\mathbf{y},$$



根据使两类的类条件概率密度函数加权估计值之间的“距离”最大进行类别划分

8.1.2 单峰子集分离的迭代算法

- 聚类准则：求子集划分能最大化

$$J=\frac{1}{2}\int\sum_{i=1}^{c}\sum_{j=1}^{c}[\,f(\mathbf{y}\,|\,\Gamma_i)-f(\mathbf{y}\,|\,\Gamma_j)]^2\,p(\mathbf{y})\,d\mathbf{y};$$

- **求解**
  - 考查某个样本 $\mathbf{y}_k$ 从 Γ_j 移入 Γ_i, 得到新的 $\tilde{\Gamma}_i$, $\tilde{\Gamma}_j$

$$f(\mathbf{y}\,|\,\tilde{\Gamma}_i)\ge f(\mathbf{y}\,|\,\Gamma_i),$$
$$f(\mathbf{y}\,|\,\tilde{\Gamma}_j)\le f(\mathbf{y}\,|\,\Gamma_j),$$

且 $\Delta f_i=-\Delta f_j=\frac{1}{N}K(\mathbf{y},\mathbf{y}_k);$

一般 N 较大；只有当 $\mathbf{y}$ 很接近 $\mathbf{y}_k$ 时，Δf_i 才能不接近于0。

8.1.2 单峰子集分离的迭代算法

- **求解**
  - 考虑 J 的变化量

$$\Delta J=\int\left[2c\Delta f_i\right]^2 p(\mathbf{y})d\mathbf{y}$$
$$+2c\int\left[\,f(\mathbf{y}\,|\,\Gamma_i)-f(\mathbf{y}\,|\,\Gamma_j)\right]\Delta f_i\,p(\mathbf{y})d\mathbf{y},$$

第一项恒大于0

第二项取决于 $f(\mathbf{y}|\,\Gamma_i)$-$f(\mathbf{y}|\,\Gamma_j)$：差越大，ΔJ 越大。
注意：当 $\mathbf{y}$ 不是 $\mathbf{y}_k$ 的近邻时，Δf_i 的值接近于0。

8.1.2 单峰子集分离的迭代算法

- **求解**
  - 通过把 $\mathbf{y}_k$ 从 Γ_j 移入 Γ_i，使得 J 增大，故移入时应该选择使 ΔJ 尽可能大的 Γ_i，即选择

$$f(\mathbf{y}_k\,|\,\Gamma_i)=\max_l f(\mathbf{y}_k\,|\,\Gamma_l),$$

从而使得 ΔJ 最大；

  - 如存在两个（或以上）子集的 $f(\mathbf{y}_k\,|\,\Gamma_i)$ 最大（相等），则可移入其中任意一类。

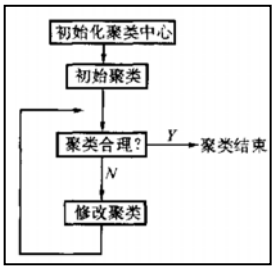
8.1.2 单峰子集分离的迭代算法

- **算法步骤**

1. 对数据集 S 选定一个初始划分；
2. 对 S 中的每一个样本 $\mathbf{y}$，逐一计算 $f(\mathbf{y}_k\,|\,\Gamma_i)$, 并把 $\mathbf{y}$ 重新分配到使得 $f(\mathbf{y}_k\,|\,\Gamma_i)$, 最大的子集中；
3. 如果有任何点进行了类别的转移，则重复上一步骤；直到不再有样本发生转移。

8.2 类别分离的间接方法

- **三个要点**:
  - 样本与样本/样本聚类间相似性的度量;
  - 准则函数: 聚类质量的判别标准;
  - 初始分类方法及迭代算法;
- **目标**:
  - 类内元素相似性高;
  - 类间元素相似性低。



8.2.1 C-均值算法（K-均值/means）

- **问题假设**: 对样本集 $K_N=\{\mathbf{x}_i\}$ 尚不知每个样本的类别，但可假设所有样本可分为 c 类，各类样本在特征空间依类聚集，且近似球形分布;
- **基本思路**: 用一代表点 (prototype) 来表示一个聚类，如类内均值 $\mathbf{m}_i$ 来代表聚类 K_i;
- **聚类准则**: 误差平方和 J

$$J = \sum_{i=1}^{c} \sum_{\mathbf{y}\in\Gamma_i} \|\mathbf{y} - \mathbf{m}_i\|^2 .$$

