第六讲

模式识别

(Pattern Recognition)

武汉大学计算机学院

Email: 18986211797@189.cn

第4章 聚类分析

4.1 分类与聚类的区别

4.2 系统聚类

4.3 分解聚类

4.4 动态聚类(兼顾系统聚类和分解聚类)

4.5 聚类分析编程举例

按照物以类聚的思想，对未知类别的样本集根据样本之间的相似程度分类，相似的归为一类，不相似的归为另一类，故这种分类称为**聚类分析(Clustering analysis)**，又常常叫做“聚类”。

相似性测度、聚类准则和聚类算法称为聚类分析的三要素。

相似性测度用于衡量同类样本的类似性和不同类样本的差异性。常用的测度有：距离、夹角余弦等(详见课件第一讲)。

为了评价聚类效果的好坏，必须定义准则函数。有了模式相似性测度和准则函数后，聚类就变成了使准则函数取极值的优化问题了。常用的准则函数是误差平方和准则。


界
动物界
植物界
门
脊索动物门
被子植物门
纲
哺乳纲
双子叶植物纲
目
食肉目
白花菜目
茄目
科
猫科
犬科
十字花科
茄科
属
猫属
豹属
犬属
芸苔属
萝卜属
辣椒属
种
猫
虎
豹
狼
狗
花菜
油菜
萝卜
辣椒

4.1 分类与聚类的区别

- 分类(Classifying)：用已知类别的样本训练集来设计分类器(有监督学习： supervised learning)

  我们在前面设计分类器时，训练样本集中每个样本的类别归属都是“被标记了”的(labeled)，这种利用已标记样本集的学习方法称为有监督学习方法。

- 聚类/集群(Clustering)：用事先不知样本的类别，而利用样本的先验知识来构造分类器(无监督学习： unsupervised learning)

4.2 系统聚类

系统聚类(又叫层次聚类/谱系聚类法：**Hierarchical Clustering Method)**：先把每个样本(或指标)各自作为一类，然后根据样本间的相似性和相邻性聚合。即将亲疏程度最高的两类合并，如此重复进行，直到所有的样本都合成一类。衡量亲疏程度的指标有两种：距离、相似系数。

相似性、相邻性一般用距离表示。

一、两类间的距离

1.最短距离：两类中相距最近的两样本间的距离。

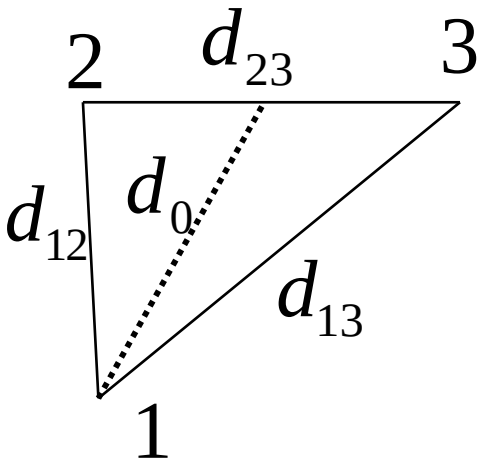
$$D_{pq} = \min_{\substack{x_i \in \omega_p \\ x_j \in \omega_q}} d_{ij}$$

2.最长距离：两类中相距最远的两个样本间的距离。

$$D_{pq} = \max_{\substack{x_i \in \omega_p \\ x_j \in \omega_q}} d_{ij}$$

3.中间距离：最短距离和最长距离都有片面性，因此有时用中间距离。假设某一步将ω_2与ω_3合并为类ω_{23}，需要计算ω_{23}类与某类ω_1的距离。设ω_1类和ω_{23}类间的最短距离为d_{12}，最长距离为d_{13}，ω_{23}类的长度为d_{23}，则类ω_1与类ω_{23}的中间距离定义为：

$$d_0^2 = \frac{1}{2}d_{12}^2 + \frac{1}{2}d_{13} - \frac{1}{4}d_{23}^2$$



上式推广为一般情况：

$$d_0^2 = \frac{1}{2}d_{12}^2 + \frac{1}{2}d_{13} + \beta d_{23}^2$$

其中β为参数，$-\frac{1}{4} \le \beta \le 0$

4.重心距离：两类的均值之间的距离

5.类平均距离：两类中各个元素两两之间的距离平方相加后取平均值

$$D_{pq}^2 = \frac{1}{N_p N_q} \sum_{\substack{x_i \in \omega_p \\ x_j \in \omega_q}} d_{ij}^2$$

其中，$N_p : \omega_p$样本数，$N_q : \omega_q$样本数

d_{ij}为ω_p类点i与ω_q类点j之间的距离

6.离差平方和：

(1)设N个样本原分q类，则定义**第i类内的离差平方和**为：

$$S_i^{(q)} = \sum_{j=1}^{N_i} (\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)^T (\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)$$

其中$\bar{\mathbf{x}}_i$为第i类样本$\mathbf{x}_{ij}$的均值，

N_i为第i类的样本数。

(2)离差平方和增量：设样本已分成ω_p、ω_q两类，若把ω_p、ω_q合并为ω_r类，则定义离差平方：

$$D_{pq}^2 = S_r - (S_p + S_q)$$

其中S_p, S_q分别为ω_p类和ω_q类的离差平方和；

S_r为ω_r类的离差平方和；

增量愈小，合并愈合理。

二、系统聚类/层次聚类算法

系统聚类基本思想：将距离阈值(distance threshold value)作为决定聚类数目的标准，基本思路是每个样本先自成一类，然后按距离准则逐步合并，减少类别数，直到达到分类要求为止。

系统聚类算法步骤描述：

(1)初始分类。 假设有N个样本，每个样本自成一类，则有N类：$G_1(0)$，$G_2(0)$，…，$G_N(0)$｛$G_i(k)$表示第k次合并时的第i类，此步k=0｝。

