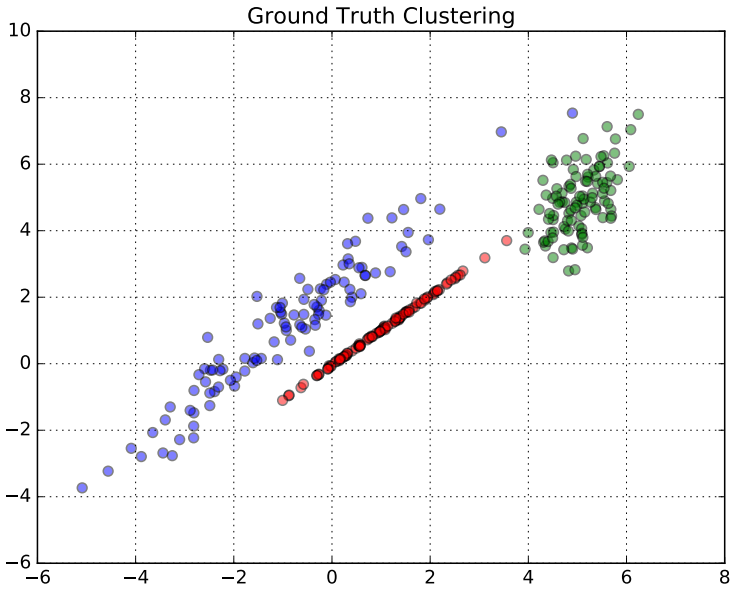
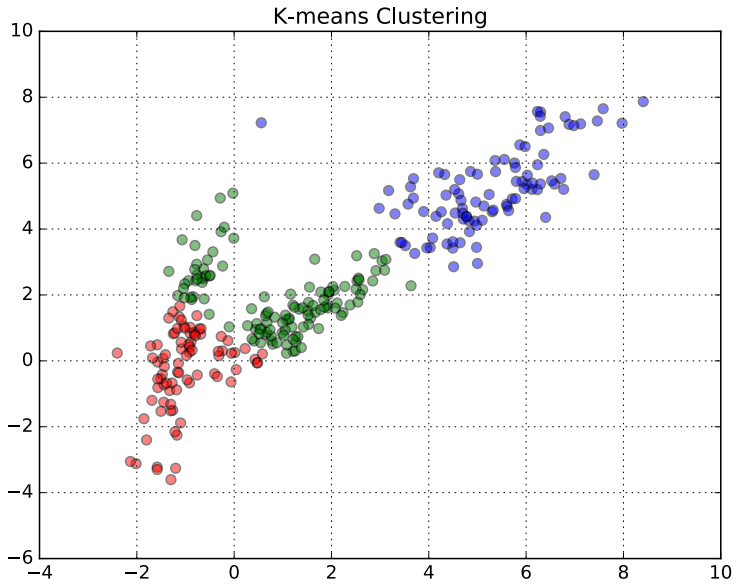
七月在线聚类算法

1.Kmeans
2.Kmeans代码实例
3.层次聚类
4.GMM 高斯混合模型
5.GMM代码实例

李老师：阿里，百度凤巢算法专家

kmeans算法，也被称为k-平均或k-均值，是一种得到最广泛使用的聚类算法。它是将各个聚类子集内的所有数据样本的均值作为该聚类的代表点，算法的主要思想是通过迭代过程把数据集划分为不同的类别，使得评价聚类性能的准则函数达到最优，从而使生成的每个类的类内紧凑，类间独立。





划分聚类方法对数据集进行聚类时包括如下三个要点：

（1）选定某种距离作为数据样本间的相似性度量

在计算数据样本之间的距离时，可以根据实际需要选择欧式距离、曼哈顿距离或者明考斯距离中的一种来作为算法的相似性度量，其中最常用的是欧式距离。

假设给定的数据集 $X=\{x_m \mid m=1,2,...,total\}$，X中的样本用d个描述属性$A_1, A_2 \cdots A_d$来表示，并且d个描述属性都是连续型属性。数据样本$x_i=(x_{i1}, x_{i2}, \cdots x_{id})$，$x_j=(x_{j1}, x_{j2}, \cdots x_{jd})$其中，$x_{i1}, x_{i2}, \cdots x_{id}$和$x_{j1}, x_{j2}, \cdots x_{jd}$分别是样本$x_i$和$x_j$对应d个描述属性$A_1, A_2, \cdots A_d$的具体取值。样本xi和xj之间的相似度通常用它们之间的距离$d(x_i, x_j)$来表示，距离越小，样本x_i和x_j越相似，差异度越小；距离越大，样本x_i和x_j越不相似，差异度越大。

欧式距离公式如下：

$$d(x_i, x_j)=\sqrt{\sum_{k=1}^{d}(x_{ik}-x_{jk})^2}$$

（2）选择评价聚类性能的准则函数

k-means聚类算法使用误差平方和准则函数来评价聚类性能。给定数据集X，其中只包含描述属性。假设X包含k个聚类子集$X_1, X_2, \cdots X_K$；各个聚类子集中的样本数量分别为$n_1, n_2, \cdots, n_k$;各个聚类子集的均值代表点（也称聚类中心）分别为$m_1, m_2, \cdots, m_k$。则误差平方和准则函数公式为：

