模式识别
第九章非监督学习方法

郭园方
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

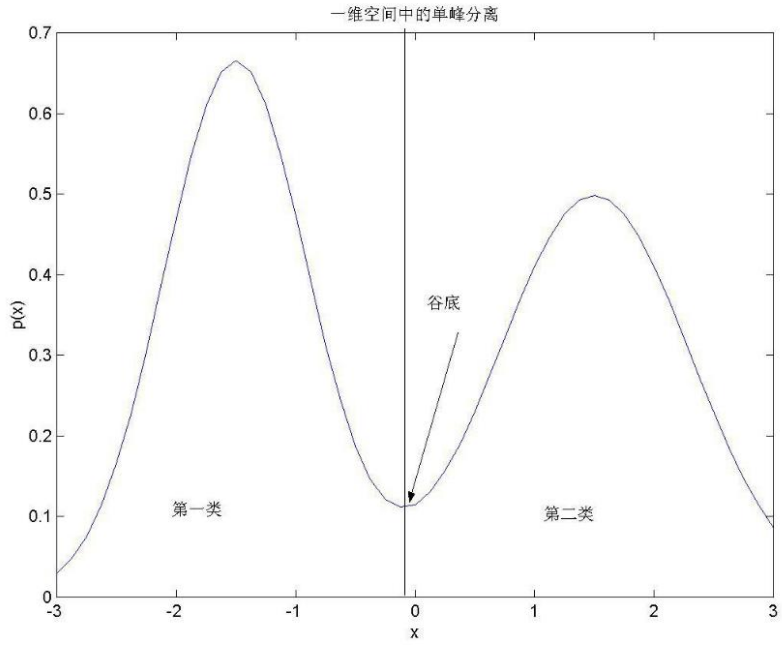
* 监督学习方法必须要有训练集与测试样本。在训练集中找规律，而对测试样本使用这种规律；而非监督学习只有一组数据，在该组数据集内寻找规律。
* 监督学习方法的目的是识别事物，给待识别数据加上标注(label)。因此训练样本集必须由带标注的样本组成。而非监督学习方法只有要分析的数据集本身，没有标注。如果发现数据集呈现某种聚集性，则可按自然的聚集性分类，但不以与某种预先的分类标注对上号为目的。