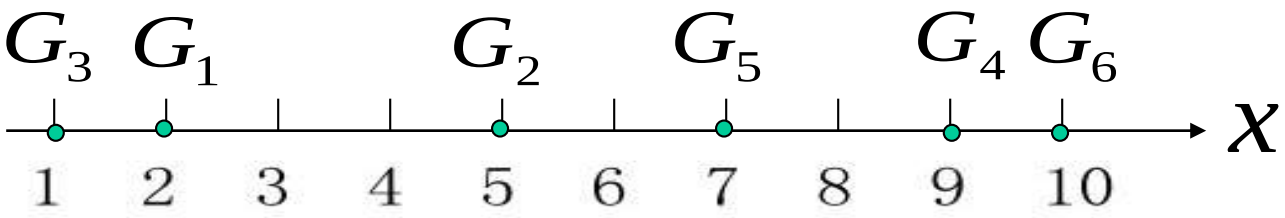
(2)计算各类之间的距离, 得m×m维的距离矩阵**D(**k), m为类别个数(初始时m=N，k=0)。

(3) 找出距离矩阵**D(**k)中类间距离最小的元素，如果它对应着$G_i(k)$和$G_j(k)$，则将类$G_i(k)$与$G_j(k)$合并为一类，由此得到新的聚类$G_1(k+1)$，$G_2(k+1)$，…。令k=k+1，m=m－1，计算距离矩阵D(k)。

(4) 若**D(**k)中类间距离最小值大于距离阈值*T* (*T*hreshold)，则算法停止(这意味着所有类间距均大于要求的阈值*T*，各类已经足够分开了)，所得分类即为聚类结果(或者，如果所有的样本被聚成两类，则算法停止)；否则，转(3)。

系统聚类法将样本逐步聚类，类别由多到少。系统聚类算法的特点是在聚类过程中类的中心不断地调整，但样本一旦归到某个类以后就不会再改变了。

三、系统聚类举例，如下图所示



(1)设全部样本分为6类；

(2)作距离矩阵D(0)，见下表；

	ω_1	ω_2	ω_3	ω_4	ω_5
ω_2	9				
ω_3	1	16			
ω_4	49	16	64		
ω_5	25	4	36	4	
ω_6	64	25	81	1	9

(3)求最小元素；

(4)把ω1,ω3合并ω7=(1,3)； ω4,ω6合并ω8=(4,6)；

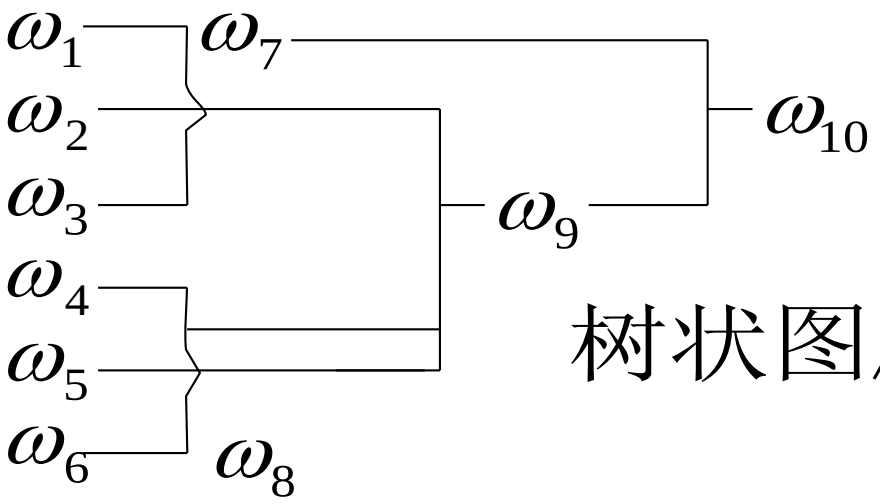
(5)作距离矩阵D(1)；

	ω_7	ω_2	ω_8
ω_2	9		
ω_8	49	16	
ω_5	25	4	4

(6)若合并的类数没有达到要求，转(3)；否则停止。

(3)求最小元素。

(4)$\omega_8,\omega_5,\omega_2$合并，$\omega_9=$（2,5,4,6）。



树状图/系统聚类树

4.3 分解聚类

分解聚类：把全部样本作为一类，然后根据相似性、相邻性分解。

通俗地说，分解聚类法是首先把全部样本当作一类，然后再分为两类，三类，…，直至所有的样本自成一类为止。

目标函数：两类均值(重心)方差

$$E=\frac{N_1N_2}{N}(\overline{x_1}-\overline{x_2})^T(\overline{x_1}-\overline{x_2})$$

N：总样本数，N_1：ω1类样本数，

N_2：ω2类样本数。

下面介绍一分为二的方法：

设有N个样本当作一类记为ω, 将它分成两个子类ω_1 和ω_2且各有N_1 和N_2个样本, 记ω, ω_1, ω_2 的重心分别为 ， ，$\bar{x}$ $\bar{x}_1$ $\bar{x}_2$, ω, ω_1, ω_2 的离差平方和分别为：

$$S = \sum_{\mathbf{x}_i \in \omega} (x_i - \bar{x})^T (x_i - \bar{x})$$

$$S_j = \sum_{\mathbf{x}_i \in \omega_j} (x_i - \bar{x}_j)^T (x_i - \bar{x}_j), \qquad j = 1, 2$$

