

模式识别

第九章 非监督学习方法

北京航空航天大学计算机学院

引言

- * 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)

监督与非监督学习方法比较

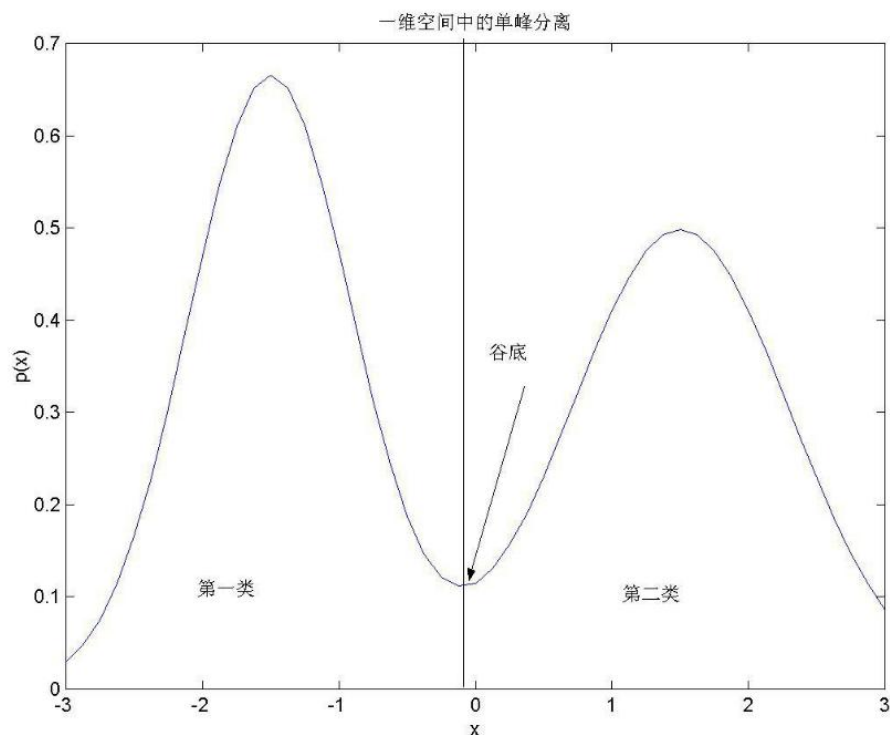
- * 监督学习方法必须要有训练集与测试样本。在训练集中找规律，而对测试样本使用这种规律；而非监督学习只有一组数据，在该组数据集内寻找规律。
- * 监督学习方法的目的是识别事物，给待识别数据加上标注（label），因此训练样本集必须由带标注的样本组成。而非监督学习方法只有要分析的数据集本身，没有标注。如果发现数据集呈现某种聚集性，则可按自然的聚集性分类，但不以与某种预先的分类标注对上号为目的。

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，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，然后依据样本与这些核心之间的相似性度量将样本聚集成不同类别。