$$E = \sum_{i=1}^{k} \sum_{p \in X_i} \left\| p - m_i \right\|^2$$

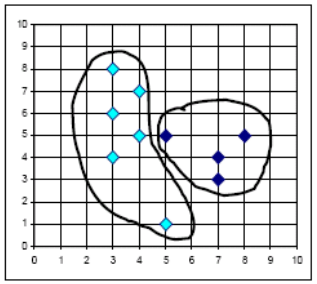
（3）相似度的计算根据一个簇中对象的平均值来进行。

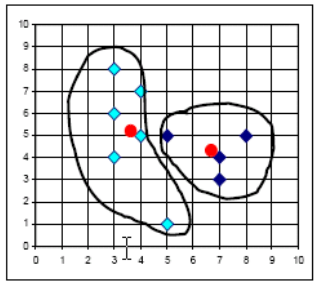
1.将所有对象随机分配到k个簇中。

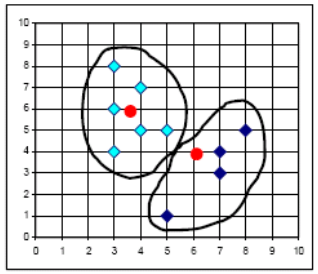
2.计算每个簇的平均值，并用该平均值代表相应的簇。

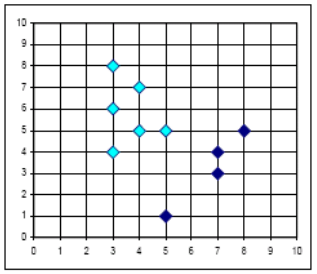
3.根据每个对象与各个簇中心的距离，分配给最近的簇。

4.然后转2，重新计算每个簇的平均值。这个过程不断重复直到满足某个准则函数阈值才停止。









K-means算法2个核心问题：
1.度量样本之间的相关性计算公式，此处采用欧式距离。
2.更新簇内质心的方法，此处采用平均值法，即means。

输入：簇的数目k和n个样本。
输出：使平方误差准则最小的k个簇。
方法：基于簇中样本的平均值。
(1) 任意选择k个样本作为初始的簇中心;
(2) repeat,
(3) 根据簇中样本的平均值，将每个样本(重新) 赋给最类似的簇:
(4) 更新簇的平均值，即计算每个簇中样本的平均值;
(5) until不再发生变化。

例子

O	x	y
1	0	2
2	0	0
3	1.5	0
4	5	0
5	5	2

数据对象集合S见表1，作为一个聚类分析的二维样本，要求的簇的数量k=2。

(1)选择 $O_1(0,2)$，$O_2(0,0)$ 为初始的簇中心，即

$$M_1 = O_1 = (0,2) \qquad M_2 = O_2 = (0,0)$$

(2)对剩余的每个对象，根据其与各个簇中心的距离，将它赋给最近的簇。

对 O_3

$$d(M_1, O_3) = \sqrt{(0-1.5)^2 + (2-0)^2} = 2.5$$

$$d(M_2, O_3) = \sqrt{(0-1.5)^2 + (0-0)^2} = 1.5$$

显然 $d(M_2, O_3) \le d(M_1, O_3)$，故将 O_3 分配给 C_2；更新得到新簇；

计算平方误差准则，单个方差为

$$E_1 = \left[(0-0)^2 + (2-2)^2\right] + \left[(0-5)^2 + (2-2)^2\right] = 25 \qquad E_2 = 27.25$$

总体平均方差是：$E = E_1 + E_2 = 25 + 27.25 = 52.25$

（3）计算新的簇的中心。

$$M_1 = \left((0+5)/2, (2+2)/2\right) = (2.5, 2)$$

$$M_2 = \left((0+1.5+5)/3, (0+0+0)/3\right) = (2.17, 0)$$

重复（2）和（3），得到O_1分配给C_1；O_2分配给C_2，O_3分配给C_2，O_4分配给C_2，O_5分配给C_1。更新得到新簇$C_1 = \{O_1, O_5\}$和$C_2 = \{O_2, O_3, O_4\}$中心$M_1 = (2.5, 2)$$M_2 = (2.17, 0)$单个方差分别为

$$E_1 = \left[(0-2.5)^2 + (2-2)^2\right] + \left[(2.5-5)^2 + (2-2)^2\right] = 12.5 \quad E_2 = 13.15$$

总体平均误差是：$E = E_1 + E_2 = 12.5 + 13.15 = 25.65$

由上可以看出，第一次迭代后，总体平均误差值52.25~25.65，显著减小。由于在两次迭代中，簇中心不变，所以停止迭代过程，算法停止。

k-means算法的性能分析

主要优点：

1. 解决聚类问题的一种经典算法，简单、快速。
2. 对处理大数据集，该算法是相对可伸缩和高效率的。因为它的复杂度是0（n k t），其中，n 是所有对象的数目，k 是簇的数目，t 是迭代的次数。通常k ＜＜n 且t ＜＜n 。
3. 当结果簇是密集的，而簇与簇之间区别明显时，它的效果较好。

主要缺点

1. 在簇的平均值被定义的情况下才能使用，这对于处理符号属性的数据不适用。
2. 必须事先给出k（要生成的簇的数目），而且对初值敏感，对于不同的初始值，可能会导致不同结果。
3. 它对于“躁声”和孤立点数据是敏感的，少量的该类数据能够对平均值产生极大的影响。

k-means算法的改进方法——k-modes 算法

1.k-modes 算法：实现对离散数据的快速聚类，保留了k-means算法的效率同时将k-means的应用范围扩大到离散数据，例如表示人的属性有：姓名、性别、年龄、家庭住址等属性。

2.K-modes算法是按照k-means算法的核心内容进行修改，针对分类属性的度量和更新质心的问题进行改进。
具体如下:

1.度量样本之间的相关性D的计算公式是比较两样本之间，属性相同为0，不同为1，并进行相加。因此D越大，即样本的不相关程度越强（与欧式距离代表的意义是一样的）；
2.更新modes，使用一个簇的每个属性出现频率最大的那个属性值作为代表簇的属性值。

k-means算法的改进方法——k-prototype算法

1.k-Prototype算法：可以对离散与数值属性两种混合的数据进行聚类，在k-prototype中定义了一个对数值与离散属性都计算的相异性度量标准。

2.K-Prototype算法是结合K-Means与K-modes算法，针对混合属性的，解决2个核心问题如下：
1.度量具有混合属性的方法是，数值属性采用K-means方法得到P1，分类属性采用K-modes方法得到P2，那么D=P1+a*P2，a是权重，如果觉得分类属性重要，则增加a，否则减少a，a=0时即只有数值属性
2.更新一个簇的中心结合K-Means与K-modes的更新方法。

k-means算法的改进方法——k-中心点算法

k-中心点算法：k -means算法对于孤立点是敏感的。为了解决这个问题，不采用簇中的平均值作为参照点，可以选用簇中位置最中心的对象，即中心点作为参照点。这样划分方法仍然是基于最小化所有对象与其参照点之间的相异度之和的原则来执行的。