“一分为二”的思想就是要使S_1+S_2 尽可能小, 或使S- S_1-S_2 尽可能大。

可以证明(此略)：

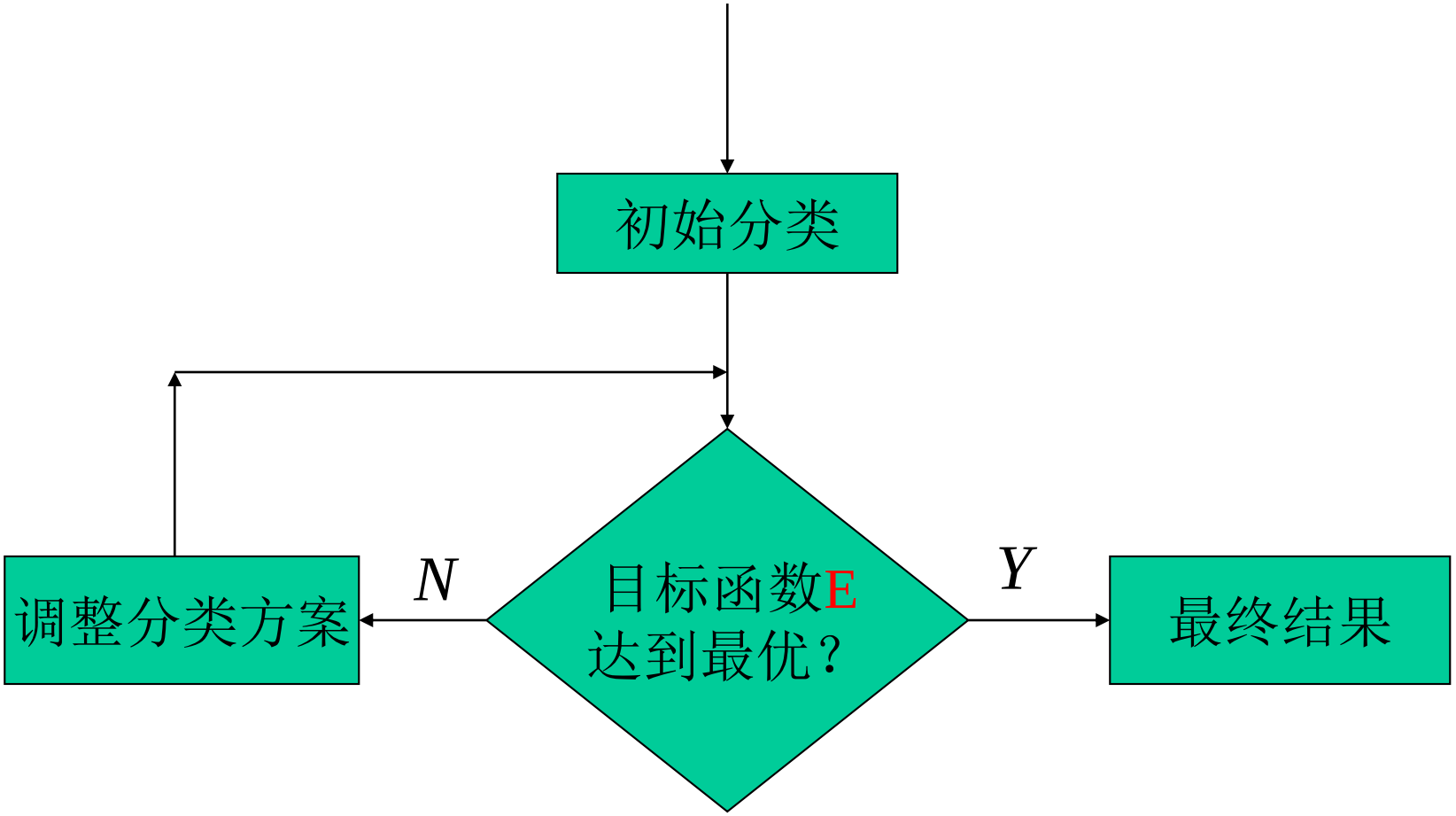
$$E = S - S_1 - S_2$$

$$= \frac{N_1 N_2}{N} (\overline{x_1} - \overline{x_2})^T (\overline{x_1} - \overline{x_2})$$

N：总样本数，N_1：ω1类样本数，
N_2：ω2类样本数。

把上面两类均值(重心)的方差E作为目标函数，选择某种分法使得E达到最大。

❖分解聚类框图



对分法(一分为二法中的一种)：

一开始所有N个样品均在ω1 中，然后找一个样品把它归入ω2 使E达到最大，接着再找第二个样品归入ω2 使E 达到最大，如此继续下去(某样品一旦归入ω2后，该样品在以后的划分中就不再回到ω1) 。

令E(k)表示ω2 中有k个样品，那么一定存在k*使得：

$$V(k^*)= \max_{1\le k\le N-1} E(k)$$

于是将前k*次进入ω2 的样品归为一类，其余n- k*个样品为另一类。再将上述步骤应用于每个子类直至每个样品都自成一类。

举例1：已知21个样本，每个样本有两个特征，原始资料列表如下：

样本号	1	2	3	4	5	6	7	8	9	10
x_1	0	0	2	2	4	4	5	6	6	7
x_2	6	5	5	3	4	3	1	2	1	0

11	12	13	14	15	16	17	18	19	20	21
-4	-2	-3	-3	-5	1	0	0	-1	-1	-3
3	2	2	0	2	1	-1	-2	-1	-3	-5

解：第一次分类时计算所有样本，分别划到G2时的E值，找出最大的。

1、开始时，$G_1^{(0)}=(x_1,x_2,\ldots,x_{21})$　　$G_2^{(0)}=$空

$$\therefore\ \bar{x}_1^{(0)}=\begin{pmatrix}0.714\\1.333\end{pmatrix}\quad \bar{x}_2^{(0)}=\begin{pmatrix}0\\0\end{pmatrix}\quad N_1^{(0)}=21, N_2^{(0)}=0$$

$$\therefore \text{目标函数}\quad E=\frac{N_1N_2}{N}(\bar{x}_1-\bar{x}_2)^T(\bar{x}_1-\bar{x}_2)=0$$

2、分别计算当$x_1,x_2,\ldots,x_n$划入G_2时的E值

把x_1划入G_2时有：

$$\bar{X}_1^{(1)} = \bar{X}_1^{(0)} + \frac{\bar{X}_1^{(0)} - x_1}{N_1^{(0)} - 1}$$

$$= \begin{pmatrix} 0.714 \\ 1.333 \end{pmatrix} + \frac{\left[\begin{pmatrix} 0.714 \\ 1.333 \end{pmatrix} - \begin{pmatrix} 0 \\ 6 \end{pmatrix} \right]}{(21-1)} = \begin{pmatrix} 0.75 \\ 1.10 \end{pmatrix},$$

$$\bar{X}_2^{(1)} = \begin{pmatrix} 0 \\ 6 \end{pmatrix}$$

$$E = \frac{20 \times 1}{21}\left[0.75^2 + (1.10-6)^2\right] = 23.40$$

然后再把$x_2, x_3, \ldots, x_{21}$划入G2时对应的E值，找出一个最大的E值。

把 x_{21}划为G2的E值最大。

$\therefore$ $G_1^{(1)} = (x_1, x_2, \ldots, x_{20}),\ G_2^{(1)} = (x_{21})$

$$\overline{x_1} = \begin{pmatrix} 0.9 \\ 1.65 \end{pmatrix},\quad \overline{x_2} = \begin{pmatrix} -3 \\ -5 \end{pmatrix},\ N_1^{(1)} = 20,\ N_2^{(1)} = 1$$