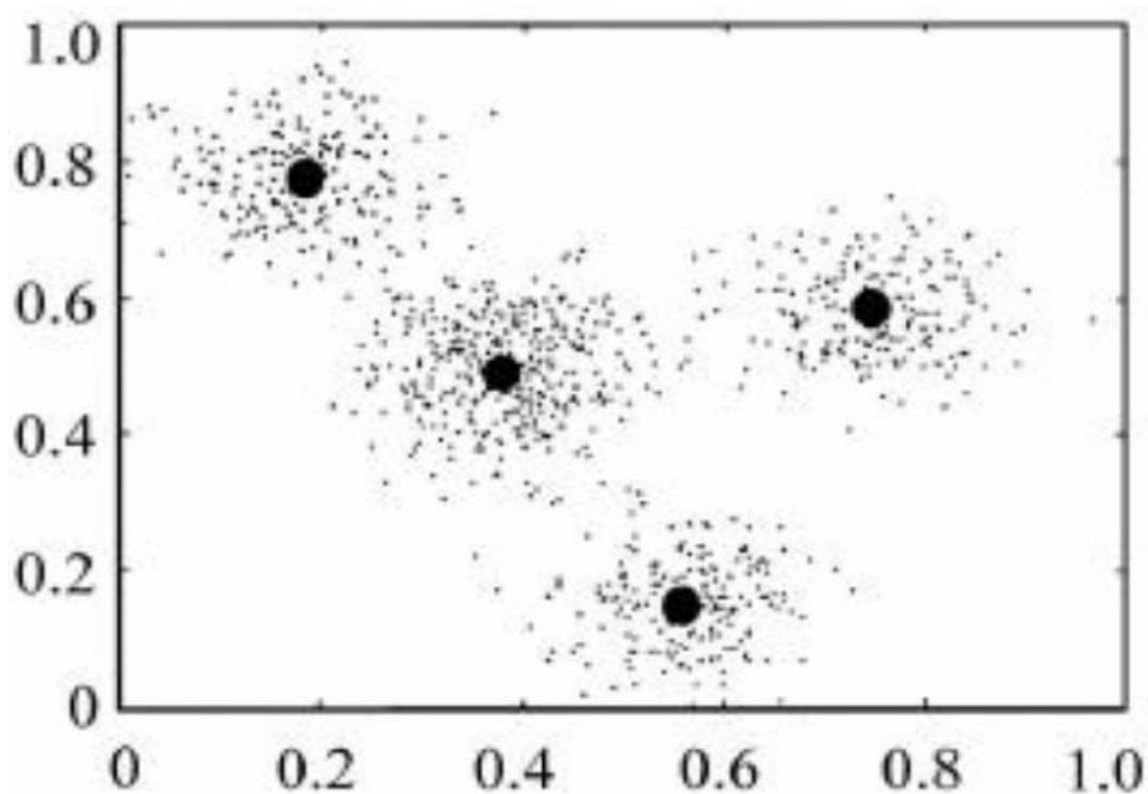
基于概率密度函数估计的直接方法

- * 划分整个空间为N个区域，使得每个区域的概率密度函数是单峰的
- * 例：玉米与杂草



基于概率密度函数估计的直接方法

* 多维分布



基于相似性度量的间接聚类方法

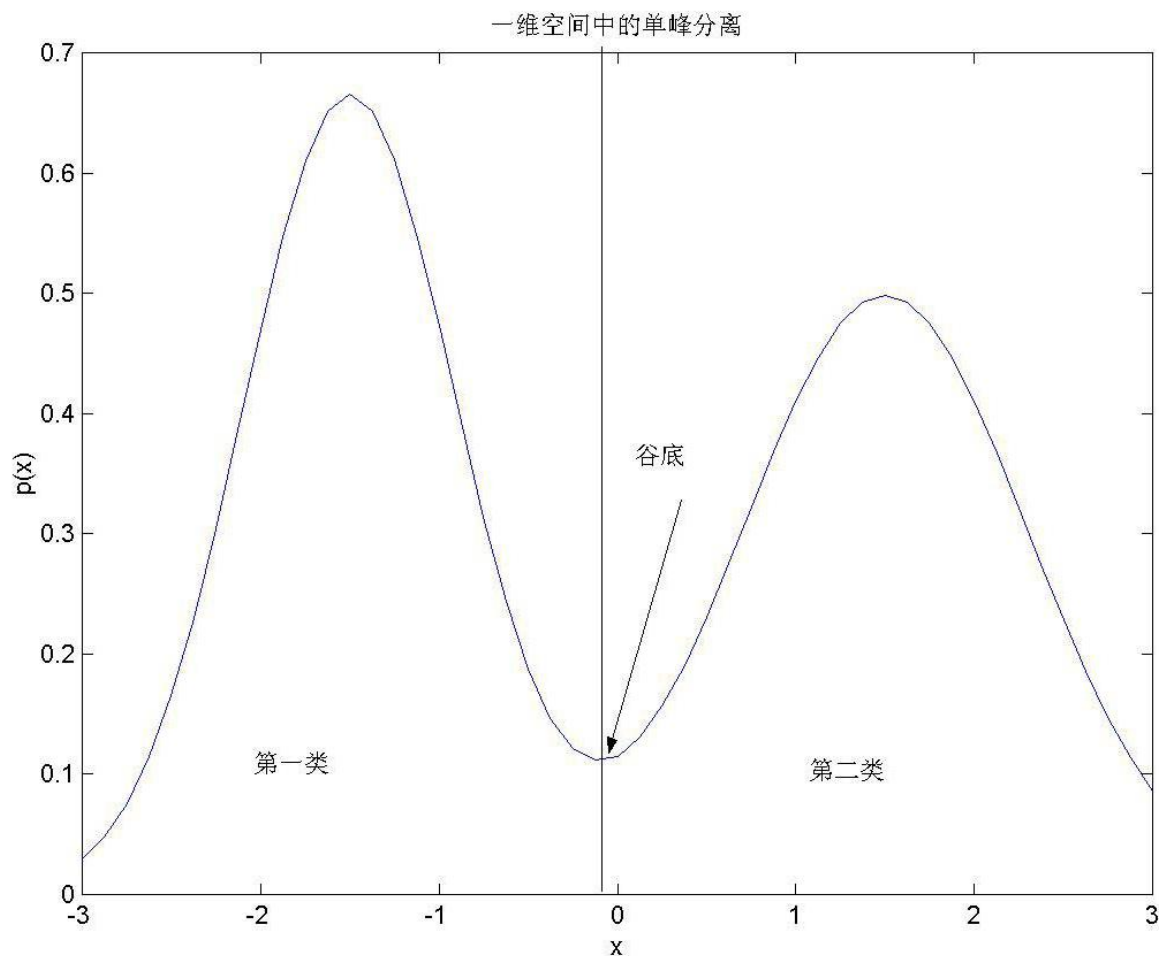
- * 根据样本间的相似性，使某种准则函数最大（小）
- * C均值方法（K均值方法），使下述准则最小：

$$J = \sum_{i=1}^C \sum_{y \in \Gamma_i} \|y - m_i\|^2$$

单峰子集的分离方法

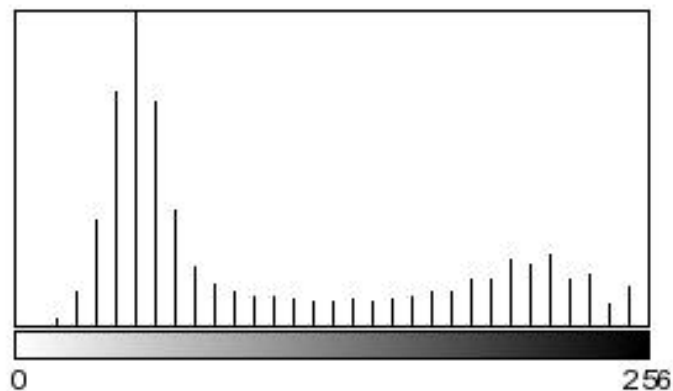
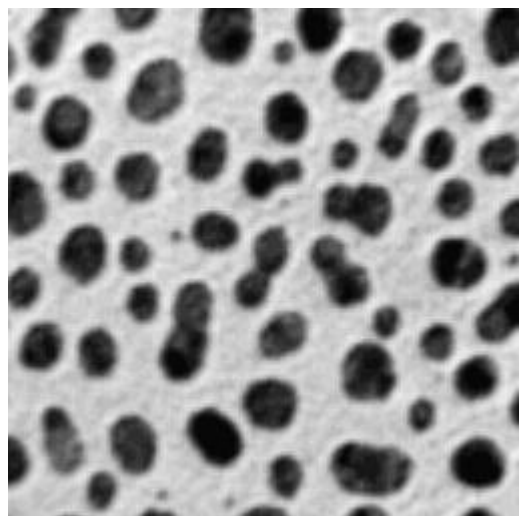
- * 思想：把特征空间分为若干个区域，在每个区域上混合概率密度函数是单峰的，每个单峰区域对应一个类别。
- * 一维空间中的单峰分离：对样本集 $K_N=\{x_i\}$ 应用直方图/Parzen窗方法估计概率密度函数，找到概率密度函数的峰以及峰之间的谷底，以谷底为阈值对数据进行分割。

一维空间中的单峰子集分离



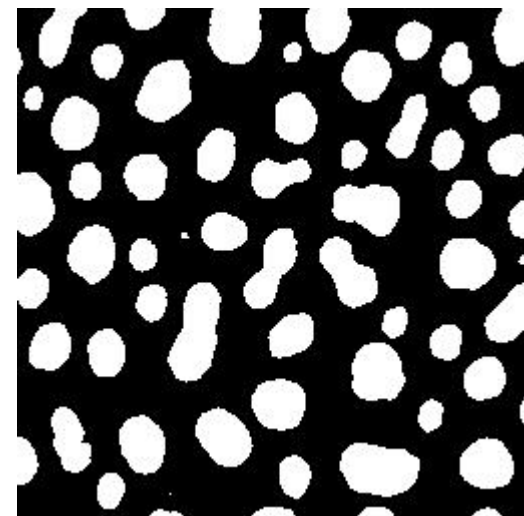
$$\operatorname{argmin}_{k=1,\dots,L} p(k)$$

灰度图像二值化算法示例



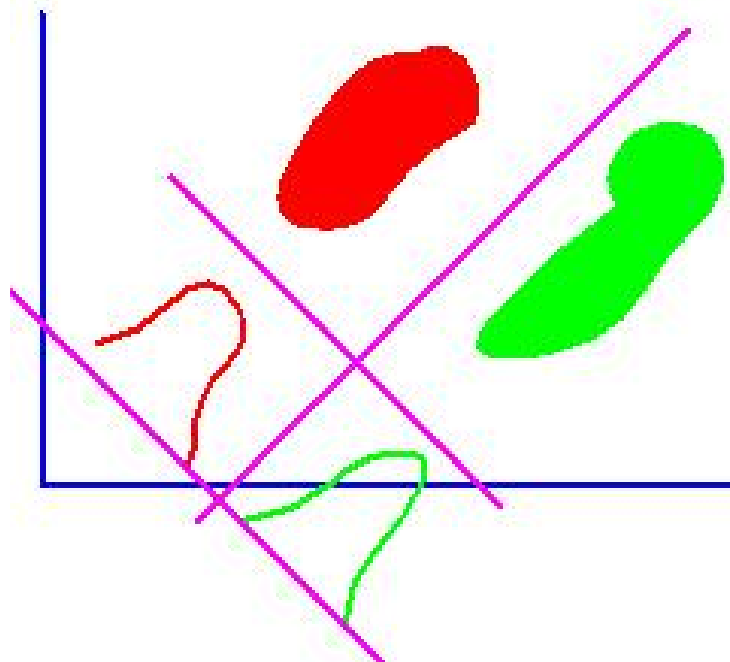
Count: 65024
Mean: 103.269
StdDev: 71.057

Min: 8
Max: 248
Mode: 48 (10396)