层次聚类

- 层次聚类方法将数据对象组成一棵聚类树。
- 根据层次分解是以自底向上（合并）还是自顶向下（分裂）方式，层次聚类方法可以进一步分为凝聚的和分裂的。
- 一种纯粹的层次聚类方法的质量受限于：一旦合并或分裂执行，就不能修正。也就是说，如果某个合并或分裂决策在后来证明是不好的选择，该方法无法退回并更正。

层次聚类方法

- 一般来说，有两种类型的层次聚类方法：
  - 凝聚层次聚类：采用自底向上策略，首先将每个对象作为单独的一个原子簇，然后合并这些原子簇形成越来越大的簇，直到所有的对象都在一个簇中（层次的最上层），或者达到一个终止条件。绝大多数层次聚类方法属于这一类。
- 输入：n个对象，终止条件簇的数目k。
- 输出：k个簇，达到终止条件规定簇数目。
- (1)将每个对象当成一个初始簇；
- (2)REPEAT
- (3)根据两个簇中最近的数据点找到最近的两个簇；
- (4)合并两个簇，生成新的簇的集合；
- (5)UNTIL达到定义的簇的数目；

层次聚类过程

- 下图描述了一种凝聚层次聚类算法和一种分裂层次聚类算法对一个包含五个对象的数据集合{a, b, c, d, e}的处理过程。

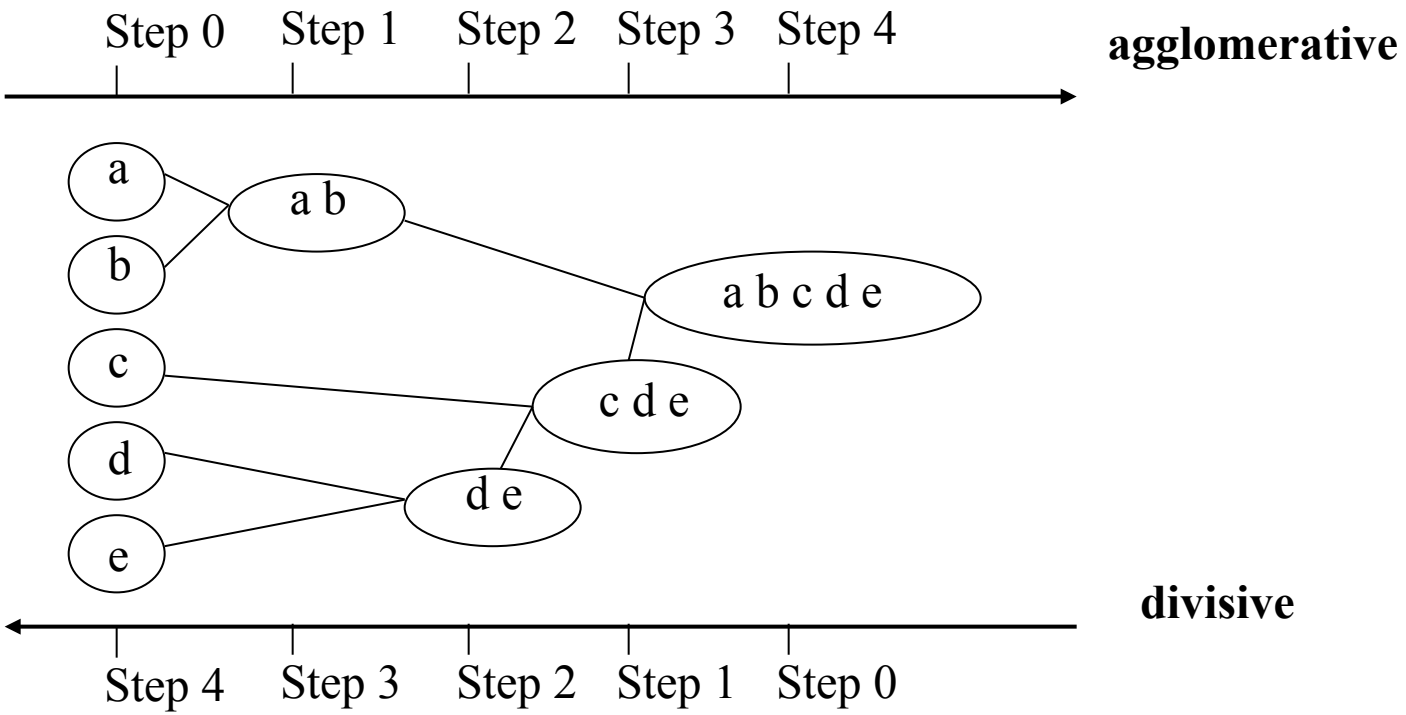


图1 对数据对象{a, b, c, d, e}的凝聚和分裂层次聚类

层次聚类步骤

1.初始，凝聚层次聚类算法将每个对象自为一簇，然后这些簇根据某种准则逐步合并，直到所有的对象最终合并形成一个簇。例如，如果簇C1中的一个对象和簇C2中的一个对象之间的距离是所有属于不同簇的对象间欧氏距离中最小的，则C1和C2合并。

2.在分裂层次聚类算法中，所有的对象用于形成一个初始簇。根据某种原则（如，簇中最近的相邻对象的最大距离），将该簇分裂。簇的分裂过程反复进行，直到最终每个新簇只包含一个对象。