E(1)=56.6

再继续进行第二、第三次迭代，…；计算出 E(2)，E(3)，…。

次数(k)	G1→G2	E値
1	x_{21}	56.6
2	x_{20}	79.16
3	x_{18}	90.90
4	x_{14}	102.61
5	x_{15}	120.11
6	x_{19}	137.15
7	x_{11}	154.10
8	x_{13}	176.15
9	x_{12}	195.26
10	x_{17}	213.07
11	x_{16}	212.01

第10次迭代 x_{17} 划入G2时，E最大。于是分成以下两类：

$$G_1 = (x_1, x_2, \ldots, x_{10}, x_{16})$$

$$G_2 = (x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{17}, x_{18}, x_{19}, x_{20}, x_{21})$$

每次分类后要重新计算 $\overline{x_1}, \overline{x_2}$ 的值。可用以下递推公式：

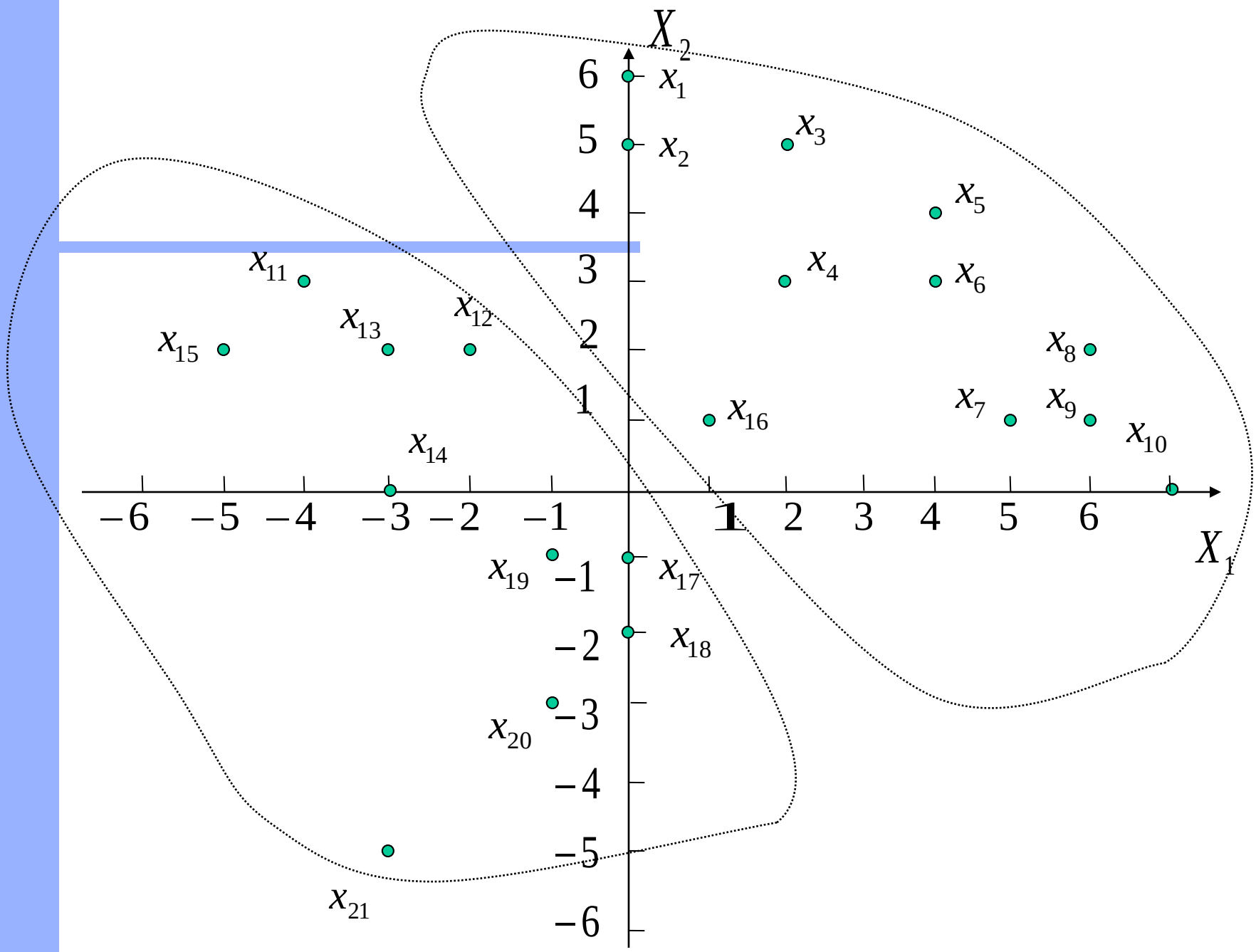
$$x_1^{(k+1)} = \overline{x}_1^{(k)} + (\overline{x}_1^{(k)} - x_i)/(N_1^{(k)} - 1)$$

$$x_2^{(k+1)} = \overline{x}_2^{(k)} - (\overline{x}_2^{(k)} - x_i)/(N_2^{(k)} + 1)$$

$\overline{x}_1^{(k)}, \overline{x}_2^{(k)}$ 是第 k 步对分时两类均值；

x_1^{k+1}, x_2^{k+1} 是下一次对分时把 x_i 从 $G_1^{(k)}$ 划到 $G_2^{(k)}$ 时的两类均值；

$N_1^{(k)}, N_2^{(k)}$ 为二类样本数。


X_2
X_1
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
x_9
x_{10}
x_{11}
x_{12}
x_{13}
x_{14}
x_{15}
x_{16}
x_{17}
x_{18}
x_{19}
x_{20}
x_{21}
6
5
4
3
2
1
−1
−2
−3
−4
−5
−6
−6
−5
−4
−3
−2
−1
1
2
3
4
5
6