8.2.1 C-均值算法（K-均值/means）

- **求解**
  - 假设已有一个初始划分，考查 Γ_k 中的样本 $\mathbf{y}$，如把 $\mathbf{y}$ 移入 Γ_j，J 的改变量是

$$\Delta J = -\frac{N_k}{N_k - 1}\|\mathbf{y}-\mathbf{m}_k\|^2 + \frac{N_j}{N_j + 1}\|\mathbf{y}-\mathbf{m}_j\|^2,$$

如果 $\Delta J < 0$,

$\Rightarrow$ 把 $\mathbf{y}$ 从 Γ_k 移入 Γ_j 会减小 J。

8.2.1 C-均值算法（K-均值/means）

- **C-均值算法步骤**

1. 选择一个初始划分，并计算各类均值;
2. 选择一个样本 $\mathbf{y}$，设 $\mathbf{y}\in\Gamma_i$; 如 $N_i = 1$, 则重选 $\mathbf{y}$; 否则继续;
3. 分别计算如把 $\mathbf{y}$ 移动到其他各类中造成 ΔJ;
4. 如果所有的 ΔJ 都大于0，则不移动 $\mathbf{y}$。否则移动 $\mathbf{y}$ 到产生最小 ΔJ 的类;
5. 更新相关类的均值，以及 J 值;
6. 如连续迭代 N 次 J 值不变，则停止; 否则转2。

8.2.1 C-均值算法（K-均值/means）

- **初始代表点的选择**

1. 经验选择;
2. 随机分成 c 类，选各类重心作为代表点;
3. "密度法"选择代表点:
   - 计算每个样本的一定球形邻域内的样本数作为"密度"，选"密度"最大的样本点作为第一个代表点，在离它一定距离之外最大"密度"点作为第二个代表点，…，依此类推;
4. 用前 c 个样本点作为代表点;
5. 用 c-1 聚类求 c 个代表点: 各类中心外加离它们最远的样本点，从 1 类开始。

8.2.1 C-均值算法（K-均值/means）

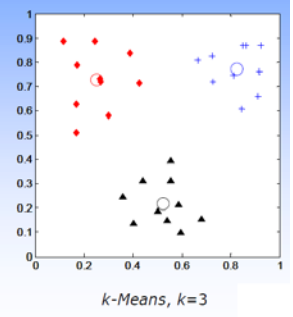
- **初始分类方法**

1. 最近距离法: 离哪个代表点近就归入哪一类;
2. 最近距离法归类，但每次都重新计算类代表点;
3. 直接划分初始分类: 第一个样本自成一类，第二个样本若离它小于某距离阈值则归入此类，否则建新类，……
4. 将特征归一化，用样本各特征之和作为初始分类依据。

8.2.1 C-均值算法（K-均值/means）

Minimize the sum of within-cluster square errors

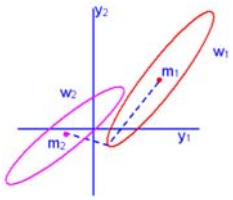
- Start with *c* cluster centers
- Iterate between
  1. Assign data points to the closest cluster centers
  2. Adjust the cluster centers to be the means of the data points
- User specified parameters: *c*, initialization of cluster centers



k-Means, k=3

8.2.1 C-均值算法（K-均值/means）

- **讨论**
  - 优点
    - 时间复杂度 O(*N*);
    - 简单易实现;
    - 适用于"球形"分布的数据;
  - 用于非监督模式识别的问题
    - 要求类别数已知;
    - 是最小方差划分，并不一定能反映内在分布;
    - 与初始划分有关，不保证全局最优。
  - 存在不少变种：初始划分的方法；更新均值的时机；聚类数目的动态决定，等等。



8.2.2 样本和核相似性度量的聚类算法

- 采用一个"核" K_j 代表一个类 Γ_j;
- 核 K_j 可以是一个函数，一个点集，某种适当的分类模型等等。
- 定义样本和各类的核之间的相似性度量 $\Delta(\mathbf{y}, K_j)$;
- 聚类准则函数，即最小化的目标函数

$$J = \sum_{j=1}^{c} \sum_{y \in \Gamma_j} \Delta(\mathbf{y}, K_j).$$

8.2.2 样本和核相似性度量的聚类算法

- **算法步骤**，类似于 C-均值:
  1. 选择初始划分，并计算初始核 K_j，j=1,…,c;
  2. 按照如下规则把各样本分类:

$$if\ \ \Delta(\mathbf{y}, K_j) = \min_{i=1,\cdots,c} \Delta(\mathbf{y}, K_i),\ \text{then}\ \ \mathbf{y} \in \Gamma_j;$$