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。

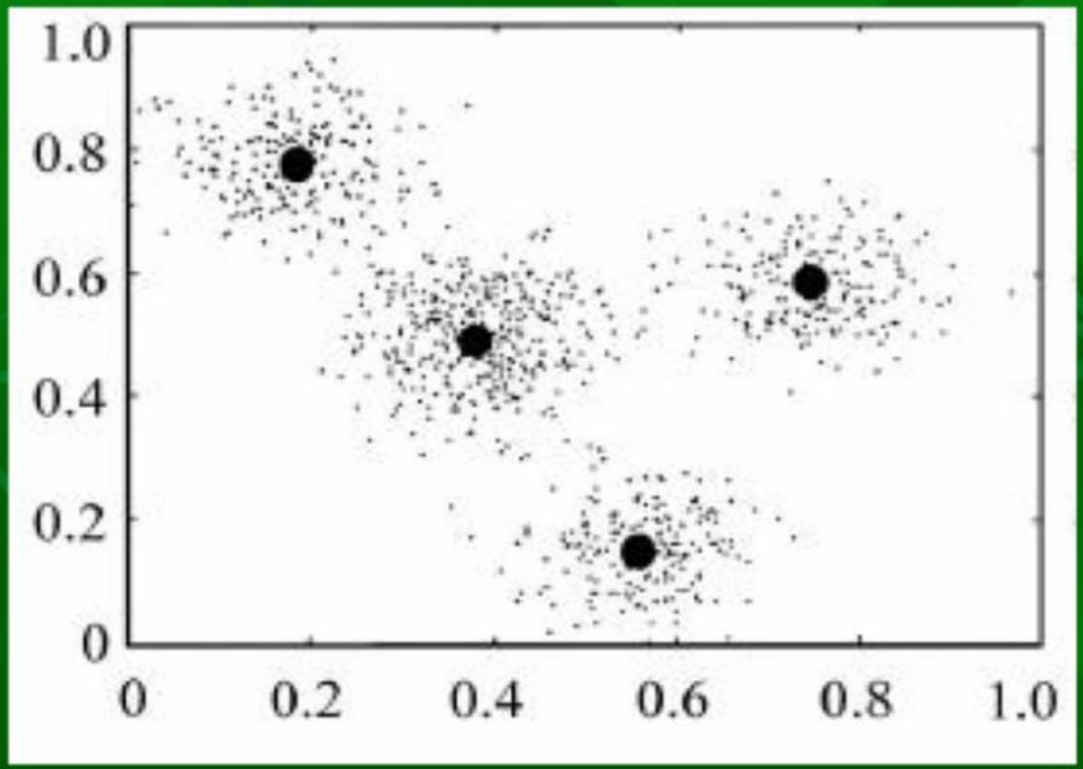
基于概率密度函数估计的方法

* 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方法

多维分布



基于相似性度量的聚类方法

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）

使准则

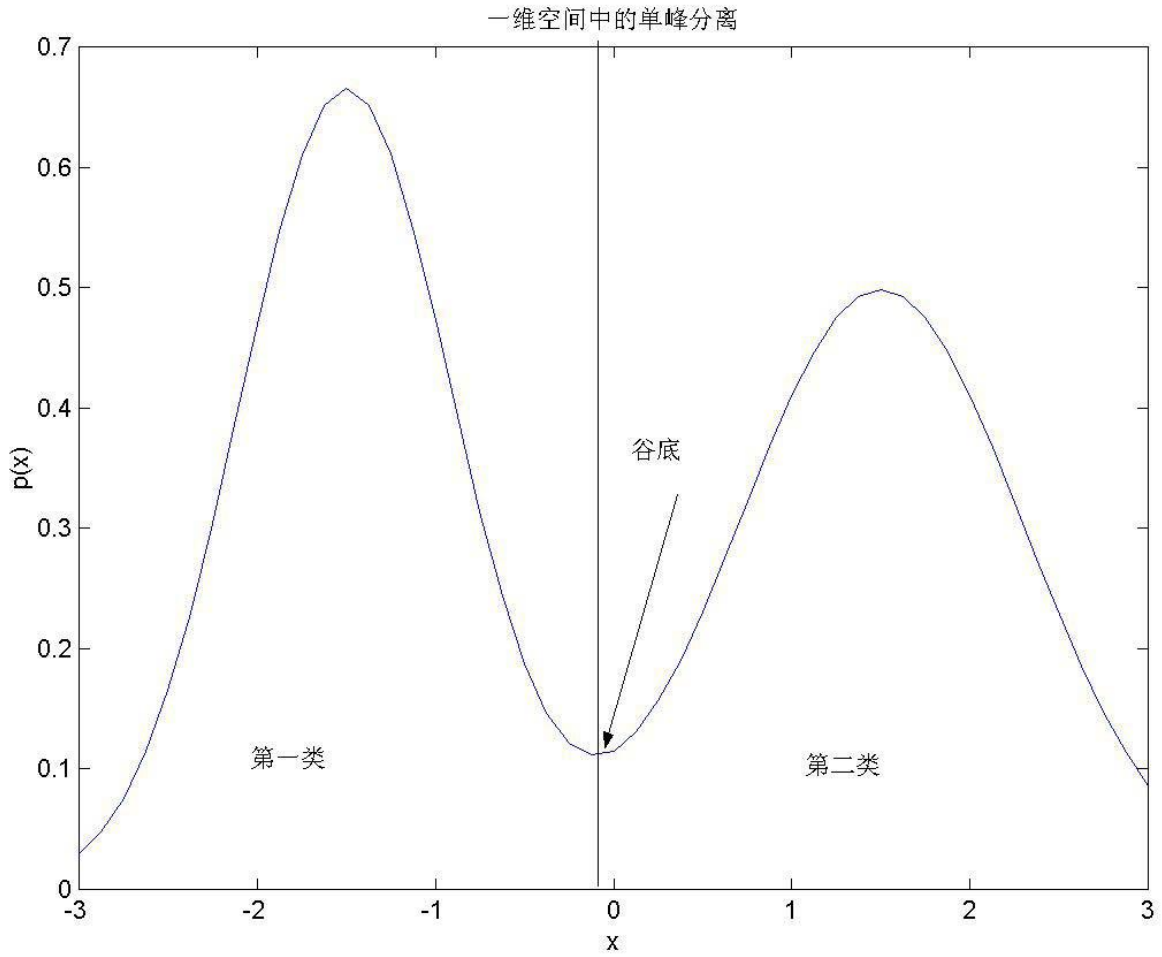
$$J(e)=\sum_{i=1}^{c}\sum_{y\in\Gamma_i}\left\|y-m_i\right\|^2$$ 最小

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。

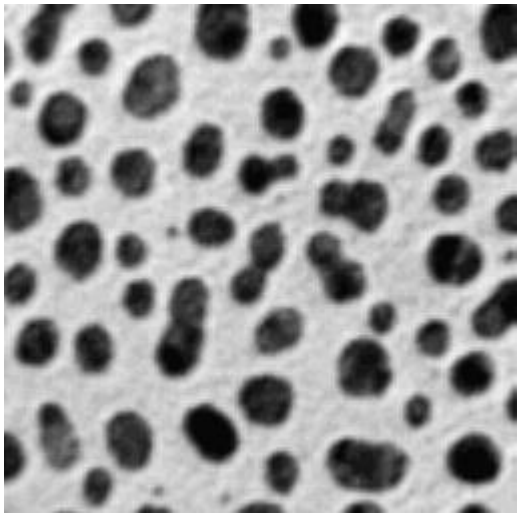
* 一维空间中的单峰分离：对样本集$K_N=\{x_i\}$应用直方图/Parzen窗方法估计概率密度函数，找到概率密度函数的峰以及峰之间的谷底，以谷底为阈值对数据进行分割。

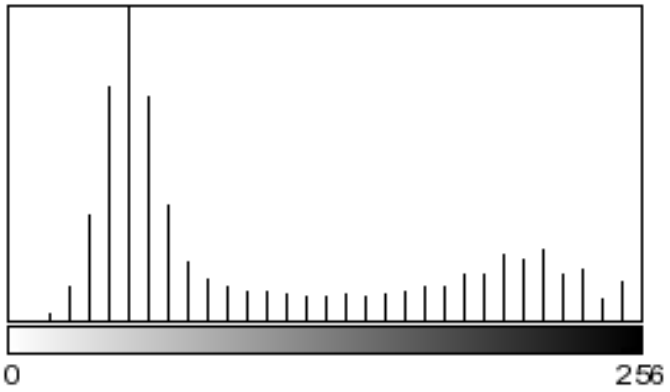
一维空间中的单峰子集分离



$$t = \underset{k=1}{\overset{L}{\operatorname{argmin}}}\, p(k)$$

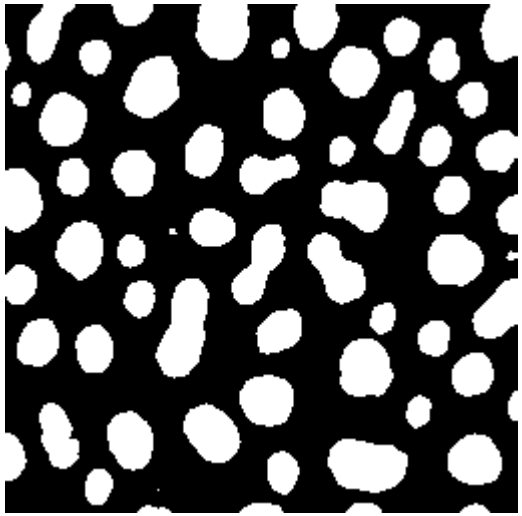
灰度图像二值化算法示例





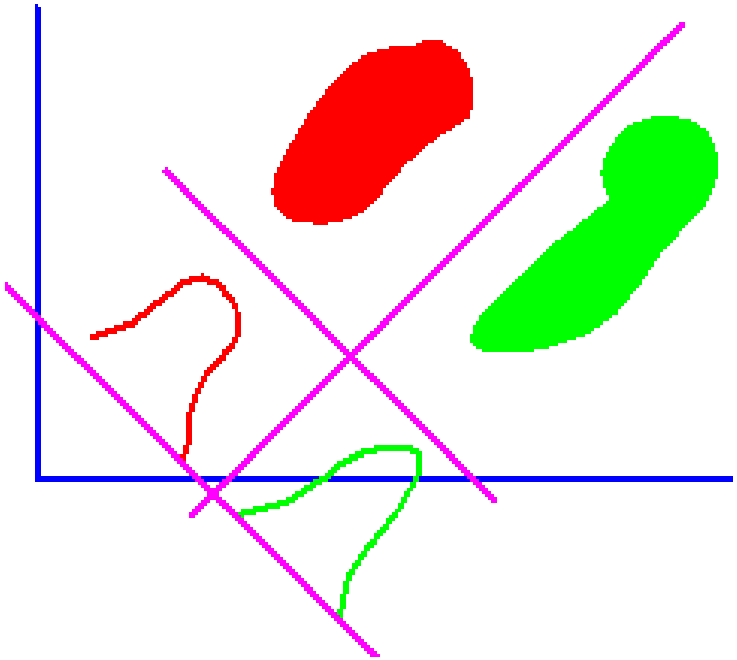
Count: 65024
Mean: 103.269
StdDev: 71.057