多维空间投影方法

- * 多维空间 y 中直接划分成单峰区域比较困难，把它投影到一维空间 x 中来简化问题。
- * 投影方法举例：

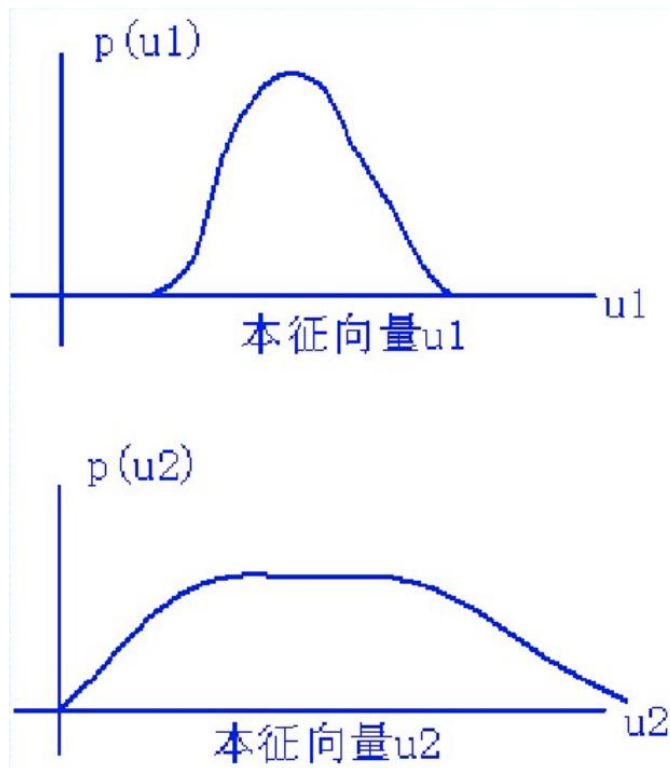
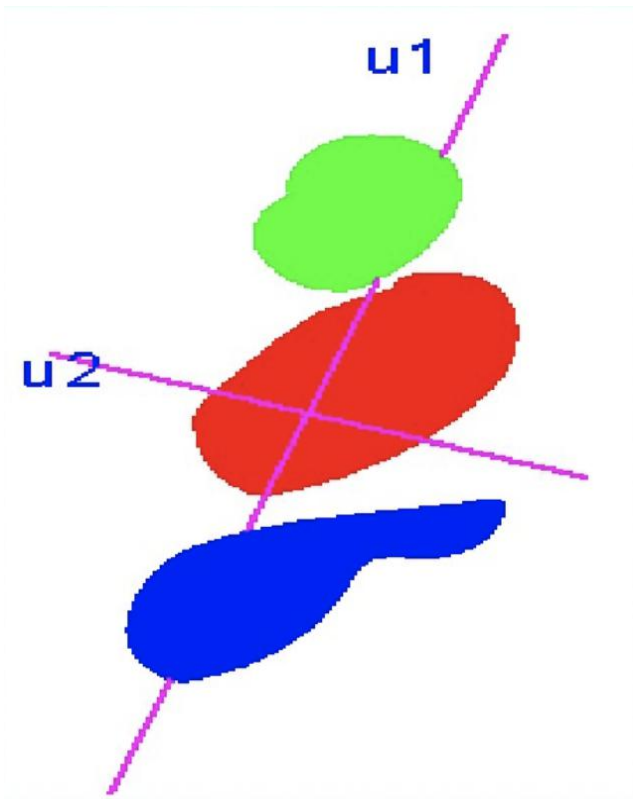


如何确定合适的投影方向 u

- * 使投影 $\{x=u^T y\}$ 的方差最大：方差越大，类之间分离的程度也可能越大
- * **样本协方差矩阵**的最大本征值对应的本征向量满足这样的要求
- * 存在问题：这样投影有时并不能产生多峰的边缘密度函数

如何确定合适的投影方向 u

* 存在问题：这样投影有时并不能产生多峰的边缘密度函数



投影方法算法步骤

- * 计算样本 y 协方差矩阵的最大本征值对应的本征向量 u ，把样本数据投影到 u 上，得到 $v=u^T y$
- * 用直方图/Parzen窗法求边缘概率密度函数 $p(v)$
- * 找到边缘概率密度函数的各个谷点，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
- * 如果没有谷点，则用下一个最大的本征值代替
- * 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

单峰子集分离的迭代算法

- * 设数据集 Y 划分为 c 个子集 Γ_i , $i = 1, 2, \dots, c$
- * 每个子集中样本数为 N_i , 总样本数为 N
- * 考查类条件概率密度的加权估计值:

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

单峰子集分离的迭代算法

* 定义指标

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

它反映了 $f(y|\Gamma_i)$ 和 $f(y|\Gamma_j)$ 之间的“距离”

* 目标：求使 J 最大的子集划分

$$f(y|\Gamma_i) = \frac{1}{N_i} \sum_{j=1}^{N_i} k(y, y_i), y \in \Gamma_i \quad (\text{Parzen窗法})$$

单峰子集分离的迭代算法

- * 考查某个样本 y_k ，若它原属于 Γ_j ，从 Γ_j 移入 Γ_i ，得到新的 $\widetilde{\Gamma}_j$ 和 $\widetilde{\Gamma}_i$ ，则显然

$$f(y|\widetilde{\Gamma}_i) \geq f(y|\Gamma_i)$$

$$f(y|\widetilde{\Gamma}_j) \leq f(y|\Gamma_j)$$

记 $f(y|\widetilde{\Gamma}_i) = f(y|\Gamma_i) + \Delta f_i,$

则 $\Delta f_i = -\Delta f_j = \frac{1}{N} k(y, y_k)$

单峰子集分离的迭代算法

* 把 y_k 从 Γ_j 移入 Γ_i 引起的指标变化量:

$$\begin{aligned}\Delta J &= \int \{ [f(y|\tilde{\Gamma}_i) - f(y|\tilde{\Gamma}_j)]^2 - [f(y|\Gamma_i) - f(y|\Gamma_j)]^2 \\ &\quad + \sum_{k=1, k \neq i, j}^c [(f(y|\Gamma_k) - f(y|\tilde{\Gamma}_j))^2 - (f(y|\Gamma_k) - f(y|\Gamma_j))^2 \\ &\quad + (f(y|\Gamma_k) - f(y|\tilde{\Gamma}_i))^2 - (f(y|\Gamma_k) - f(y|\Gamma_i))^2] \} p(y) dy \\ &= \int [2c\Delta f_i]^2 p(y) dy + 2c \int [f(y|\Gamma_i) - f(y|\Gamma_j)] \Delta f_i p(y) dy\end{aligned}$$

第一项恒大于0，第二项差越大， ΔJ 越大

单峰子集分离的迭代算法

- * 通过把 y_k 从 Γ_j 移入 Γ_i ，使 J 增大，故应选择使 ΔJ 尽可能大的 Γ_i 移入，即选择 $f(y_k|\Gamma_i) = \max_l f(y_k|\Gamma_l)$

以使 $|f(y|\Gamma_i) - f(y|\Gamma_j)|$ 最大，从而使 ΔJ 最大。

- * 若存在两个（或以上）子集的 $f(y_k|\Gamma_i)$ 最大（相等），则可移入其中任一类。

单峰子集分离的迭代算法

* 算法步骤:

- (1) 初始划分 Υ
- (2) 对每个样本 $y_k, k = 1, \dots, N$, 逐一计算 $f(y_k|\Gamma_i)$, 并归入使 $f(y_k|\Gamma_i)$ 最大的子集中
- (3) 重复(2), 直到不再有样本发生转移

类别分离的间接方法

- * 目标: 类内元素相似性高, 类间元素相似性低
- * 该类方法的两个要点:
 - * 相似性度量
 - * 准则函数
- * 相似性度量:

样本间相似性度量:
特征空间的某种距离度量

$$\delta(x_i, x_j) = (x_i - x_j)^T (x_i - x_j)$$

样本与样本聚类间
相似性度量

$$\delta(x_i, K_j)$$

总结

- * 不同的聚类方法实际上反映了对聚类（及数据）的不同理解：
 - * 混合模型：数据服从混合分布，聚类对应于各分布
 - * 单峰子集：聚类即概率分布中的单峰，即样本分布相对集中的区域
 - * 间接方法：相似的样本聚类，不同聚类的样本不相似

动态聚类方法

- * 距离函数：进行相似性度量
- * 准则函数：评价聚类结果的质量
- * 迭代，直到准则函数取得极值

K均值算法

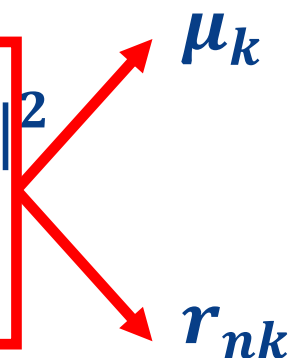
- * 给定 D 维空间上的数据集 $\{x_1, \dots, x_N\}$, 并不知道这些数据集所对应的类型和标号, 通过聚类方法将这些数据集划分成 K 类。
- * 对于 K 个聚类中的每一类 k , 分别建立一个代表点 μ_k , 将每一个样本划归到离该样本最近的 μ_k 所代表的聚类。
- * 目的: 最小化一个准则函数 J

K均值聚类

- * 对于样本 x_n ，定义一个聚类标注 r_n ，即如果样本 x_n 属于第 k 个聚类，则：

$$r_{nk} = 1, r_{nj} = 0 \quad \text{for } j \neq k$$

- * 希望每个样本与最接近它的聚类代表点之间的距离尽可能小，定义准则函数：

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|x_n - \mu_k\|^2$$


The diagram shows the objective function J enclosed in a red rectangular box. Two red arrows originate from the right side of the box: one points diagonally upwards and to the right towards the symbol μ_k , and the other points diagonally downwards and to the right towards the symbol r_{nk} .

K均值聚类

* 两步走策略

* 初始化：对聚类代表点 μ_k 进行初始化

* 迭代进行下述步骤，直至收敛：

* 第一步：根据 μ_k ，按照最优化准则计算 r_{nk}

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|x_n - \mu_k\|^2 \quad \longrightarrow \quad r_{nk} = \begin{cases} 1 & \text{if } k = \underset{j}{\operatorname{argmin}} \|x_n - \mu_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$$

* 第二步：根据 r_{nk} ，按照最优准则计算 μ_k

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|x_n - \mu_k\|^2 \quad \longrightarrow \quad 2 \sum_{n=1}^N r_{nk} (x_n - \mu_k) = 0 \quad \longrightarrow \quad \mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$

迭代 r_{nk} \longrightarrow **Expectation** 迭代 μ_k \longrightarrow **Maximization**

K均值聚类

- * 初始划分：一般可先选代表点，再进行初始分类
- * 代表点选择方法：
 - * 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
 - * 计算每个样本的一定球形领域内的样本数作为“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选最大“密度”点作为第二个代表点，…，依次类推

K均值聚类

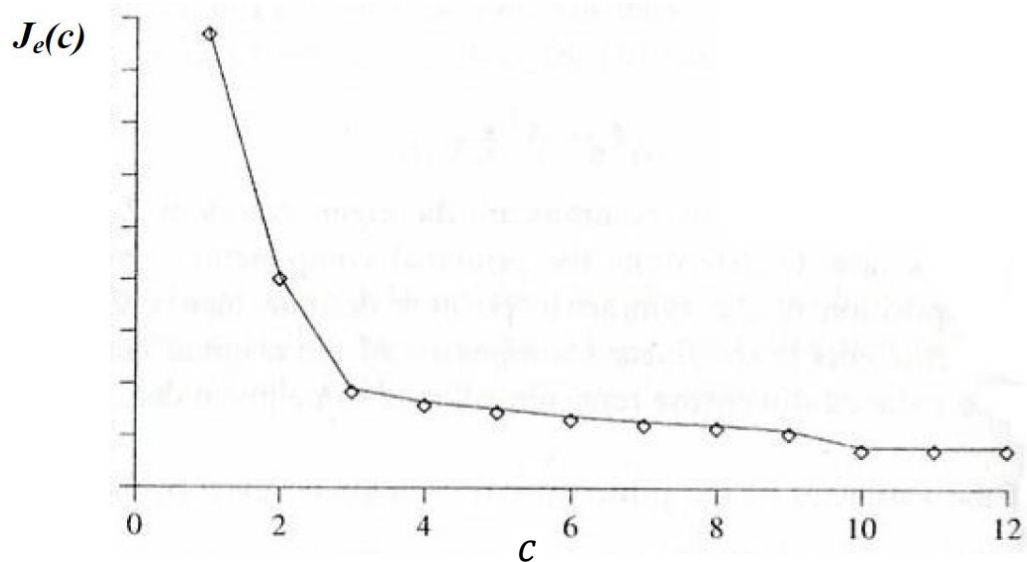
- * 4. 用前 c 个样本点作为代表点
- * 5. 用 $c - 1$ 聚类求 c 个代表点：各类中心外加离它们最远的样本点，从1类开始
- *

K均值聚类

- * K 均值聚类方法用于非监督模式识别的问题：
 - * 1. 要求类别数已知；
 - * 2. 是最小方差划分，并不一定能反映内在分布；
 - * 3. 与初始划分有关，不保证全局最优。

如何获取类别C

- * 一种实验确定方法:
- * 对 $c = 1, 2, 3, \dots$, 取类, 求 $J_e(c)$, 如图找其中的拐点 (图中 $\hat{c} = 3$)
(此方法并不总有效, 并非所有情况都能找到明显的转折点)



ISODATA方法

- * 迭代自组织数据分析算法
- * ISODATA算法功能于K均值算法相比，在以下几方面有所改进：
 - * 考虑了类别的合并与分裂，因而有自我调整类别数的能力。合并主要发上在某一类内样本个数太少的情况，或两类聚类中心之间的距离太小的情况
 - * 算法具有自我调整的能力

ISODATA方法

- * ISODATA算法与K均值算法有相似之处，即聚类中心根据样本的均值来修改。
- * 不同的是，这种算法进行的过程中聚类中心的数目不是固定不变、而是反复进行修改。聚类既有合并也有分裂，合并与分裂是在一组预先选定的参数指导下进行的。

ISODATA方法

* 算法步骤:

- * (1) 初始化, 聚类数 c , 中心 m_i , $i = 1, \dots, c$ (期望聚类数 k)
- * (2) 把所有样本分到距离最近的类中, Γ_i , $i = 1, \dots, c$
- * (3) 若某个类 Γ_j 中样本数过少 ($N_j < \theta_N$), 则去掉这一类 (合入其他类), 置 $c = c - 1$
- * (4) 重新计算均值 $m_j = \frac{1}{N_j} \sum_{y \in \Gamma_j} y$, $y = 1, \dots, c$

ISODATA方法

- * (5) 计算第 j 类样本与其中心的平均距离

$$\bar{\delta}_j = \frac{1}{N_j} \sum_{y \in \Gamma_j} \|y - m_j\|, \quad j = 1, \dots, c$$

和总平均距离 $\bar{\delta} = \frac{1}{N} \sum_{j=1}^c N_j \bar{\delta}_j$

- * (6) 若是最后一次迭代（由参数 I 确定），则程序停止；

若 $c \leq k/2$ ，则转(7)（分裂）（聚类中心小于或等于希望数的一半）；

若 $c \geq 2k$ ，或是偶数次迭代，则转(8)（合并）（聚类中心数目大于或等于希望数的两倍）。

ISODATA方法

* (7) (分裂)