3.在凝聚或者分裂层次聚类方法中，用户可以定义希望得到的簇数目作为一个终止条件。

树状图

通常，使用一种称作树状图的树形结构表示层次聚类的过程。它展示出对象是如何一步步分组的。图2显示图1的五个对象的树状图。

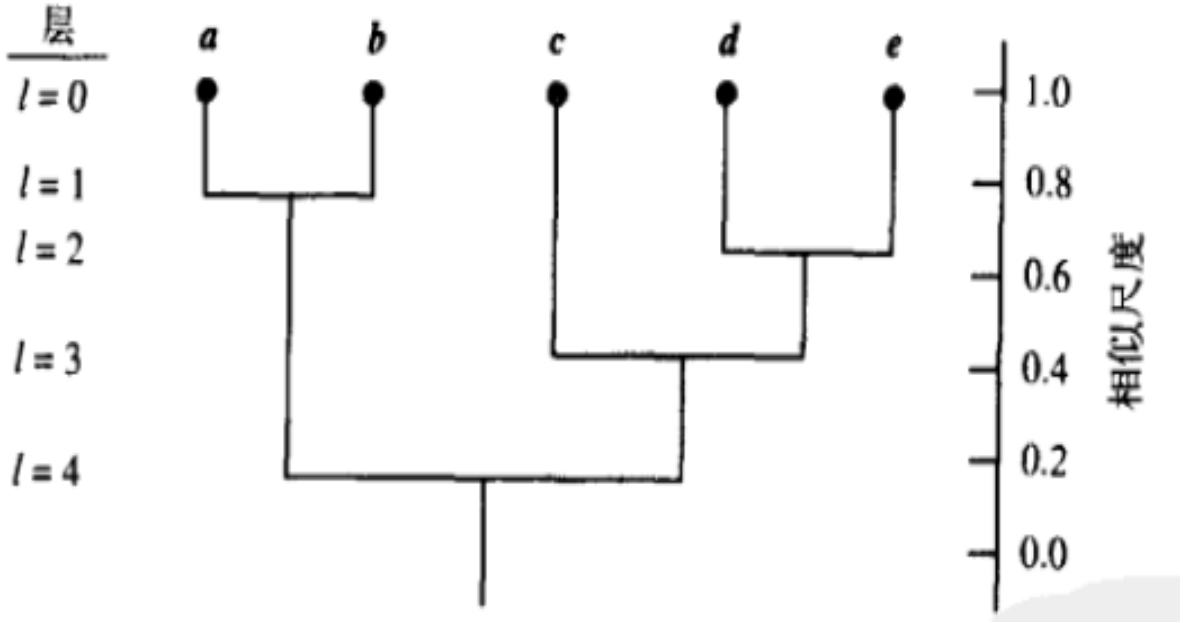


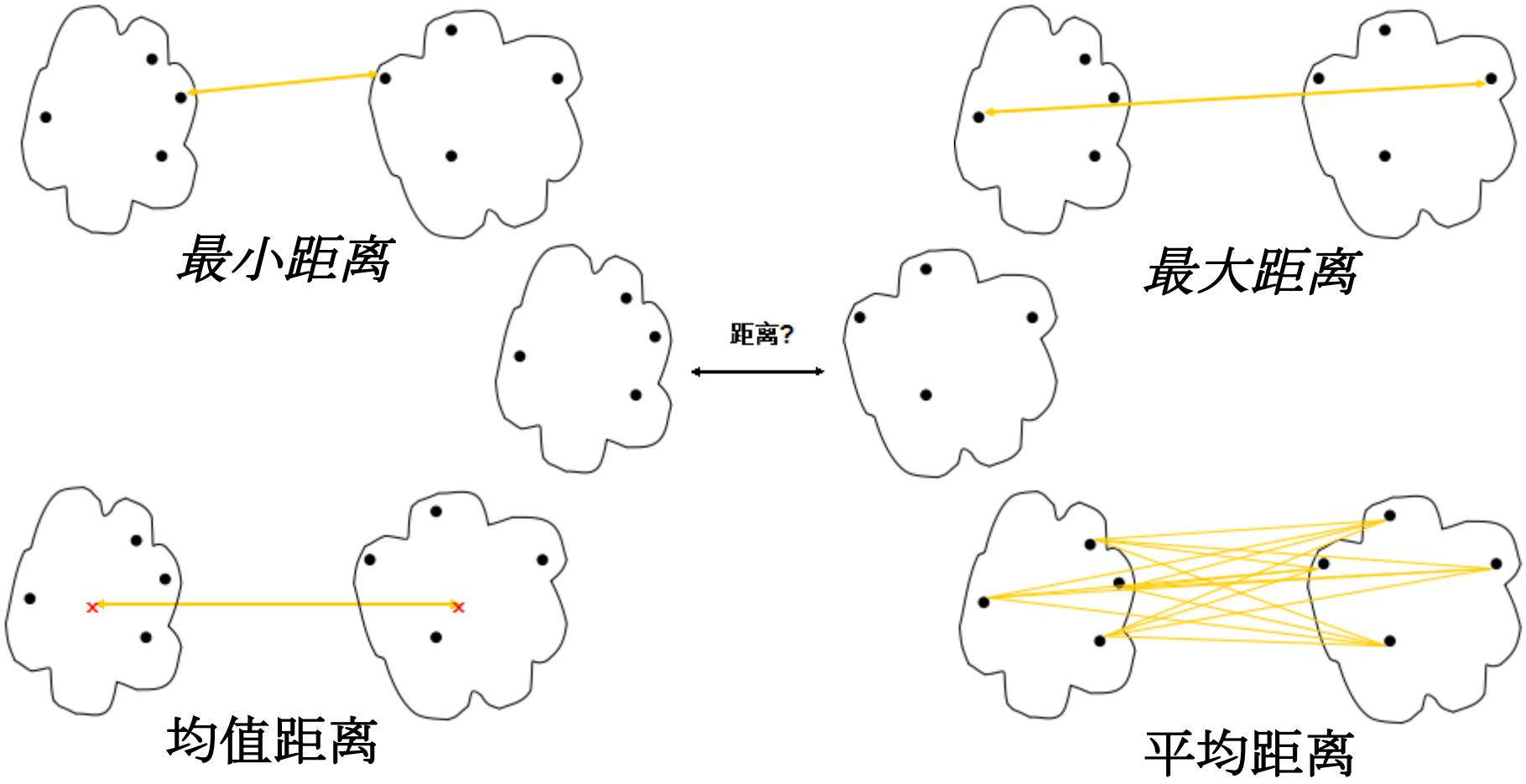
图2 数据对象{a,b,c,d,e}层次聚类的树状图表示

簇间距离

四个广泛采用的簇间距离度量方法如下，其中|p-p'|是两个对象或点p和p'之间的距离，m_i是簇C_i的均值，而n_i是簇C_i中对象的数目。

- 最小距离：$d_{\min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} |p - p'|$
- 最大距离：$d_{\max}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} |p - p'|$
- 均值距离：$d_{mean}(C_i, C_j) = |m_i - m_j|$
- 平均距离：$d_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p \in C_i} \sum_{p' \in C_j} |p - p'|$

簇间距离