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{$x=u^Ty$}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

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$v=u^Ty$
* 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 Υ 划分为 c 个子集 Γ_i，$i = 1,2,\cdots,c$

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：

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

单峰子集分离的迭代算法

定义指标

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

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

目标：求使 J 最大的子集划分

$$p(y\mid\Gamma_i)=\frac{1}{N_i}\sum_{j=1}^{N_i}k(y,y_i),\ y_j\in\Gamma_i$$

（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，得新的 $\widetilde{\Gamma}_j$ 和 $\widetilde{\Gamma}_i$，

则显然

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

$$f(y \mid \widetilde{\Gamma}_j) \geq f(y \mid \Gamma_j)$$

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

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

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：

$$\Delta J = \int \Big\{ \Big[f(y \mid \tilde{\Gamma}_i) - f(y \mid \tilde{\Gamma}_{j)} \Big]^2 - \Big[f(y \mid \Gamma_i) - f(y \mid \Gamma_j) \Big]^2$$

$$+ \sum_{\substack{k=1 \\ k \neq i,j}}^{c} \Big[(f(y \mid \Gamma_k) - f(y \mid \tilde{\Gamma}_j))^2 - (f(y \mid \Gamma_k) - f(y \mid \Gamma_j))^2$$

$$+ (f(y \mid \Gamma_k) - f(y \mid \tilde{\Gamma}_i))^2 - (f(y \mid \Gamma_k) - f(y \mid \Gamma_i))^2 \Big] \Big\} p(y) dy$$

$$= \int \big[2c \Delta f_i \big]^2 p(y) dy + 2c \int \big[f(y \mid \Gamma_i) - f(y \mid \Gamma_j) \big] \Delta f_i \, p(y) dy$$

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 $$f(y_k \mid \Gamma_i) = \max_l f(y_k \mid \Gamma_l)$$

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

若存在两个（或以上）子集的 $f(y_k \mid \Gamma_i)$ 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，$k=1,\cdots,N$，逐一计算 $f(y_k \mid \Gamma_i)$，并归入使 $f(y_k \mid \Gamma_i)$ 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

$$\delta(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j)$$

样本与样本聚类间相似性度量

$$\delta(\mathbf{x}_i, K_j)$$

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$\{\mathbf{x}_1,\ldots,\mathbf{x}_N\}$，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，分别建立一个代表点$\boldsymbol{\mu}_k$，将每一个样本划归到离该样本最近的$\boldsymbol{\mu}_k$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x_n属于第k个聚类，则
* 准则函数：$r_{nk} = 1, \text{and } r_{nj} = 0 \text{ for } j \neq k$

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

$\boldsymbol{\mu}_k$

r_{nk}

K均值聚类

* 两步走策略
  * 第一步：初始化 $\boldsymbol{\mu}_k$ 按照最优化准则产生 r_{nk}

$$J = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 \Rightarrow r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$$

  * 第二步：根据产生的 r_{nk} 按照最优准则产生 $\boldsymbol{\mu}_k$

$$J = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 \Rightarrow 2\sum_{n=1}^{N} r_{nk}(\mathbf{x}_n - \boldsymbol{\mu}_k) = 0 \Rightarrow \boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\mathbf{x}_n}{\sum_n r_{nk}}$$

  * 第三步：根据产生的 $\boldsymbol{\mu}_k$，按照最优准则产生 r_{nk}

迭代 r_{nk} → Expectation 迭代 $\boldsymbol{\mu}_k$ → 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

C 均值聚类方法用于非监督模式识别的问题：

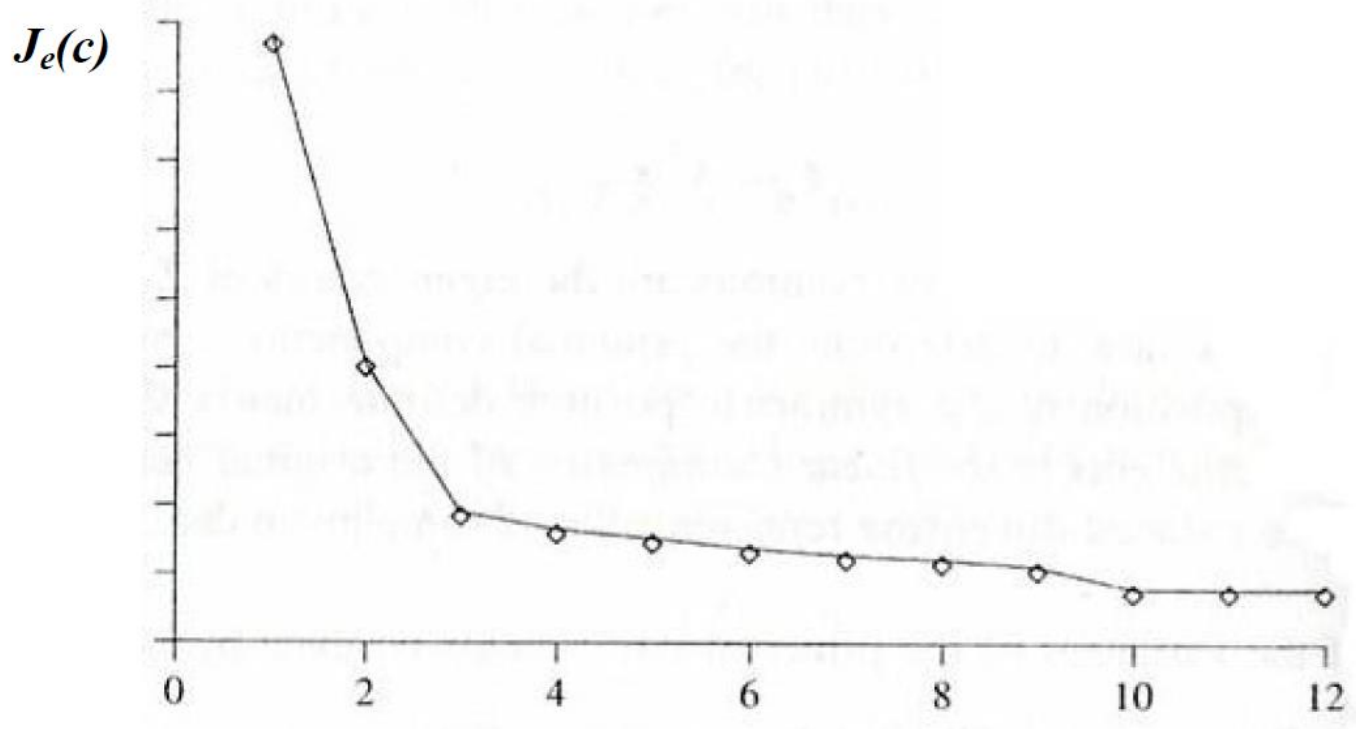
1. 要求类别数已知；
2. 是最小方差划分，并不一定能反映内在分布；
3. 与初始划分有关，不保证全局最优。