7-1 对每个类, 求各维标准偏差 $\sigma_j = [\sigma_{j1}, \sigma_{j2}, \dots, \sigma_{jd}]^T$

$$\sigma_{ji} = \sqrt{\frac{1}{N_j} \sum_{y_k \in \Gamma_j} (y_{ki} - m_{ji})^2}, j = 1, \dots, c, \quad i = 1, \dots, d$$

7-2 对每个类, 求出标准偏差最大的分量 $\sigma_{j \max}, j = 1, \dots, c$

7-3 若对 $\sigma_{j \max}, j = 1, \dots, c$, 存在 $\sigma_{j \max} > \theta_s$ (标准偏差参数, 即该类样本在 $\sigma_{j \max}$ 对应方向上的标准偏差大于允许的值)

且 $\bar{\delta}_j > \bar{\delta}$ 且 $N_j > 2(\theta_N + 1)$ (类内平均距离大于总体平均距离, 并且该类中样本数很大)

或 $c \leq k/2$ (聚类数小于或等于希望数目的一半)

则 Γ_j 分裂为两类, 中心分别为 m_j^+ 和 m_j^- , 置 $c = c + 1$

$$m_j^+ = m_j + r_j, \quad m_j^- = m_j - r_j$$

其中 $r_j = k\sigma_{j \max}, 0 < k \leq 1$

ISODATA方法

* (8) (合并)

8-1 计算各类中心之间的距离

$$\delta_{ij} = \|m_i - m_j\|, i, j = 1, \dots, c, i \neq j$$

8-2 比较 δ_{ij} 与 θ_c (合并参数), 对小于 θ_c 者排序:

$$\delta_{i_1j_1} < \delta_{i_2j_2} < \dots < \delta_{i_lj_l}$$

8-3 把 m_{i_l} 和 m_{j_l} 合并:

$$m_l = \frac{1}{N_{i_l} + N_{j_l}} [N_{i_l} m_{i_l} + N_{j_k} m_{j_k}]$$

并置 $c = c - 1$ 。每次迭代中避免同一类被合并两次。

ISODATA方法

- * (8) 若是最后一次迭代，则终止。
否则转(2)。（必要时可调整算法参数）

基于样本与聚类间相似性度量的动态聚类算法

- * C均值方法的缺点：用均值代表类，适用于近似球状分布的类
 - * 改进：
 - * 用核 $K_j = k(y, V_j)$ 来代表一个类 Γ_j 。 V_j 是参数集。核 k_j 可以是一个函数、一个点集或某种分类模型
 - * 定义样本 y 到类 Γ_j （核 k_j ）之间的相似性度量 $\Delta(y, k_j)$
- 准则函数
$$J_k = \sum_{j=1}^c \sum_{y \in \Gamma_j} \Delta(y, k_j)$$

基于样本与聚类间相似性度量的动态聚类算法

- * (1) 初始划分, 得到初始核 $k_j, j = 1, \dots, c$

- * (2) 按以下规则把各样本分类:

$$\text{若 } \Delta(y, k_j) = \min_{h=1, \dots, c} \Delta(y, k_h)$$

$$\text{则 } y \in \Gamma_j$$

- * (3) 更新 $k_j, j = 1, \dots, c$, 若 k_j 不变, 则终止; 否则转 (2)

C均值可看作 k_j 为 m_j , Δ 为欧氏距离下的特例

核函数示例

* 1. 正态核函数:

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

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

* 2. 主轴核函数:

用K-L变换得到样本子集的主轴方向作为核:

$$k(\mathbf{y}, V_j) = U_j^T \mathbf{y} \quad (V_j \text{ 表示参数集})$$

$U_j^T = [u_1, u_2, \dots, u_{d_j}]$ 是样本协方差矩阵 $\hat{\Sigma}_j$ 的 d_j 个最大本征值的本征向量系统, 则样本到核的相似性度量为:

$$\Delta(\mathbf{y}, k_j) = [(\mathbf{y} - \mathbf{m}_j) - U_j V_j^T (\mathbf{y} - \mathbf{m}_j)]^T [(\mathbf{y} - \mathbf{m}_j) - U_j V_j^T (\mathbf{y} - \mathbf{m}_j)]$$

分级聚类方法

- * 在人类认识客观世界过程中，将事物分级分类是一种很有效的手段。
- * 最典型的例子就是生物学上对物种的分类，将所有生物按照界、门、纲、目、科、属、种等级别进行分类。越相似的物种就在越低的层次上被归为一类；最相似的物种被分在同一个“种”，相似的种又被分在同一个“属”，以此类推
- * 这种思想也可以自然运用到聚类分析中，即分级聚类(hierarchical clustering)方法

分级聚类方法

- * 聚类分析是把 N 个没有类别标签的样本分成一些合理的类，在极端的情况下，最多可以分成 N 类，即每个样本自成一类；最少可以只有一个类，即全部样本都归为一类。
- * 可以从 N 类到1类逐级地进行类别划分，求得一系列类别数从多到少的划分方案，然后根据一定的指标选择中间某个适当的划分方案作为聚类的结果，这就是分级聚类的基本思想