4.4 动态聚类(兼顾系统聚类和分解聚类)

一、动态聚类法概要

(Dynamic Clustering Algorithm)

动态聚类法首先选择若干个样本作为聚类中心，再按照事先确定的聚类准则进行聚类。在聚类过程中，根据聚类准则对聚类中心进行反复修改，直到分类合理为止。

动态聚类有如下三个要点：

1) 先选定某种距离作为样本间的相似性度量；

2) 确定评价聚类结果的准则函数；

3) 给出某种初始分类，用迭代法找出使准则函数取极值的最好的聚类结果。

动态聚类法基本思想如下图所示。“动态”即指聚类过程中，聚类中心不断被修改的变化状态。

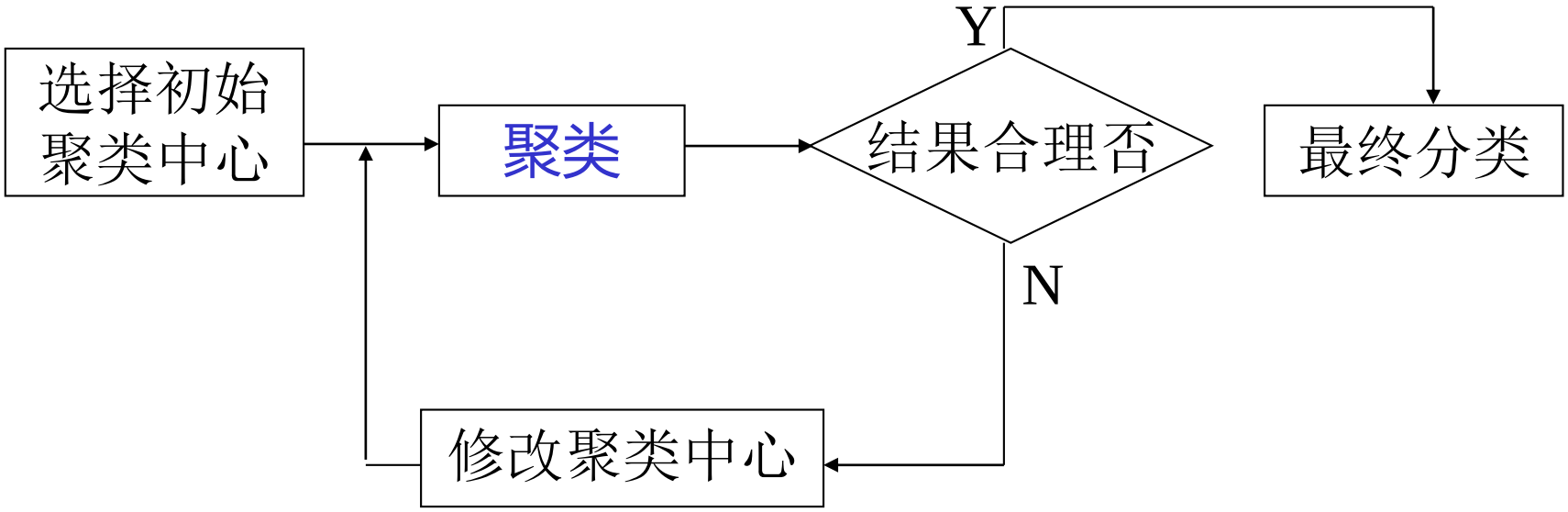


图 动态聚类基本思路框图

二、代表点的选取方法：代表点就是初始分类的聚类中心数k

1.凭经验选代表点，根据问题的性质、数据分布，从直观上看来较合理的代表点k；

2.将全部样本随机分成k类，计算每类重心，把这些重心作为每类的代表点；

3. 按密度大小选代表点

以每个样本作为球心，以d为半径做球形；落在球内的样本数称为该点的密度，并按密度大小排序。首先选密度最大的作为第一个代表点，即第一个聚类中心。再考虑第二大密度点，若第二大密度点距第一代表点的距离大于d_1(人为规定的正数)，则把第二大密度点作为第二代表点，否则不能作为代表点，这样按密度大小考察下去，所选代表点间的距离都大于d_1。d_1太小，代表点太多，d_1太大，代表点太小，一般选$d_1=2d$。对代表点内的密度一般要求大于T。T>0为规定的一个正数。