如何获取类别c

一种实验确定方法：

对 $c = 1,2,3,\cdots$，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）



ISODATA方法

算法步骤：

（1）初始化，聚类数c，中心m_i，$i=1,\cdots,c$（期望聚类数k）

（2）把所有样本分到距离最近的类中，Γ_i，$i=1,\cdots,c$

（3）若某个类Γ_j中样本数过少$\left(N_j<\theta_N\right)$，则去掉这一类（合入其它类），

置$c=c-1$

（4）重新计算均值 $$m_j = \frac{1}{N_j}\sum_{y\in\Gamma_j} y,\quad y = 1,\cdots,c$$

（5）计算第 j 类样本与其中心的平均距离

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

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 $c \le k/2$，则转（7）（分裂）；

若 $c \ge 2k$，或是偶数次迭代，则转（8）（合并）。

（7）（分裂）

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

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

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

7-3　若对 $\sigma_{j\max}$，　$j = 1, \cdots, c$，存在 $\sigma_{j\max} > \theta_s$（标准偏差参数

且　$\overline{\delta}_j > \overline{\delta}$ 且 $N_j > 2(\theta_N + 1)$

或　$c \le k/2$

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

（8）（合并）

8-1　计算各类中心之间的距离

$$\delta_{ij} = \left\| m_i - m_j \right\|, \quad i, j = 1, \cdots, c, \quad i \neq j$$

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

$$\delta_{i_1 j_1} < \delta_{i_2 j_2} < \cdots < \delta_{i_l j_l}$$

8-3　把 m_{i_l} 和 m_{j_l} 合并：

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

并置 $c = c - 1$.　每次迭代中避免同一类被合并两次。

ISODATA方法

（9）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 $$J_k = \sum_{i=1}^{c} \sum_{y \in \Gamma_j} \Delta(y, k_j)$$

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， $j=1,\cdots,c$

（2）按以下规则把各样本分类：

$$\text{若 } \Delta(y,k_j)=\min_{k=1,\cdots,c}\Delta(y,k_k)$$

$$\text{则 } y\in\Gamma_j$$

（3）更新 k_j， $j=1,\cdots,c$，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：

$$k_k(y,v_j)=\frac{1}{(2\pi)^{d/2}\left|\hat{\Sigma}_j\right|^{1/2}}\exp\left\{-\frac{1}{2}(y-m_j)^T\hat{\Sigma}_j^{-1}(y-m_j)\right\}$$

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

核函数示例

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：

$$k(y,V_j)=U_j^T y$$

$U_j^T=\left[u_1,u_2,\cdots,u_{d_j}\right]$是$\hat{\Sigma}_j$的$d_j$个最大本征值的本征向量系统。

$$\Delta(y,k_j)=\left[(y-m_j)-U_jV_j^T(y-m_j)\right]^T\left[(y-m_j)-U_jV_j^T(y-m_j)\right]$$