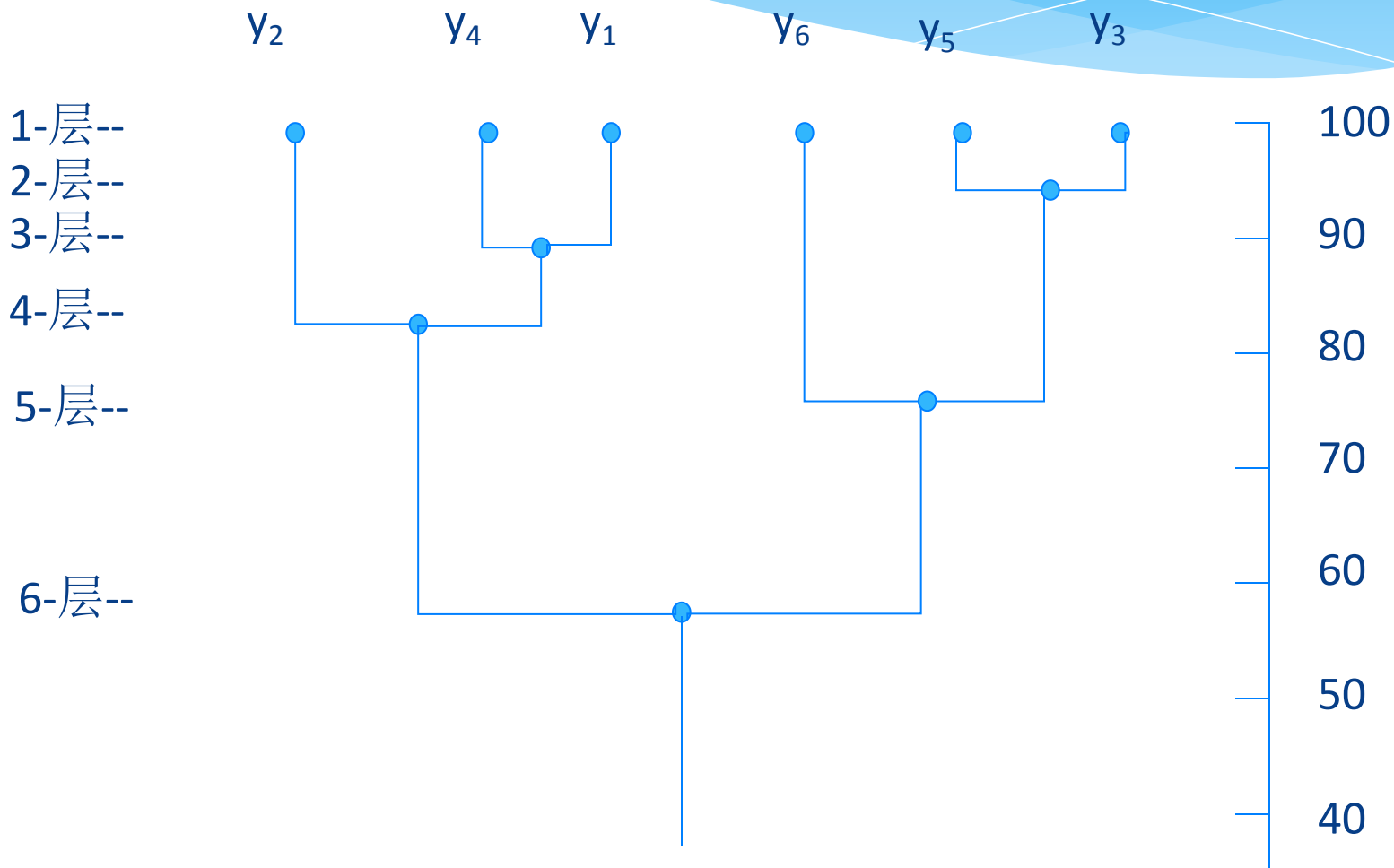
分级聚类方法

- * 聚类划分序列：N个样本自底向上逐步合并成一类，算法步骤为
 - * (1) 初始化，每个样本自成一类（划分水平1）
 - * (2) K水平划分的进行（合并）：计算已有的 $c=N-K+1$ 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，直至形成包含所有样本的类（划分水平N）

分级聚类方法

- * 划分处于 K 水平时，类数 $c=N-K+1$ ，类间距离矩阵 $\mathbf{D}^{(K)}=[d_{ij}]^{(K)}$ ，其最小元素记作 $d^{(K)}$
- * 如果 $d^{(K)} > \text{阈值} d^T$ ，则说明此水平上的聚类是适宜的

分级聚类树表示方法



两聚类间的距离度量

* 聚类 K_i 与 K_j 间的距离度量

* 最近距离: $\Delta(K_i, K_j) = \min_{x \in K_i, y \in K_j} \delta(x, y)$

* 最远距离: $\Delta(K_i, K_j) = \max_{x \in K_i, y \in K_j} \delta(x, y)$

* 均值距离: $\Delta(K_i, K_j) = \delta(m_i, m_j)$

非监督学习的一些问题

- * 非监督学习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

非监督学习的一些问题

- * 一些解决办法:

- * 先验知识

- * 多次试算

- * 改进算法（如自组织映射SOM，模糊C均值方法Fuzzy C-means）

- * 实现确定类别数

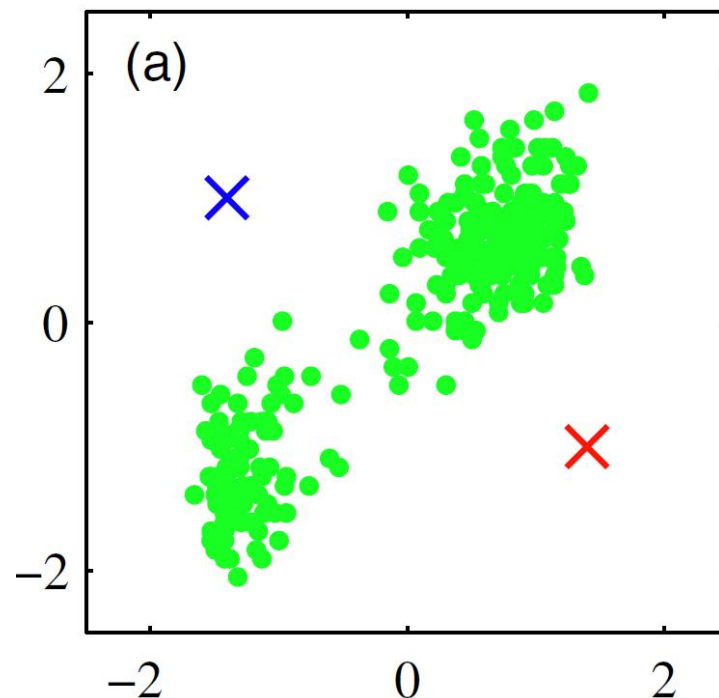
- * 对相似性度量的依赖性

具体实现及应用

- * K均值聚类算法具体实现
- * K均值聚类算法的应用
 - * 基于K均值聚类的图像分割

K均值过程示例

- * 以 $K=2$ 为例，说明K均值的实现过程
- * 初始化：对聚类代表点 μ_k 进行初始化
- * 图中蓝色及红色的x：初始化的两个聚类代表点 μ_1 及 μ_2



K均值过程示例

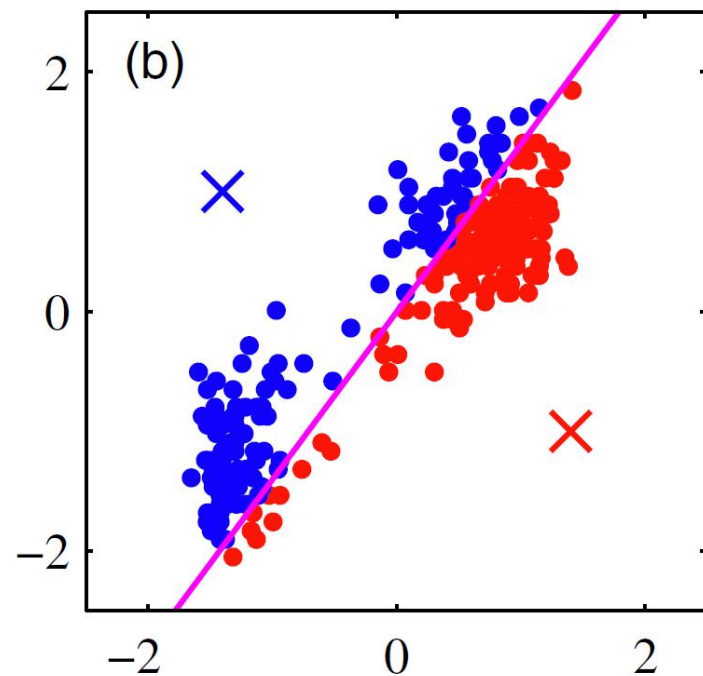
* 迭代进行下述步骤，直至收敛：

* 第一步：根据 μ_k ，按照最优化准则计算 r_{nk}

$$r_{nk} = \begin{cases} 1 & \text{if } k = \underset{j}{\operatorname{argmin}} \|x_n - \mu_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$$