4. 用前k个样本点作为代表点。

三、初始分类和调整

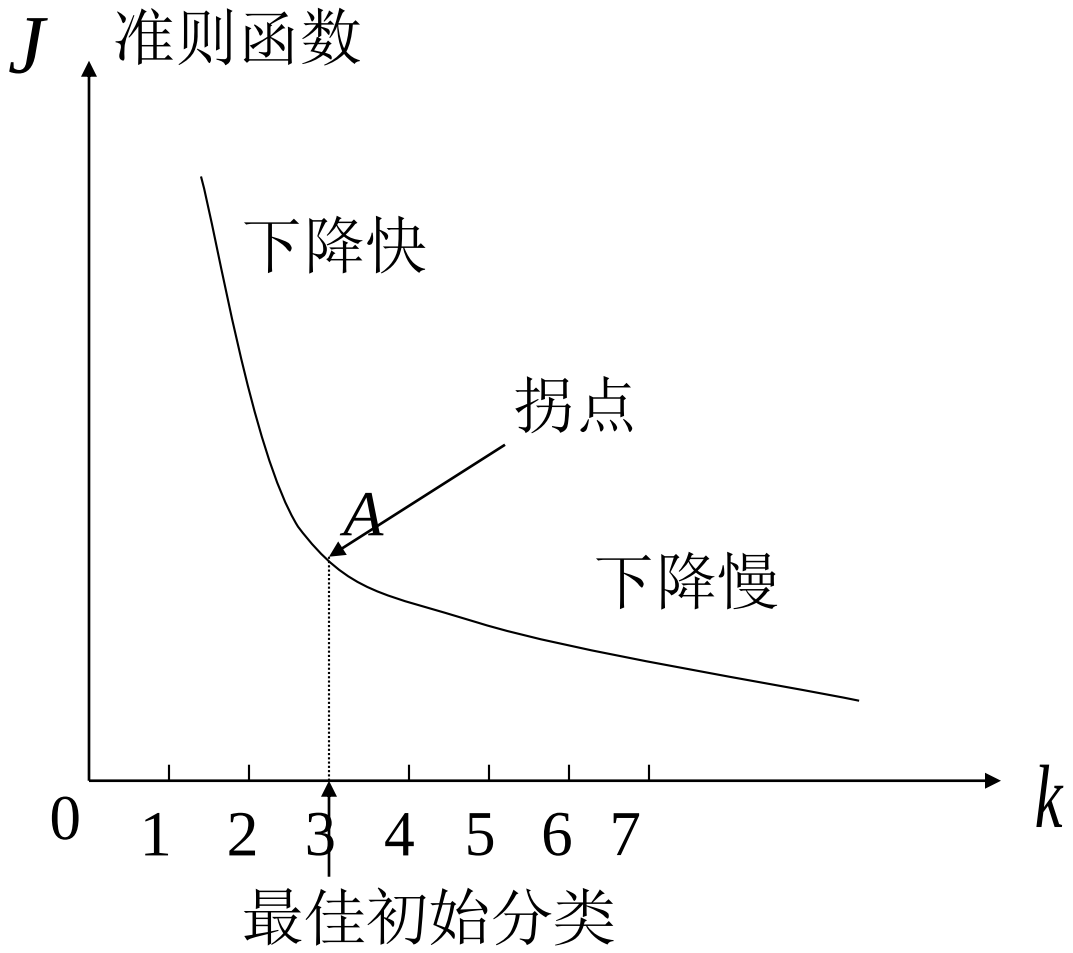
1.**选一批代表点后，代表点就是聚类中心**，计算其他样本到聚类中心的距离，把所有样本归于最近的聚类中心点，形成初始分类，再重新计算各聚类中心，称为成批处理法。

2.选一批代表点后，依次计算其他样本的归类，当计算完第一个样本时，把它归于最近的一类，形成新的分类。再计算新的聚类中心，并计算第二个样本到新的聚类中心的距离，对第二个样本归类。即每个样本的归类都改变一次聚类中心。此法称为逐个处理法。

3.直接用样本进行初始分类，先规定距离d，把第一个样本作为第一类的聚类中心，考察第二个样本，若第二个样本距第一个聚类中心距离小于d，就把第二个样本归于第一类，否则第二个样本就成为第二类的聚类中心，再考虑其他样本，根据样本到聚类中心距离大于还是小于d，决定分裂还是合并。

4.最佳初始分类

如下图所示，随着初始分类k的增大，准则函数下降很快，经过拐点A后，下降速度减慢。拐点A就是最佳初始分类。



四、*K*次平均算法/*K*均值算法：成批处理法 (K/C-means)

*K*均值算法也称*C*均值算法，是根据函数准则进行分类的聚类算法，基于使聚类准则函数最小化。这里所用的**聚类准则函数**是聚类中每一个样本点到类聚类中心的平方和。对于第*j*个聚类集，准则函数定义为

$$J_j = \sum_{i=1}^{N_j} \left\| \mathbf{x}_i - \mathbf{z}_j \right\|^2, \mathbf{x}_i \in G_j$$

式中,G_j表示第*j*个聚类集, 也称为聚类域, 其聚类中心为$\mathbf{z}_j$; N_j为第*j*个聚类域G_j所包含的样本个数.

对所有K个模式类有准则函数:

$$J=\sum_{j=1}^{K}\sum_{i=1}^{N_j}\left\|\mathbf{x}_i-\mathbf{z}_j\right\|^2,\mathbf{x}_i\in G_j$$

K均值算法的聚类准则是:聚类中心$\mathbf{z}_j$的选择应使准则函数J极小,也就是使J_j的值极小,要满足这一点,应有:

$$\frac{\partial J_j}{\partial \mathbf{z}_j}=0$$

即: $$\frac{\partial}{\partial \mathbf{z}_j}\sum_{i=1}^{N_j}\left\|\mathbf{x}_i-\mathbf{z}_j\right\|^2=\frac{\partial}{\partial \mathbf{z}_j}\sum_{i=1}^{N_j}(\mathbf{x}_i-\mathbf{z}_j)^T(\mathbf{x}_i-\mathbf{z}_j)=0$$

可解得:

$$\mathbf{z}_j=\frac{1}{N_j}\sum_{i=1}^{N_j}\mathbf{x}_i,\quad \mathbf{x}_i\in G_j$$

上式表明,G_j类的聚类中心应选为该类样本的均值.

四、K次平均算法/K均值算法：成批处理法 (K/C-means)

K均值算法使用的聚类准则函数是误差平方和准则，通过反复迭代优化聚类结果，使所有样本到各自所属类别的中心的距离平方和达到最小。

条件：设待分类的模式向量集为$\{x_1, x_2, \ldots, x_N\}$，类的数目$K$是事先取定的。

K均值算法步骤：

(1) 任选K个初始聚类中心：$z_1(0), z_2(0), \ldots, z_K(0)$；其中，圆括号中的数字表示聚类过程中的迭代运算次数，用r表示，令初值r=0。

(2) 将待分类的模式特征集$\{x_i\}$中的模式逐个按最小距离原则划分给K类中的某一类，即：

如果 $$d_{il}(r)=\min_j\{d_{ij}(r)\}=\min_j\left\|x_i-z_j(r)\right\| \quad (i=1,2,...,N)$$

则判定 $$x_i\in G_l(r+1)$$

式中$d_{ij}(r)$表示x_i和$G_j(r)$的聚类中心$z_j(r)$的距离，

r表示迭代次数。于是产生新的聚类$G_j(r+1)$ $(j=1,2,...,K)$

(3) 计算重新分类后的各聚类中心，即：

$$\mathbf{z}_j(r+1)=\frac{1}{n_j(r+1)}\sum_{\mathbf{x}\in G_j(r+1)}x_i \qquad (j=1,2,...,K)$$