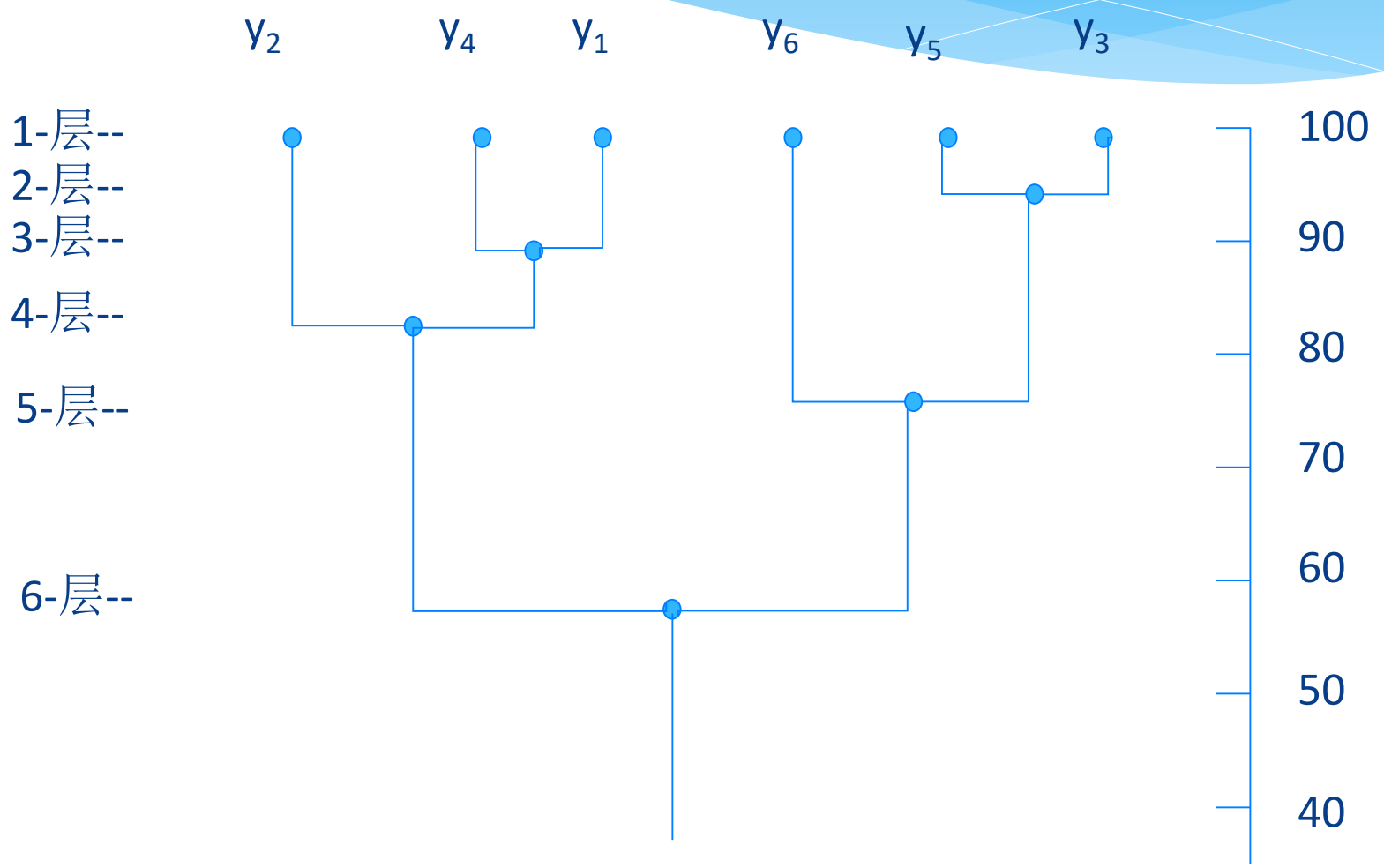
分级聚类方法

* 聚类划分序列：N个样本自底向上逐步合并成一类
  * 每个样本自成一类（划分水平1）
  * K水平划分的进行：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
  * 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$d^{(K)}$ > 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： $\Delta(K_i, K_j) = \min_{\substack{\mathbf{x}\in K_i \\ \mathbf{y}\in K_j}} \delta(\mathbf{x}, \mathbf{y})$
  * 最远距离： $\Delta(K_i, K_j) = \max_{\substack{\mathbf{x}\in K_i \\ \mathbf{y}\in K_j}} \delta(\mathbf{x}, \mathbf{y})$
  * 均值距离： $\Delta(K_i, K_j) = \delta(\mathbf{m}_i, \mathbf{m}_j)$