1. 当算法使用最小距离 $d_{\min}(C_i,C_j)$ 衡量簇间距离时，有时称它为最近邻聚类算法。此外，如果当最近的簇之间的距离超过某个任意的阈值时聚类过程就会终止，则称其为单连接算法。

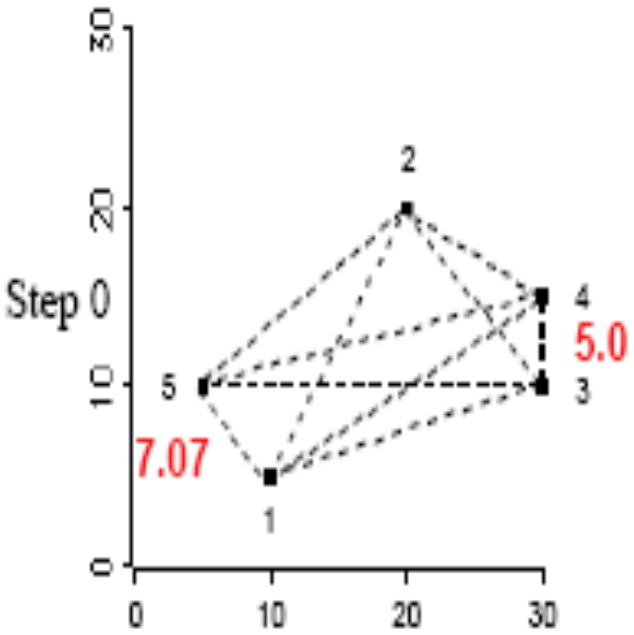
2. 当一个算法使用最大距离 $d_{\max}(C_i,C_j)$ 度量簇间距离时，有时称为最远邻聚类算法。如果当最近簇之间的最大距离超过某个任意阈值时聚类过程便终止，则称其为全连接算法。

单连接算法例子

- 先将五个样本都分别看成是一个簇，最靠近的两个簇是3和4，因为他们具有最小的簇间距离D（3，4）=5.0。
- 第一步：合并簇3和4，得到新簇集合1，2，（34），5

	x1	x2
1	10	5
2	20	20
3	30	10
4	30	15
5	5	10

	1	2	3	4	5
1	0.00				
2	18.0	0.00			
3	20.6	14.1	0.00		
4	22.4	11.2	5.00	0.00	
5	7.07	18.0	25.0	25.5	0.00

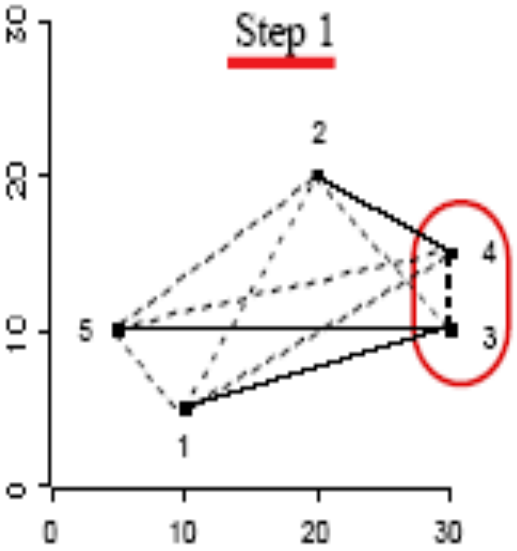


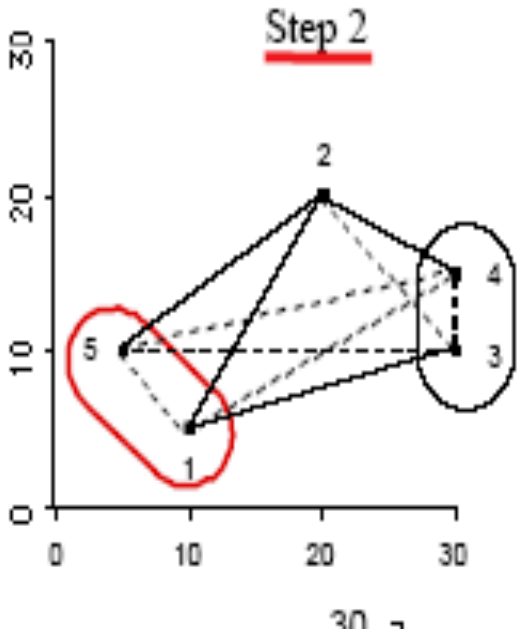
单连接算法例子

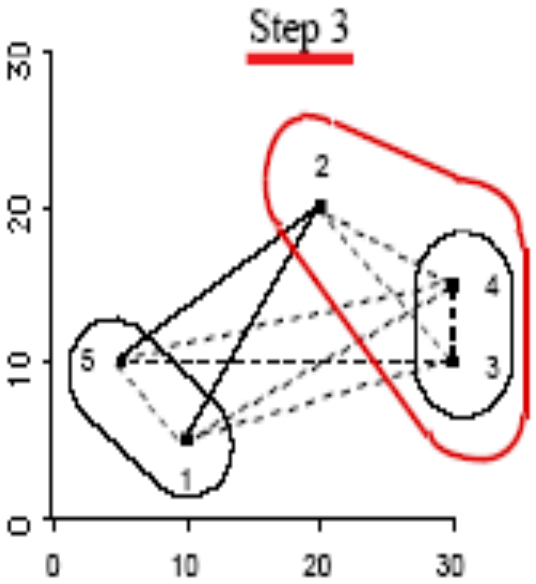
	1	2	5	(34)
1	0.00			
2	18.0	0.00		
5	17.07	18.0	0.00	
(34)	20.6	11.2	25.0	0.00