* 对每个蓝色圆点代表的样本 n ，有 $r_{n1} = 1$ 且 $r_{n2} = 0$

* 对每个红色圆点代表的样本 n ，有 $r_{n1} = 0$ 且 $r_{n2} = 1$



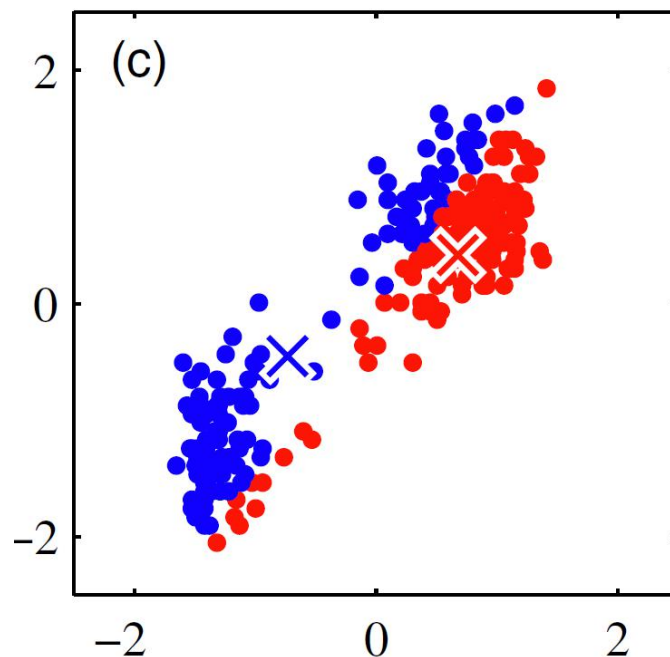
K均值过程示例

* 迭代进行下述步骤，直至收敛：

* 第二步：根据 r_{nk} ，按照最优准则计算 μ_k

$$\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$

* 图中蓝色及红色x：
更新后的两个类别的
代表点 μ_1 及 μ_2

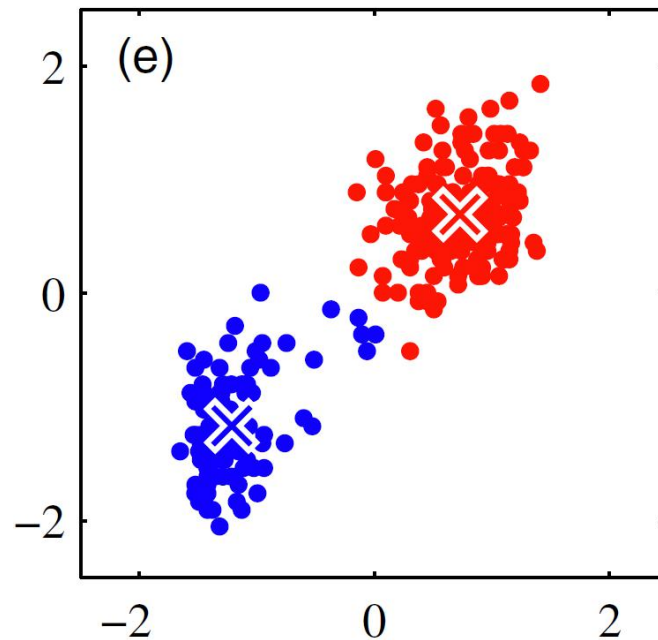
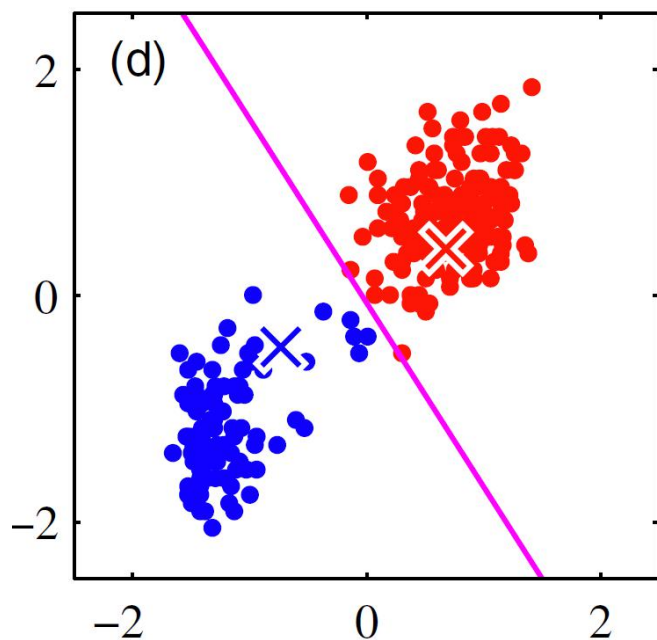


K均值过程示例

- * 迭代进行两步走策略

- * 第一步：根据 μ_k ，按照最优化准则计算 r_{nk}

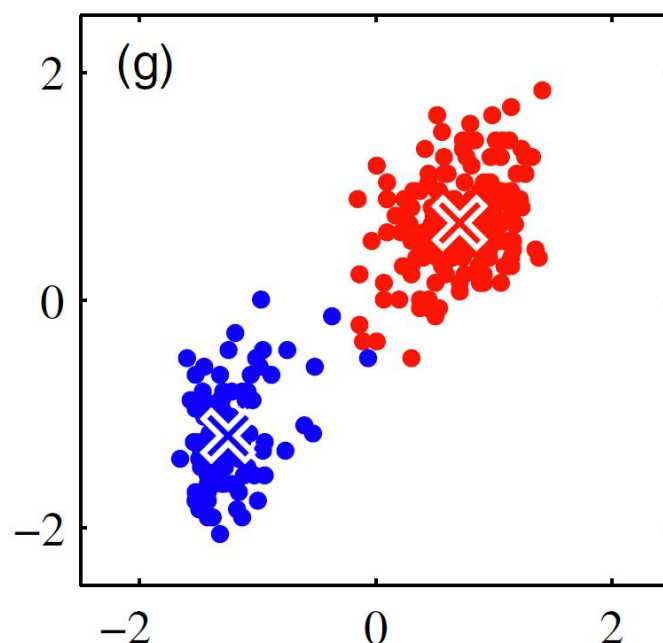
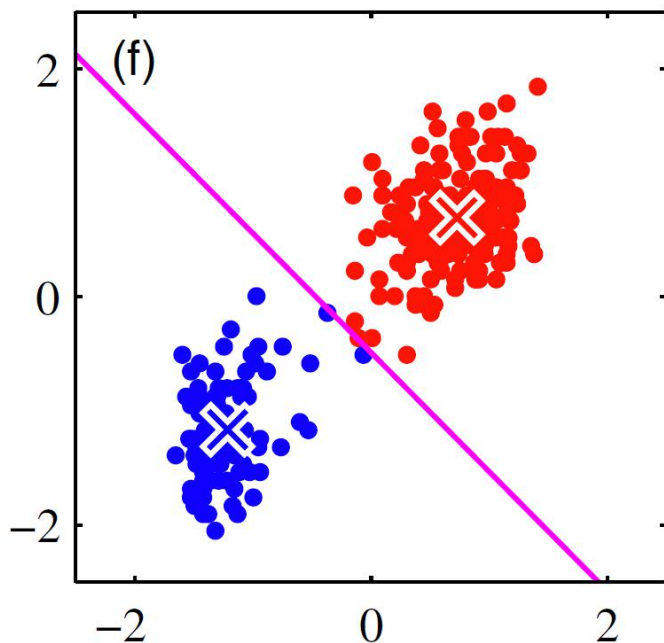
- * 第二步：根据 r_{nk} ，按照最优准则计算 μ_k