非监督学习的一些问题

直接方法：概率密度函数的估计问题

根本原因：已知信息的不足

解决办法：　先验知识

多次试算

改进算法。（如 SOM，Fuzzy C-means）

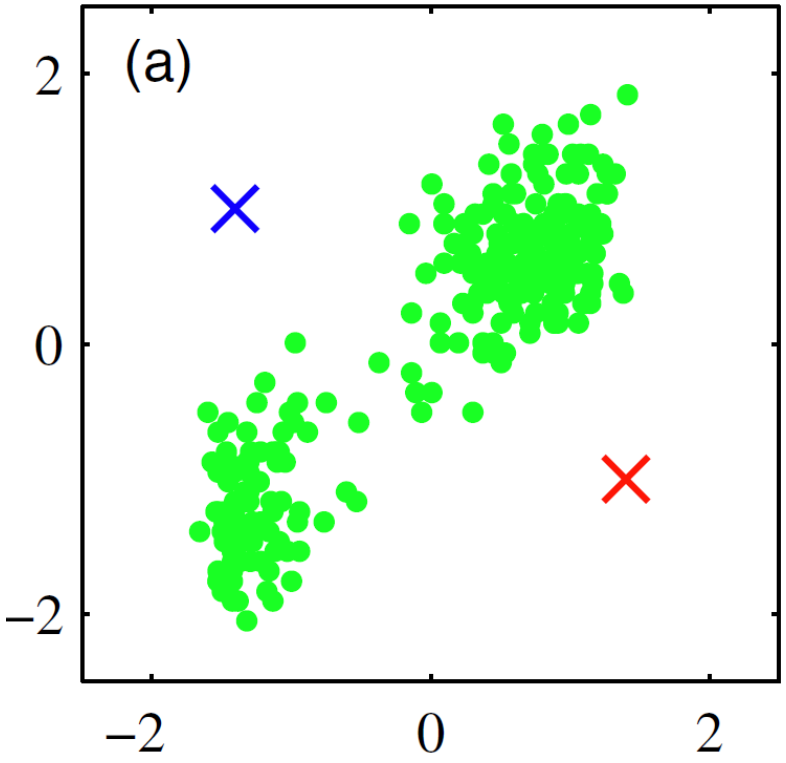
事先确定类别数

对相似性度量的依赖性

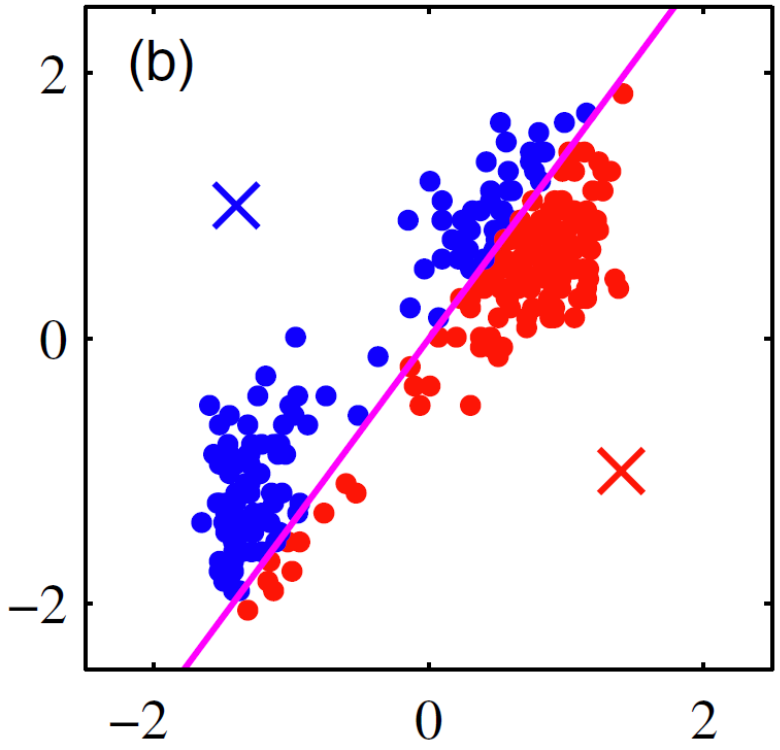
具体实现及应用

* K均值聚类算法
* 基于非监督学习的医学图像分割

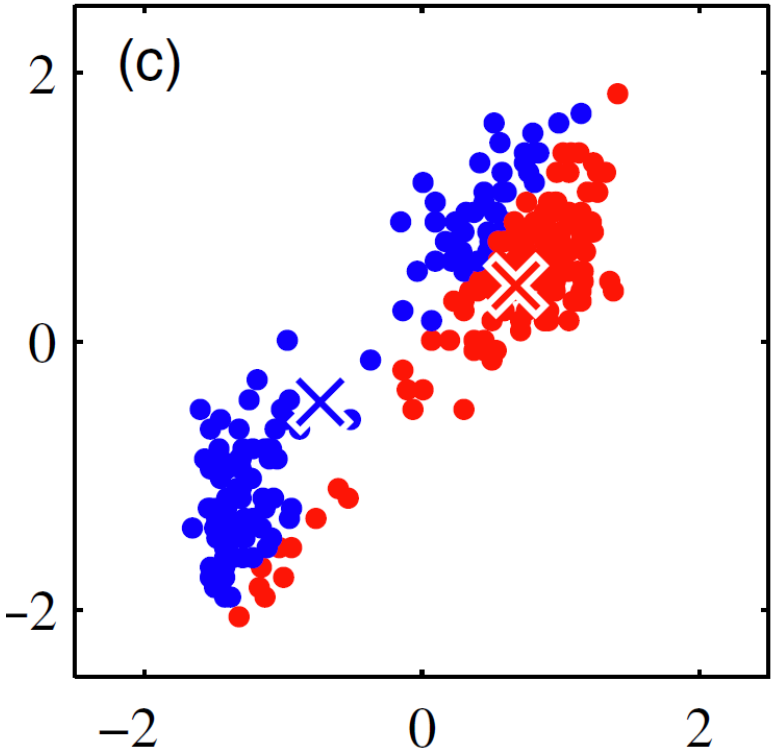
K均值过程示例