	2	(34)	(15)
2	0.00		
(34)	11.2	0.00	
(15)	18.0	20.6	0.00

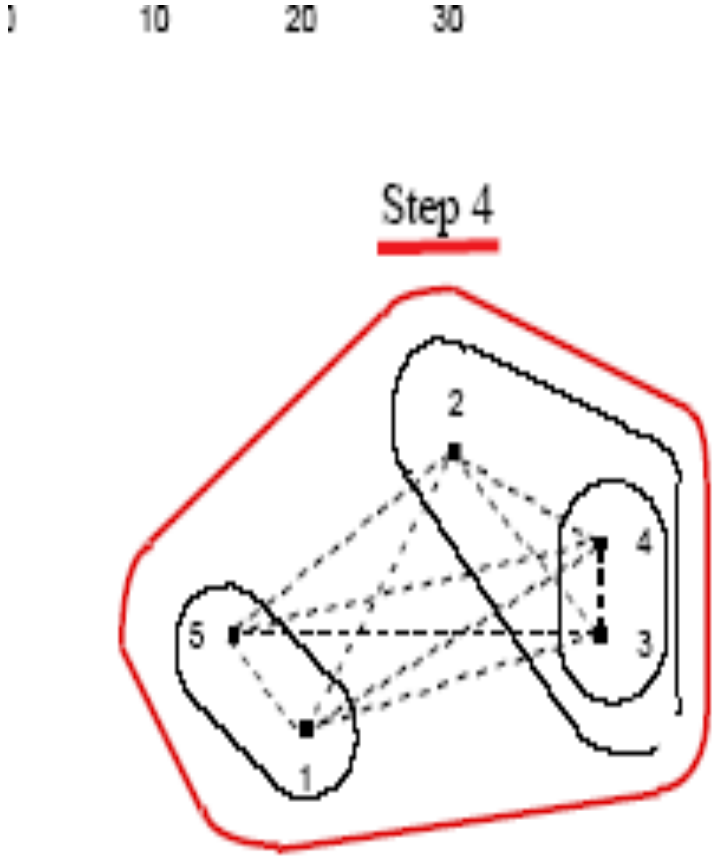
	(15)	(234)
(15)	0.00	
(234)	18.0	0.00



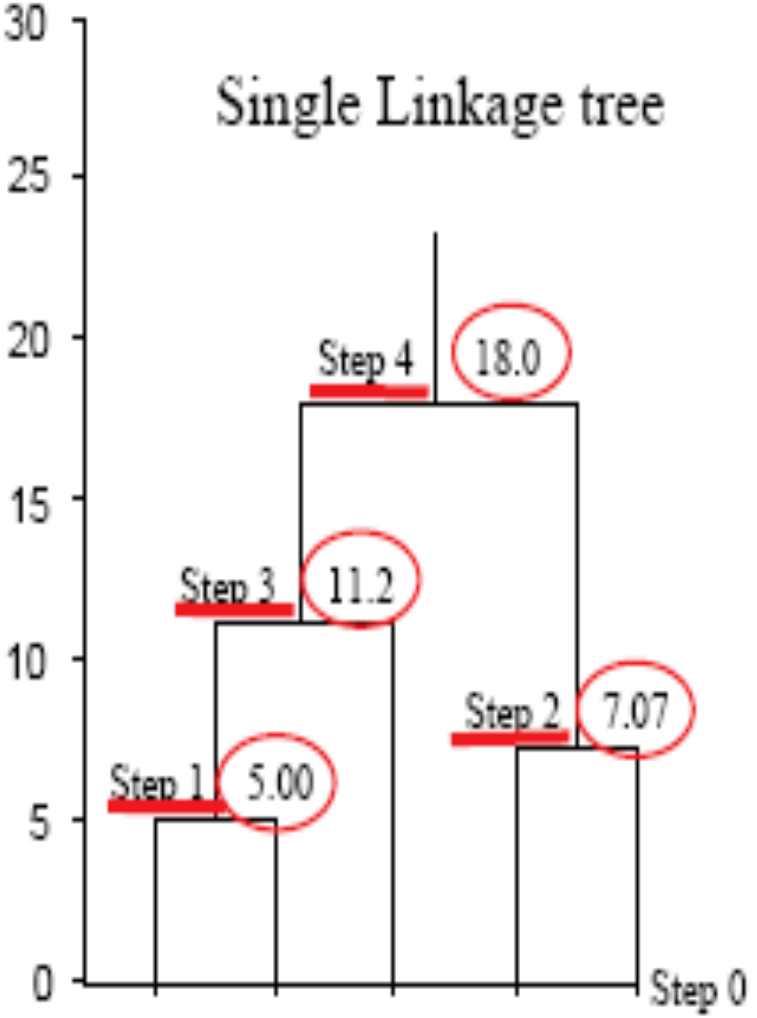




单连接算法例子



** solid lines show the minimum distances between clusters



单连接算法例子

- 最小和最大度量代表了簇间距离度量的两个极端。它们趋向对离群点或噪声数据过分敏感。
- 使用均值距离和平均距离是对最小和最大距离之间的一种折中方法，而且可以克服离群点敏感性问题。
- 尽管均值距离计算简单，但是平均距离也有它的优势，因为它既能处理数值数据又能处理分类数据。