式中$n_j(r+1)$为$\mathbf{x}\in G_j(r+1)$类中所含模式的个数。

因为这步采取平均的方法计算调整后各类的中心，且定为K类，故称为K-均值法。

(4) 如果$\mathbf{z}_j(r+1)=\mathbf{z}_j(r)(j=1,2,...,K)$，则结束；

否则，r=r+1，转至(2)。

*K*均值算法讨论：

*K*均值算法是否有效主要受以下几个因素的影响：

(1)选取的聚类中心数(代表点)是否符合模式的实际分布；

(2)所选聚类中心的初始位置；

(3)模式样本分布的几何性质；

(4)样本读入的次序。

实际应用中，需要试探不同的*K*值和选择不同的聚类中心初始值。如果模式样本形成几个距离较远的孤立分布的小块区域，一般结果都收敛。

五、K次平均算法举例

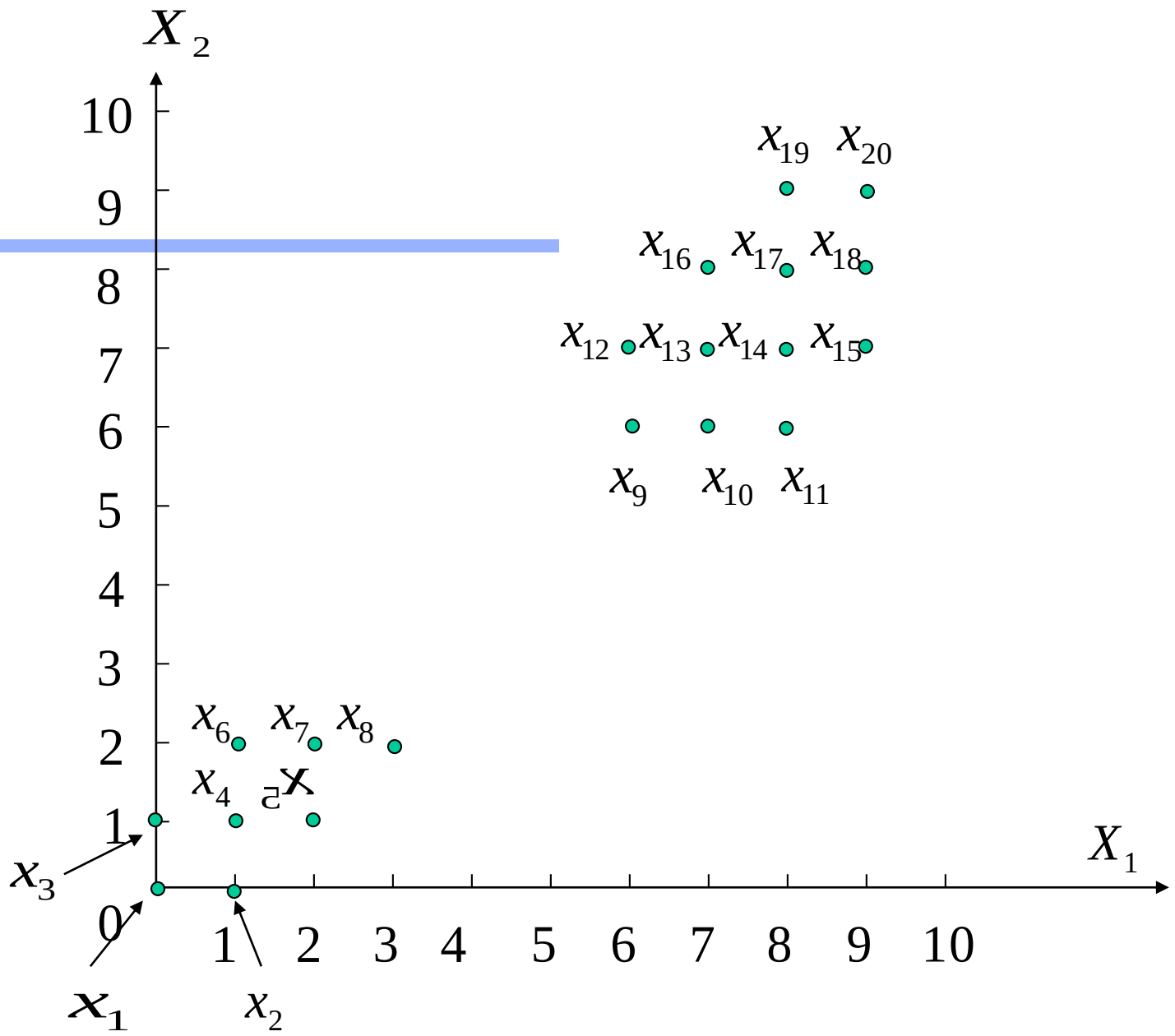
已知有20个样本，每个样本有2个特征，数据分布如下图，K=2，试用K均值法进行聚类。

样本序号	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
特征x_1	0	1	0	1	2	1	2	3	6	7
特征x_2	0	0	1	1	1	2	2	2	6	6

x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}
8	6	7	8	9	7	8	9	8	9
6	7	7	7	7	8	8	8	9	9

第一步：K=2，选初始聚类中心为

$$Z_1(0) = x_1 = (0,0)^T; Z_2(0) = x_2 = (1,0)^T$$


X_2
10
9
8
7
6
5
4
3
2
1
0
x_{19} x_{20}
x_{16} x_{17} x_{18}
x_{12} x_{13} x_{14} x_{15}
x_9 x_{10} x_{11}
x_6 x_7 x_8
x_4 x_5
x_3
X_1
1 2 3 4 5 6 7 8 9 10
x_1
x_2

第二步：$\|x_1 - Z_1(0)\| = \left\| \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\| = 0$

$\|x_1 - Z_2(0)\| = \left\| \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\| = 1$

因为$\|x_1 - Z_1(0)\| < \|x_1 - Z_2(0)\|$

所以$x_1 \in G_1(1)$

$\|x_2 - Z_1(1)\| = \left\| \begin{pmatrix} 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\| = 1$

$$\|x_2 - Z_2(1)\| = \left\| \begin{pmatrix} 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\| = 0$$