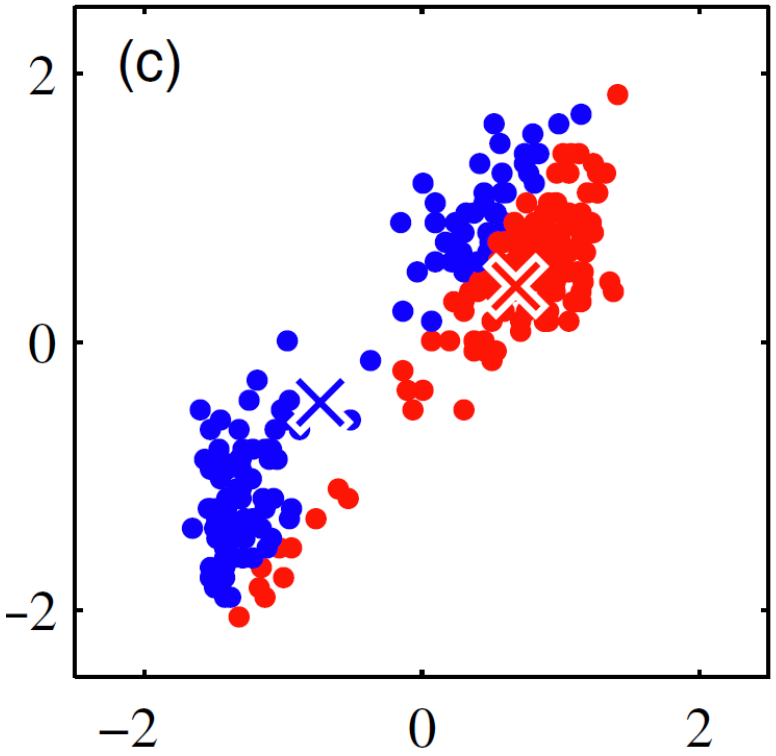
K均值过程示例



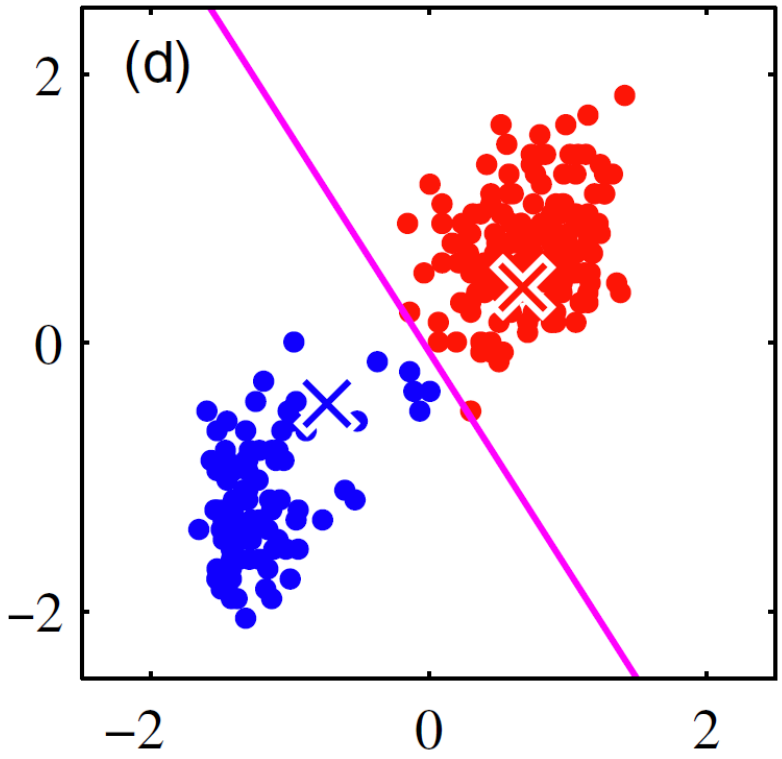
K均值过程示例



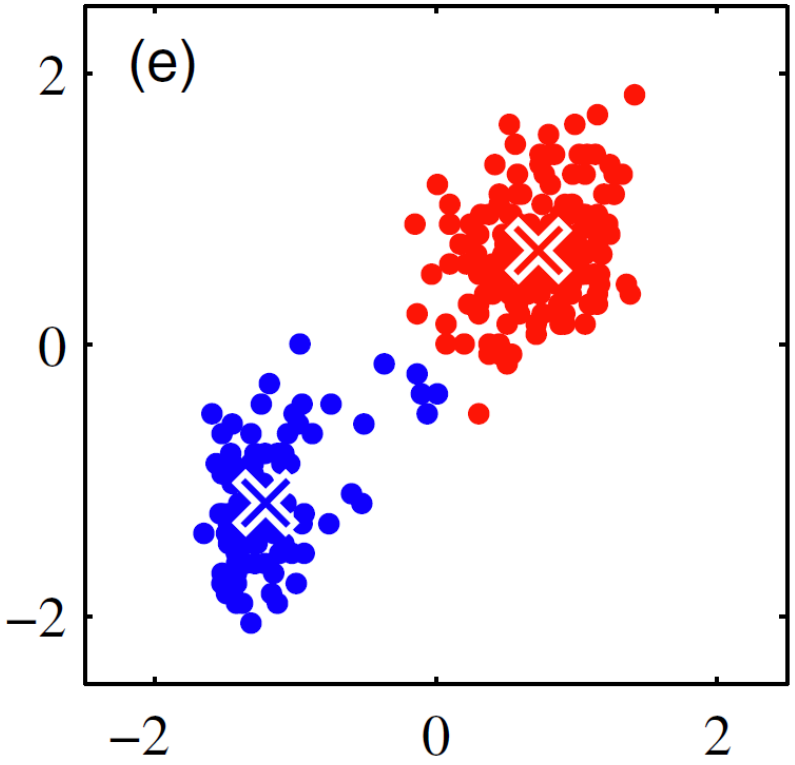
K均值过程示例



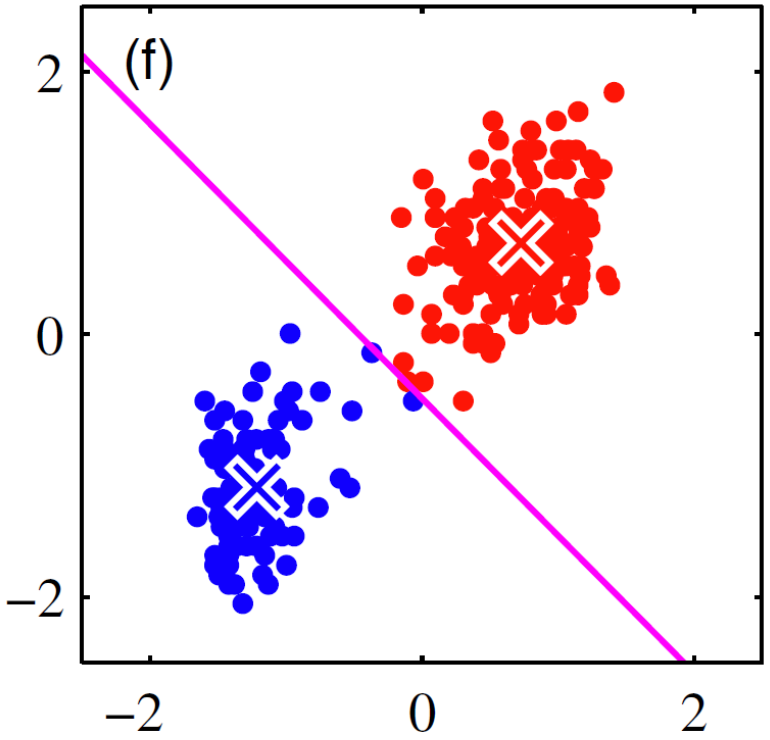
K均值过程示例



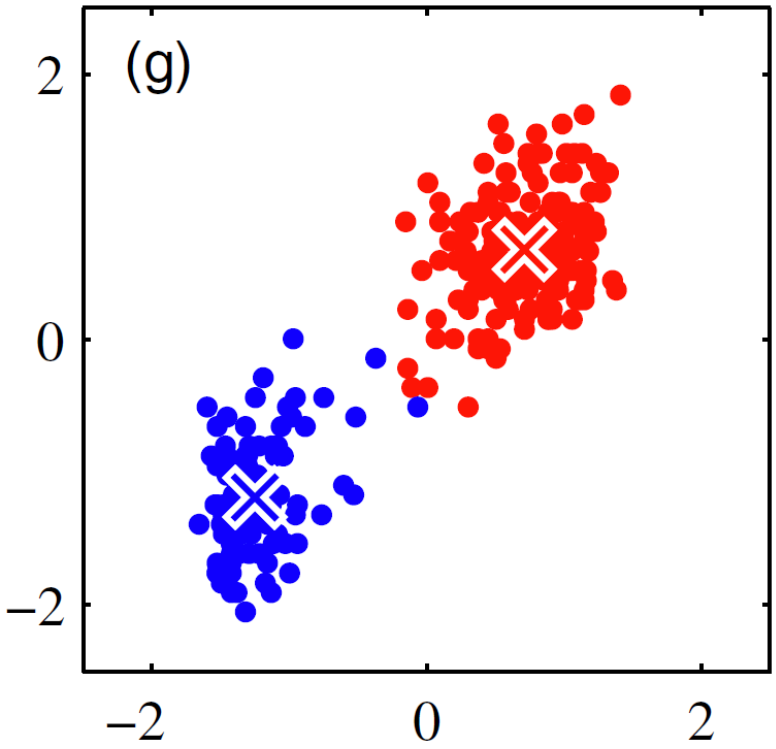
K均值过程示例



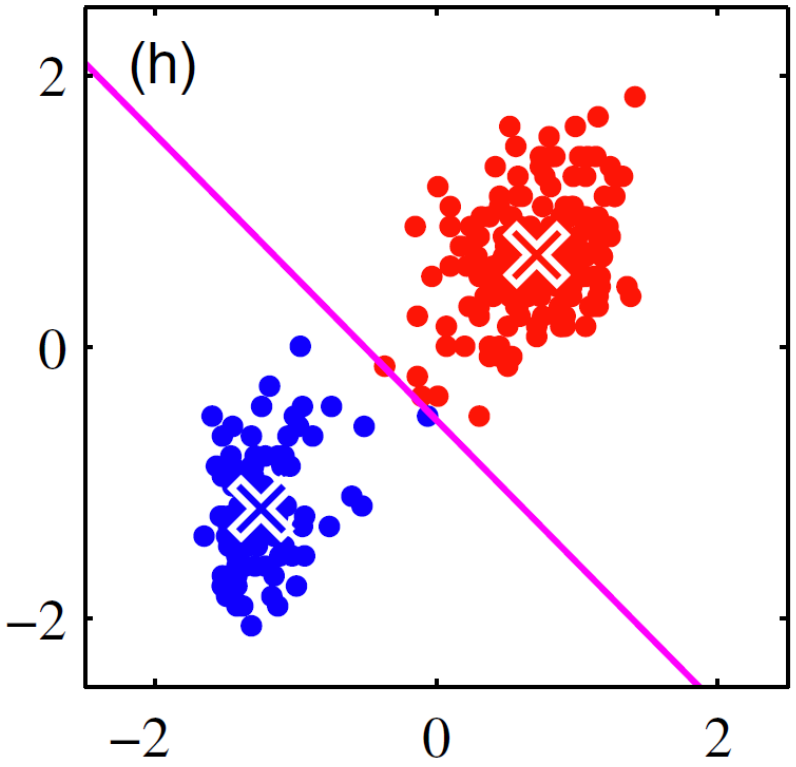
K均值过程示例