层次聚类方法的困难之处

①层次聚类方法尽管简单，但经常会遇到合并或分裂点选择的困难。这样的决定是非常关键的，因为一旦一组对象合并或者分裂，下一步的处理将对新生成的簇进行。

②不具有很好的可伸缩性，因为合并或分裂的决定需要检查和估算大量的对象或簇。

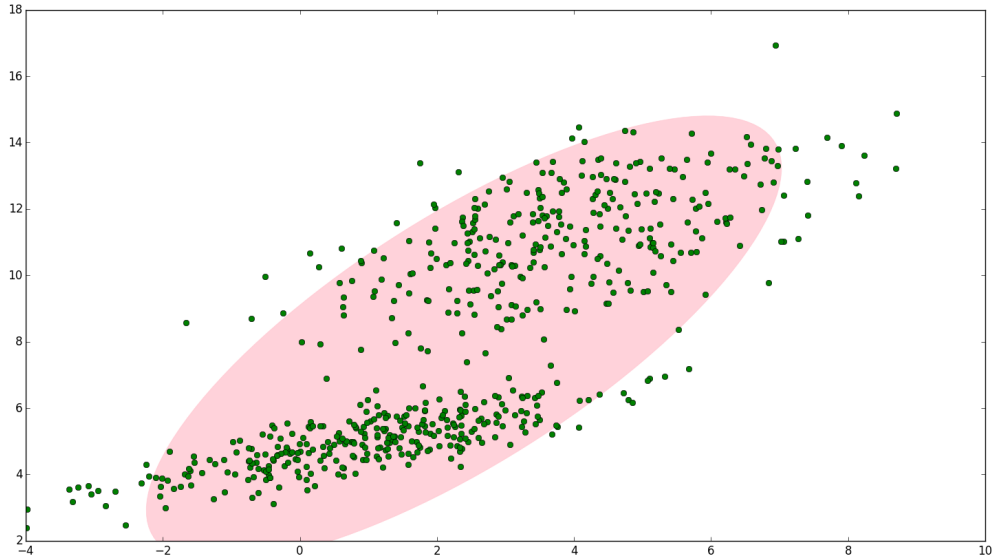
GMM

- 高斯混合模型（Gaussian Mixed Model, GMM）指的是多个高斯分布函数的线性组合，理论上GMM可以拟合出任意类型的分布，通常用于解决同一集合下的数据包含多个不同的分布的情况（或者是同一类分布但参数不一样，或者是不同类型的分布，比如正态分布和伯努利分布）。