K均值过程示例

* 迭代进行两步走策略

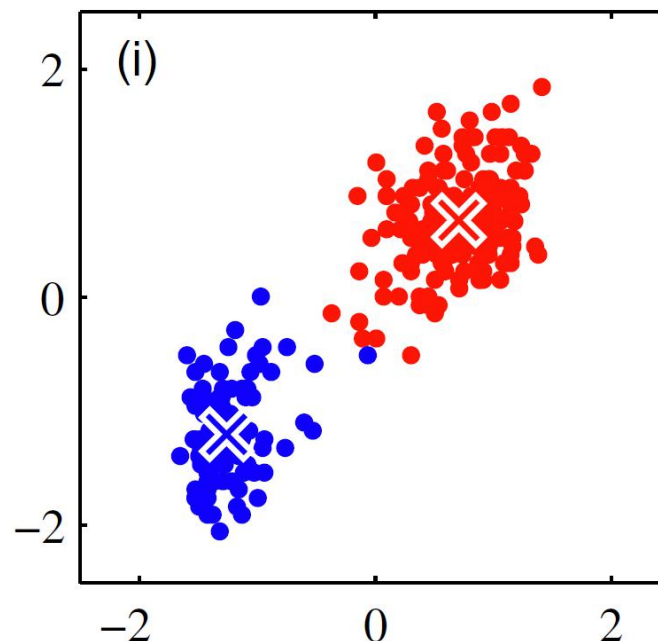
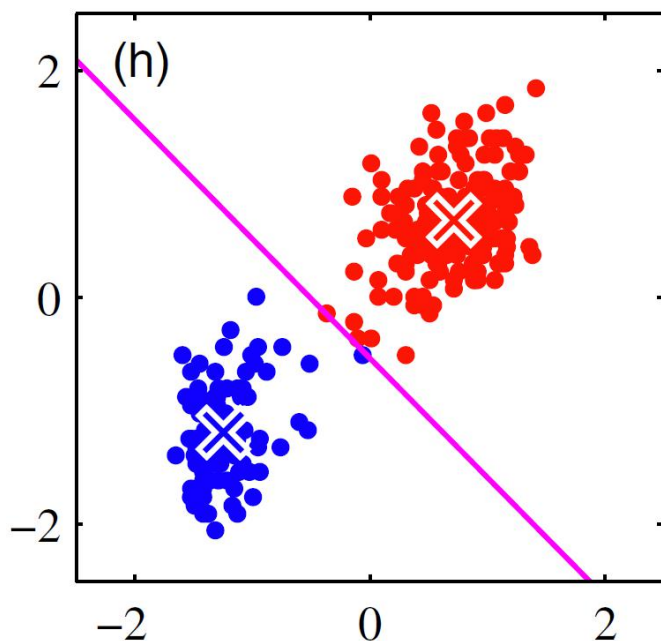
- * 第一步：根据 μ_k ，按照最优化准则计算 r_{nk}
- * 第二步：根据 r_{nk} ，按照最优准则计算 μ_k



K均值过程示例

* 迭代进行两步走策略

- * 第一步：根据 μ_k ，按照最优化准则计算 r_{nk}
- * 第二步：根据 r_{nk} ，按照最优准则计算 μ_k

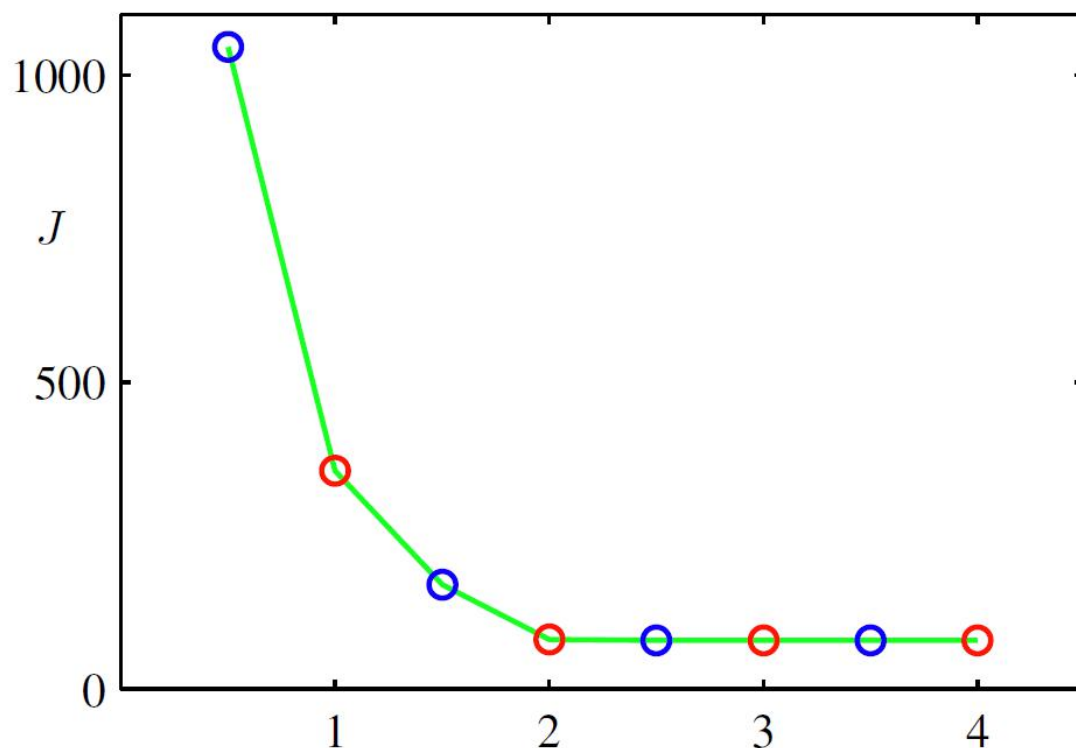


K均值过程示例

- * 迭代何时停止?
 - * 判断聚类代表点 μ_k 是否改变
 - * 判断准则函数 J 是否收敛
 - * 迭代超过一定的次数
 - * ...

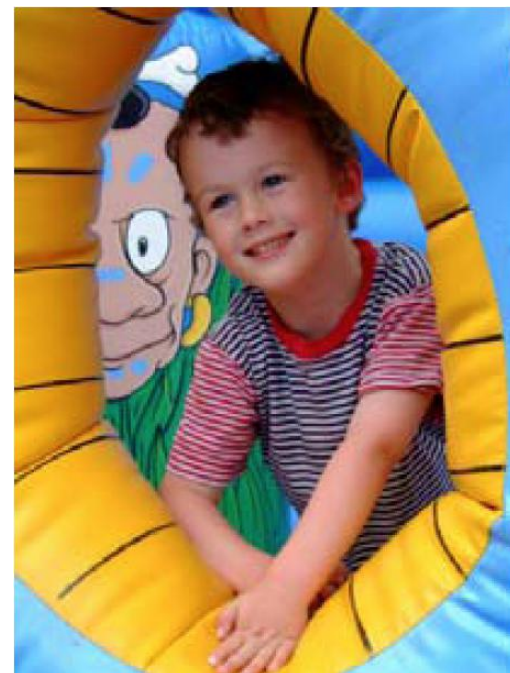
K均值过程示例

* 准则函数 J 的变化曲线



应用实例：基于K均值聚类的图像分割

- * 数字图像
 - * M 行 N 列构成的一个像素矩阵 ($M \times N$)
- * 像素
 - * R, G, B
- * 数字图像就是一个三维矩阵 ($M \times N \times 3$)
- * 图像分割就是把图像分成若干个特定的、具有独特性质的区域的过程，而分割的目的是将图像分解不同部分，这些分解结果对于特定应用具有意义
- * 我们可以将每个像素视为是一个样本点，利用K均值聚类算法，得到每个像素点对应的聚类标注，那么就可以得到图像的分割结果



应用实例：基于K均值聚类的图像分割

- * K=2时图像的分割结果（蓝色及黄色分别对应两个聚类，显示了所有对应像素的均值）



原始图像



图像分割结果 (K=2)

应用实例：基于K均值聚类的图像分割

* K=3时图像的分割结果



原始图像



图像分割结果 (K=3)

应用实例：基于K均值聚类的图像分割

* K=10时图像的分割结果



原始图像



图像分割结果 (K=10)

应用实例：基于K均值聚类的图像分割

- 可以通过图像分割，对图像进行压缩
- 当图像分割的目标是图像压缩时，在聚类时，需要对压缩率和压缩后的图像质量进行权衡。聚类数 K 值越大，压缩率越低，压缩后的图像质量越高，越接近原始图像



K=2



K=3



K=10



原始图像

谢谢