  3. 更新核，并重复步骤 2-3 直至收敛。
- C-均值算法 = 核是类均值，样本和核之间的相似性度量是欧式距离的特例。

8.2.2 样本和核相似性度量的聚类算法

- 算法收敛的**充分条件**：准则函数 J 满足

如果 $J(\boldsymbol{\Gamma}, \tilde{\mathbf{K}}) \le J(\boldsymbol{\Gamma}, \mathbf{K})$,

那么 $J(\tilde{\boldsymbol{\Gamma}}, \tilde{\mathbf{K}}) \le J(\boldsymbol{\Gamma}, \tilde{\mathbf{K}})$;

- $\boldsymbol{\Gamma}, \mathbf{K}$： 修正之前的分类集合和对应的核集合;
- $\tilde{\boldsymbol{\Gamma}}, \tilde{\mathbf{K}}$： 修正之后的分类集合和对应的核集合;

8.2.2 样本和核相似性度量的聚类算法

- **正态核函数**：适用于各类为正态分布

$$K_j(\mathbf{y}, V_j) = \frac{1}{(2\pi)^{d/2} |\hat{\boldsymbol{\Sigma}}_j|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{y}-\mathbf{m}_j)^T \hat{\boldsymbol{\Sigma}}_j^{-1}(\mathbf{y}-\mathbf{m}_j)\right\},$$

- 参数集 $V_j = (\mathbf{m}_j, \hat{\boldsymbol{\Sigma}}_j)$ 从各类样本中估计;
- 相似性度量:

$$\Delta(\mathbf{y}, K_j) = \frac{1}{2}(\mathbf{y}-\mathbf{m}_j)^T \hat{\boldsymbol{\Sigma}}_j^{-1}(\mathbf{y}-\mathbf{m}_j) + \frac{1}{2}\log|\hat{\boldsymbol{\Sigma}}_j|.$$

8.2.2 样本和核相似性度量的聚类算法

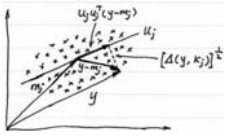
- **主轴核函数**：适用于各类样本集中分布在各自的主轴方向上的子空间里的情况

$$K(\mathbf{y},V_j)=\mathbf{U}_j^T\mathbf{y},$$

其中 $\mathbf{U}_j=(\mathbf{u}_1,\mathbf{u}_2,\cdots,\mathbf{u}_{d_j})$ 是第 j 类样本协方差阵的前 d_j 个最大特征值对应的特征向量系统；

$$\Delta(\mathbf{y},K_j)=[(\mathbf{y}-\mathbf{m}_j)-\mathbf{U}_j\mathbf{U}_j^T(\mathbf{y}-\mathbf{m}_j)]^T\cdot[(\mathbf{y}-\mathbf{m}_j)-\mathbf{U}_j\mathbf{U}_j^T(\mathbf{y}-\mathbf{m}_j)]$$

是样本到主轴子空间的距离。



8.2.3 近邻函数准则函数

- **近邻函数**：不同样本间相似性的度量
  - 如果 $\mathbf{y}_i$ 是 $\mathbf{y}_j$ 的第 I 个近邻，则 $\mathbf{y}_i$ 对 $\mathbf{y}_j$ 的近邻系数为 I;
  - 如果 $\mathbf{y}_j$ 是 $\mathbf{y}_i$ 的第 K 个近邻，则 $\mathbf{y}_j$ 对 $\mathbf{y}_i$ 的近邻系数为 K;
  - $\mathbf{y}_i$ 和 $\mathbf{y}_j$ 之间的近邻函数：

$$\alpha_{ij}=I+K-2,\quad i\neq j;$$

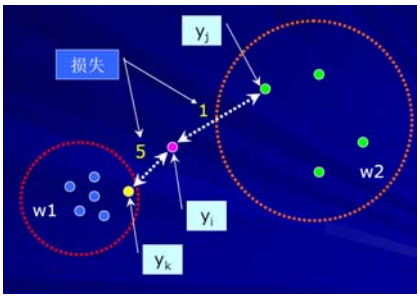
8.2.3 近邻函数准则函数

- **连接**：如 $\mathbf{y}_i$ 和 $\mathbf{y}_j$ 被分到同一类，则称它们相互连接;
- 每个连接对应一个**连接损失**：两点之间的近邻函数 α_{ij};
- 一个点和其自身的连接损失为 $\alpha_{ii}=2N$（N 是样本总数），以惩罚只有一个点的聚类;
- 不同类的点不存在连接，故连接损失 $\alpha_{ij}=0$;
- 总类内损失：