- 目的
  - （1）求出每一个样本属于哪个分布
  - （2）求出每一个分布对应的参数

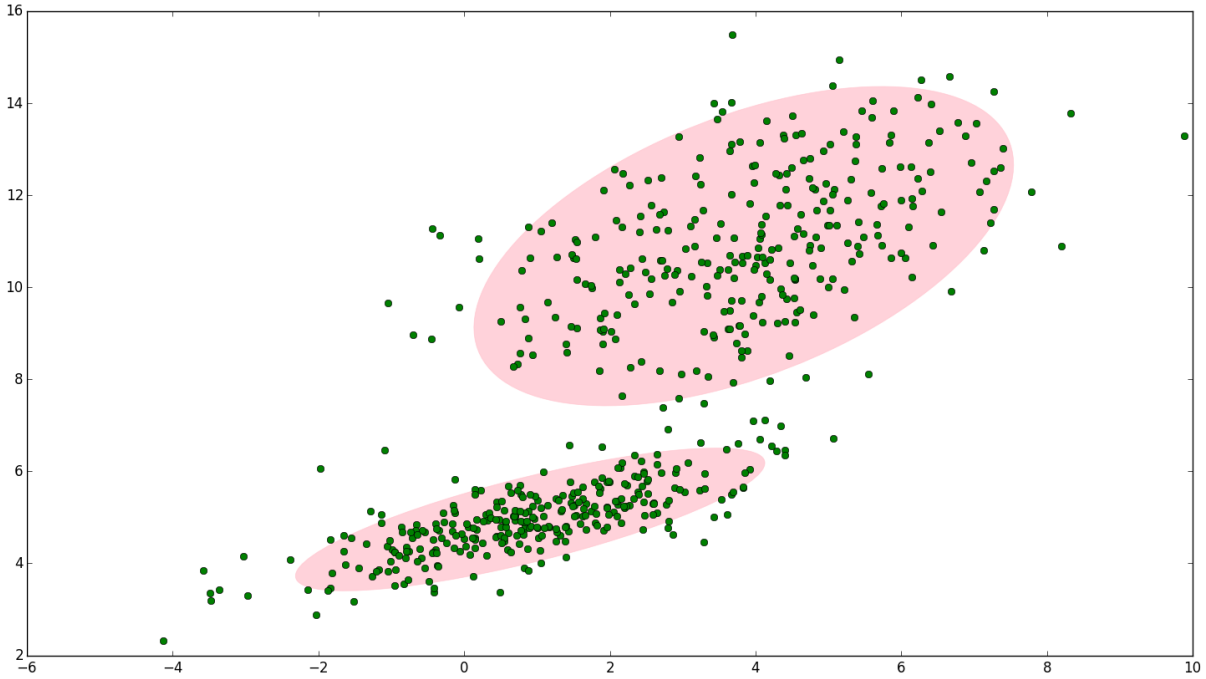
GMM

- 使用一个高斯模型拟合分布



GMM

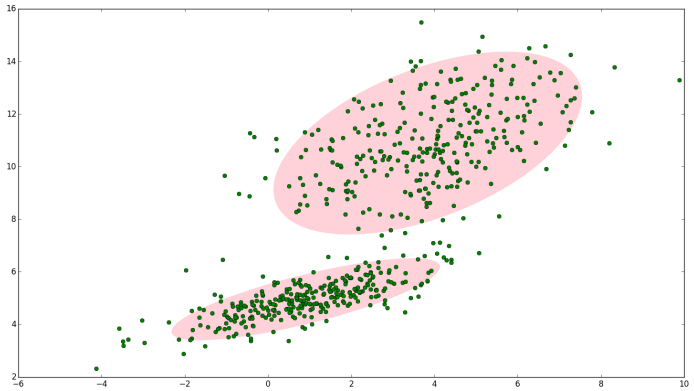
- 使用两个高斯模型拟合分布



GMM

- 原始形式 $p(x)=\sum_{k=1}^{K}\pi_k N\left(x|\mu_k,\sum_k\right)$

权重 π_k可以看成使第k类被选中的概率。



$$0 \leqslant \pi_k \leqslant 1$$

$$\sum_{k=1}^{K} \pi_k = 1$$

GMM的例子

- 例1：一个班级每个学生的身高为 X

  假设男生和女生的身高分别服从高斯分布 $N(\mu_1,\sigma_1^2)$、$N(\mu_2,\sigma_2^2)$

  则 $X \sim \alpha_1 \mathcal{N}(\mu_1,\sigma_1^2)+\alpha_2 \mathcal{N}(\mu_2,\sigma_2^2)$

  其中α_1为男生的比例,

  $$\alpha_1+\alpha_2=1, 0\le\alpha_1,\alpha_2\le 1$$

- 问题：给定独立同分布(independent and identically distributed----IID)的数据 $X_1,...,X_n$求参数

  $(\alpha_1,\mu_1,\sigma_1,\alpha_2,\mu_2,\sigma_2)$

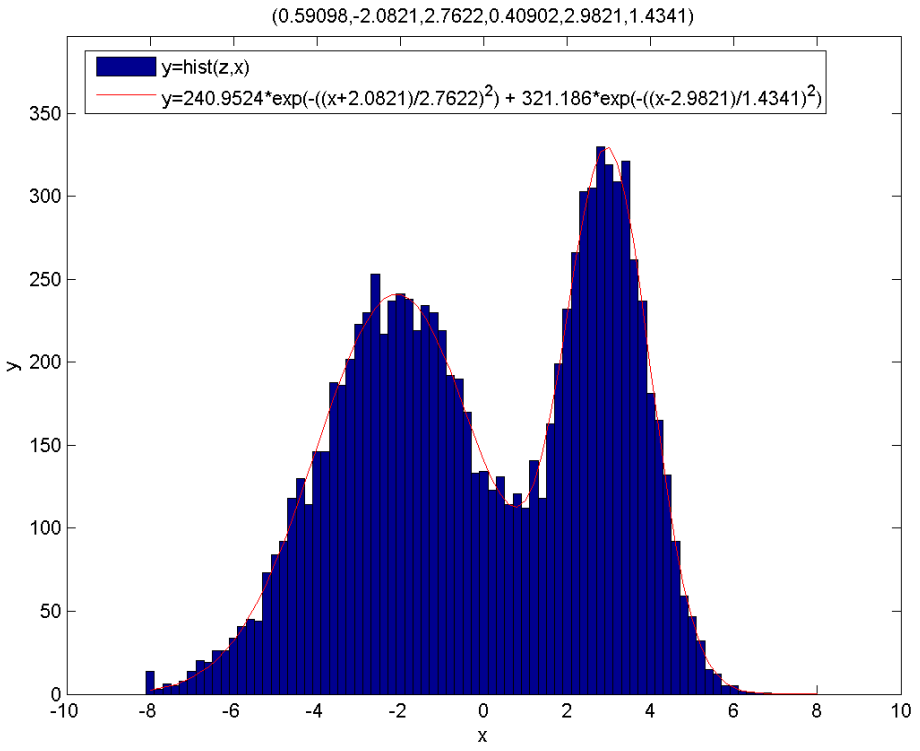
- 混合模型的参数估计是EM(Expectation Maximization)算法最典型的应用

GMM的例子

- 例2 ：$0.4\mathcal{N}(3,1)+0.6\mathcal{N}(-2,4)$分布的随机数的直方图

```
n = 10000;
z = zeros(1,n);
pw1 = 0.6;
u1 = -2;
std1 = 2;
pw2 = 0.4;
u2 = 3;
std2 = 1;
y1 = randn(1,floor(n*pw1))*std1 + u1;
y2 = randn(1,floor(n*pw2))*std2 + u2;
z(1,1:floor(n*pw1)) =y1;
z(1,(floor(n*pw1)+1):n) = y2;
```

$$(\alpha_1,\mu_1,\sigma_1,\alpha_2,\mu_2,\sigma_2)=(0.6,-2,2,0.4,3,1)$$



EM算法

对于给出的GMM

1.初始化参数：均值 μ_k，协方差矩阵 Σ_k，混合系数 π_k

2 E步骤：

$$\gamma(z_{nk}) = \frac{\pi_k N(x_n \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n \mid \mu_j, \Sigma_j)}$$

3.M步骤：通过下列重估共识修正参数

$$\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n \qquad \pi_k^{new} = \frac{N_k}{N}$$

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k^{new})(x_n - \mu_k^{new})^T$$

EM算法

4.检验是否收敛

$$\ln p(\boldsymbol{x}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_k|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

GMM和K-means直观对比

比较GMM和K-means两个算法的步骤:

GMM：

先计算所有数据对每个分模型的响应度

根据响应度计算每个分模型的参数迭代

K-means：

先计算所有数据对于K个点的距离，取距离最近的点作为自己所属于的类

根据上一步的类别划分更新点的位置（点的位置就可以看做是模型参数）

迭代

可以看出GMM和K-means还是有很大的相同点的。GMM中数据对高斯分量的响应度就相当于K-means中的距离计算，GMM中的根据响应度计算高斯分量参数就相当于K-means中计算分类点的位置。然后它们都通过不断迭代达到最优。不同的是：GMM模型给出的是每一个观测点由哪个高斯分量生成的概率，而K-means直接给出一个观测点属于哪一类。

谢谢