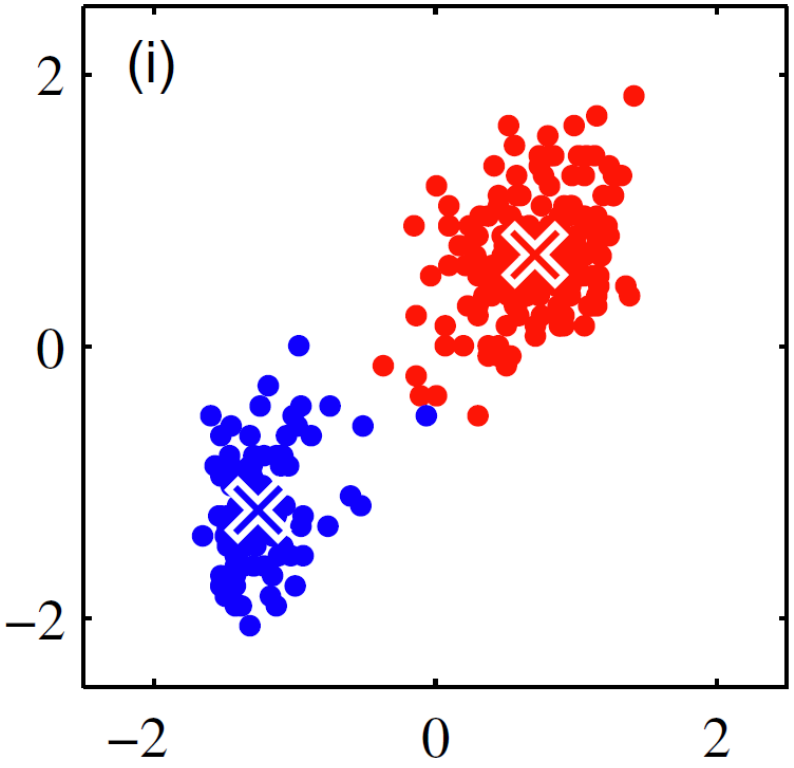
K均值过程示例



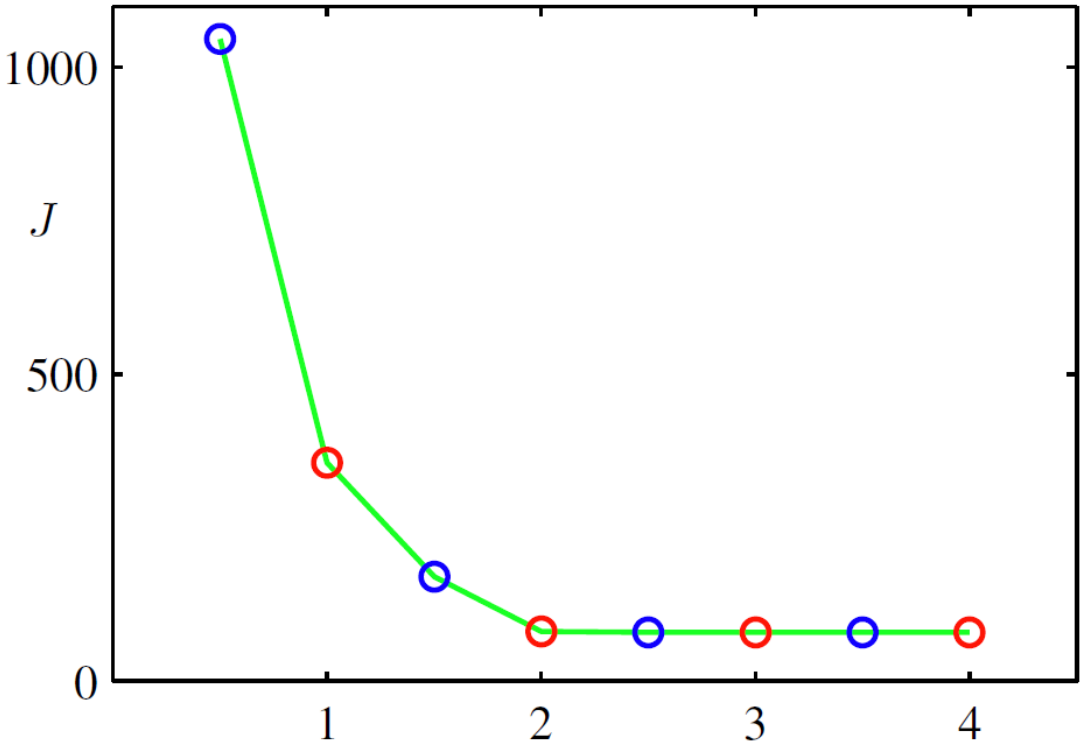
K均值过程示例



K均值过程示例



K均值过程示例



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×N）
* 像素
  * R, G, B
* 数字图像就是一个三维矩阵

实例：基于K均值聚类的图像分割

* K=2

实例：基于K均值聚类的图像分割

* K=3

实例：基于K均值聚类的图像分割

* K=10

问题

* K均值算法如何改进?

谢谢