$$L_{within}=\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_{ij}.$$

8.2.3 近邻函数准则函数

- 连接损失可使得密度相近的点容易聚成一类



损失：$\alpha_{ik}=1+6-2=5,\alpha_{ij}=2+1-2=1;$

8.2.3 近邻函数准则函数

- 第 i 类和第 j 类之间的最小近邻函数值定义为：

$$\gamma_{ij}=\min_{\mathbf{y}_k\in\Gamma_i,\mathbf{y}_l\in\Gamma_j}(\alpha_{ij}),$$

- 记第 i 类内最大连接损失为 α_{imax};
- 定义第 i 类和第 j 类之间的连接损失为 β_{ij}，其**设计目标**是：如果两类间的最小近邻值小于任何一方的类内的最大连接损失时，损失代价就是正的，从而应该考虑把这两类合并。

$$b_{ij}=\begin{cases}-[(\gamma_{ij}-a_{imax})+(\gamma_{ij}-a_{jmax})], & if\ \gamma_{ij}>a_{imax},\gamma_{ij}>a_{jmax}\\ \gamma_{ij}+a_{imax} & if\ \gamma_{ij}\le a_{imax},\gamma_{ij}>a_{jmax}\\ \gamma_{ij}+a_{jmax} & if\ \gamma_{ij}>a_{imax},\gamma_{ij}\le a_{jmax}\\ \gamma_{ij}+a_{imax}+a_{jmax} & if\ \gamma_{ij}\le a_{imax},\gamma_{ij}\le a_{jmax}\end{cases}$$

8.2.3 近邻函数准则函数

- 总类间损失：

$$L_{between}=\sum_{i\neq j}\beta_{ij};$$

- 总类内损失：

$$L_{within}=\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_{ij};$$

- 准则函数：

$$J=L_{within}+L_{between}.$$

8.2.3 近邻函数准则函数

- **算法步骤**
  1. 计算距离矩阵 $\Delta_{ij} = \Delta(\mathbf{y}_i, \mathbf{y}_j)$;
  2. 用距离矩阵计算近邻矩阵 $\mathbf{M}_{ij}$，$\mathbf{M}_{ij}$ 表示 $\mathbf{y}_i$ 是 $\mathbf{y}_j$ 的第几个近邻;
  3. 计算近邻函数矩阵$\mathbf{L}_{ij} = \mathbf{M}_{ij} + \mathbf{M}_{ji} - 2\mathbf{I}$，$\mathbf{L}_{ii} = 2N$;
  4. 在 $\mathbf{L}$ 中，每个点与其最近邻连接，形成初始的划分;
  5. 对每两个类计算 γ_{ij}和 α_{imax}，α_{jmax}，只要 γ_{ij}小于 α_{imax}，α_{jmax}中的任何一个，就合并两类（建立连接）。重复至没有新的连接为止。

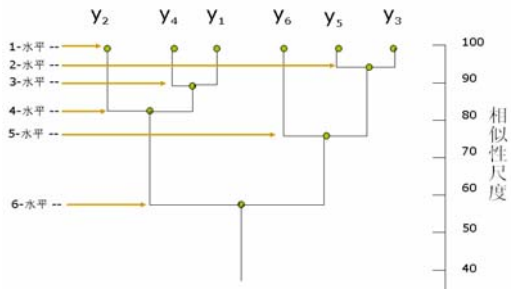
8.3 分级聚类方法

- **要解决的问题**：按相似性、或内在联系、或本质把各种事物组成有层次的结构，把最接近的划归一类，然后把相近的几个类再合并成一个类;
- **目的与用途**：将复杂的事物组织与管理起来;
- **技术问题**：如何定义相似性？
  - 可按度量值之间的差异。

动态聚类算法	分级聚类算法
迭代	非迭代

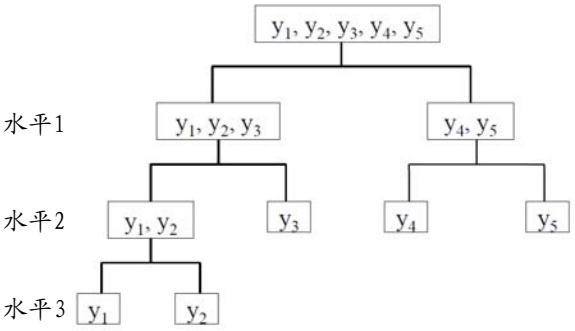
8.3 分级聚类方法

- **自底向上逐步合并**：从各类只有一个样本点开始，逐级考查类间相似度并合并，每级只合并两类，直到最后所有样本都归到一类。



8.3 分级聚类方法

- **自顶向下逐步分割**



8.3 分级聚类方法

- 两个聚类K_i与K_j之间的**相似性度量**
  - 最近距离（single-link）：

$$\Delta(K_i, K_j) = \min_{\substack{\mathbf{x}\in K_i \\ \mathbf{y}\in K_j}} \delta(\mathbf{x}, \mathbf{y}),$$

  - 最远距离（complete-link）：

$$\Delta(K_i, K_j) = \max_{\substack{\mathbf{x}\in K_i \\ \mathbf{y}\in K_j}} \delta(\mathbf{x}, \mathbf{y}),$$

  - 均值距离（average-link）：

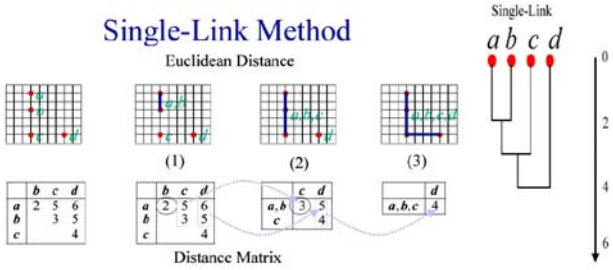
$$\Delta(K_i, K_j) = \delta(\mathbf{m}_i, \mathbf{m}_j).$$

8.3 分级聚类方法

- **算法步骤**：N 个样本自底向上逐步合并一类
  1. 初始化：每个样本自成一类（划分水平1）;
  2. K 水平划分：计算已有的$c=N-K+2$个类的类间距离矩阵$D^{(K-1)}=[d_{ij}]^{(K-1)}$，其最小元素记作$d^{(K-1)}$，相应的两个类合并成一类;
  3. 重复第2步，直到形成包含所有样本的类（划分水平N）。

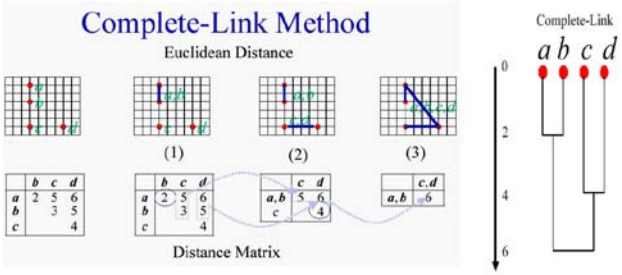
8.3 分级聚类方法

- 不同相似性度量对结果的影响



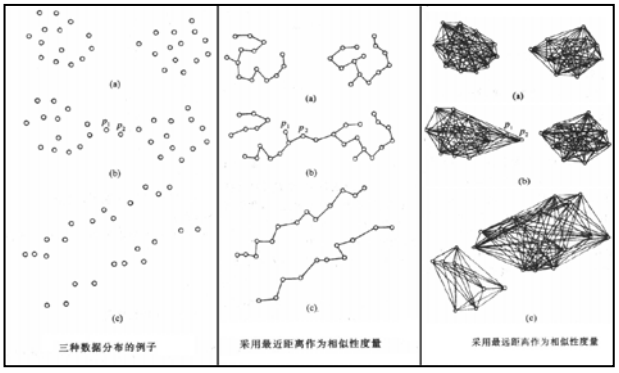
8.3 分级聚类方法

- 不同相似性度量对结果的影响



8.3 分级聚类方法

- 不同相似性度量对结果的影响



8.4 非监督学习方法中的一些问题

- 非监督模式识别问题存在更大的不确定性：可利用信息少
  - 相似性度量一般对数据尺度 (scale) 较敏感；
- 影响聚类结果的因素：样本的分布，样本数量，聚类准则，相似性度量，预分类数等；
- 针对不同数据，不同目标选择不同的聚类算法；
- 动态聚类算法计算效率高，实际应用多。