因为$\|x_2 - Z_1(1)\| > \|x_2 - Z_2(1)\|$,

所以$x_2 \in G_2(1)$

同理

$\|x_3 - Z_1(1)\| = 1 < \|x_3 - Z_2(1)\| = 2, \therefore x_3 \in G_1(1)$

$\|x_4 - Z_1(1)\| = 2 > \|x_4 - Z_2(1)\| = 1, \therefore x_4 \in G_2(1)$

同样把所有$x_5, x_6, ..., x_{20}$与第二个聚类中心的距离计算出来，判断$x_5, x_6, ..., x_{20}$都属于$G_2(1)$

因此分为两类：

一、$G_1(1) = (x_1, x_3)$,

二、$G_2(1) = (x_2, x_4, x_5, ..., x_{20})$

$N_1 = 2, N_2 = 18$

第三步：计算新的聚类中心

$$Z_1(1)=\frac{1}{N_1}\sum_{x\in G_1(1)}X=\frac{1}{2}(x_1+x_3)=\frac{1}{2}\left[\begin{pmatrix}0\\0\end{pmatrix}+\begin{pmatrix}0\\1\end{pmatrix}\right]$$

$$=\frac{1}{2}\begin{pmatrix}0\\1\end{pmatrix}=(0,0.5)^T$$

$$Z_2(1)=\frac{1}{N_2}\sum_{x\in G_2(1)}X=\frac{1}{18}(x_2+x_4+x_5+\ldots+x_{20})$$

$$=(5.67,5.33)^T$$

第四步：因$Z_j(1) \neq Z_j(0)(j=1,2)$，故r=r+1=1，转第二步’。

第二步’：由新的聚类中心，得

$$\|x_l - Z_1(1)\| < \|x_l - Z_2(1)\| \qquad l = 1,2,...,8$$

$$\|x_l - Z_1(1)\| > \|x_l - Z_2(1)\| \qquad l = 9,10,...,20$$

故得：

$$G_1(2)=(x_1,x_2,...,x_8),N_1=8$$

$$G_2(2)=(x_9,x_{10},...,x_{20}),N_2=12$$

第三步′ ： 计算聚类中心

$$Z_1(2)=\frac{1}{N_1}\sum_{x\in G_1(2)}X=\frac{1}{8}(x_1+x_2+x_3+...+x_8)$$

$$=(1.25,1.13)^T$$

$$Z_2(2)=\frac{1}{N_2}\sum_{x\in G_2(2)}X=\frac{1}{12}(x_9+x_{10}+...+x_{20})$$

$$=(7.67,7.33)^T$$

第四步’：因$Z_j(2) \neq Z_j(1), j=1,2, r=r+1=2,$转第二步。

第二步’’：重新计算$x_1, x_2, ..., x_{20}$到$Z_1(2), Z_2(2)$的距离，

分别把$x_1, x_2, ..., x_{20}$归于最近的那个聚类中心，

重新分为二类$G_1(3)=(x_1, x_2, ..., x_8)$

$G_2(3)=(x_9, x_{10}, ..., x_{20}), N_1=8, N_2=12$

第三步’’：更新聚类中心

$Z_1(3)=Z_1(2)=(1.25,1.13)^T$

$Z_2(3)=Z_2(2)=(7.67,7.33)^T$

第四步’’：

因$Z_j(3)=Z_j(2)(j=1,2)$，不再出现新的类别划分，故分类过程结束。

4.5 聚类分析编程举例

一.MATLAB中常用的计算距离的函数

MATLAB软件包中主要使用系统聚类法。

设有$m \times n$阶的数据矩阵$\mathbf{x} = (x_1, x_2, ..., x_n)$，每一行是一个样本数据。

MATLAB中常用计算样本点间距离的函数如下：

```
y=pdist(x)                  %计算样本点之间的欧氏距离
y=pdist(x,'mahal')          %计算样本点之间的马氏距离
y=pdist(x,'minkowski',p)    %计算样本点之间的明考夫斯基距离
y=pdist(x,'cosine')         %计算样本点之间的余弦距离
```

另外，函数yy=sqareform(y)表示将样本点之间的距离用矩阵的形式输出。

二. 创建系统聚类树

设已经得到样本点之间的距离y, 可以用linkage函数创建系统聚类树，格式为：

z=linkage(y)%用最短距离法创建系统聚类树

其中：z为一个包含聚类树信息的(m-1)×3的矩阵，前两列为索引标识，表示哪两个序号的样本可以聚为同一类；第三列为这两个样本之间的距离；另外，除了m个样本以外，对于每次新产生的类，依次用m+1、m+2、...来标识。如：

z=

2.000 5.000 0.2

3.000 4.000 1.28

则z的第一行表示第2、第5个样本点连接为一个类，它们的距离为0.2;z的第二行表示第3、第4个样本点连接为一个类，它们的距离为1.28。

二. 创建系统聚类树

MATLAB创建聚类树的函数**linkage**：

```
z=linkage(y)                %表示用最短距离法创建系统聚类树
z=linkage(y,'single')       %表示用最短距离法创建系统聚类树
z=linkage(y,'complete')     %表示用最长距离法创建系统聚类树
z=linkage(y,'average')      %表示用平均距离法创建系统聚类树
z=linkage(y,'centroid')     %表示用重心距离法创建系统聚类树
z=linkage(y,'ward')         %表示用离差平方和递增法创建系统聚类树
```

为了表示z矩阵，我们可以用更直观的聚类树状图来展示，方法为使用**dendrogram(z)** 产生的聚类树状图，图的最下边表示样本，然后一级一级往上聚类，最终成为最顶端的一类；纵轴高度代表距离列。

三. 聚类分析程序举例

例1：在MATLAB中编辑名为ex6_1.m的文件。

```
%Filename:ex6_1.m
%创建系统聚类树
x=[3 1.7;1 1;2 3;2 2.5;1.2 1;1.1 1.5;3 1];
y=pdist(x,'mahal');
yy=squareform(y)
z=linkage(y,'centroid') %用重心距离法创建系统聚类树
h=dendrogram(z)         %生成树状图（这里即为聚类树状图）
```

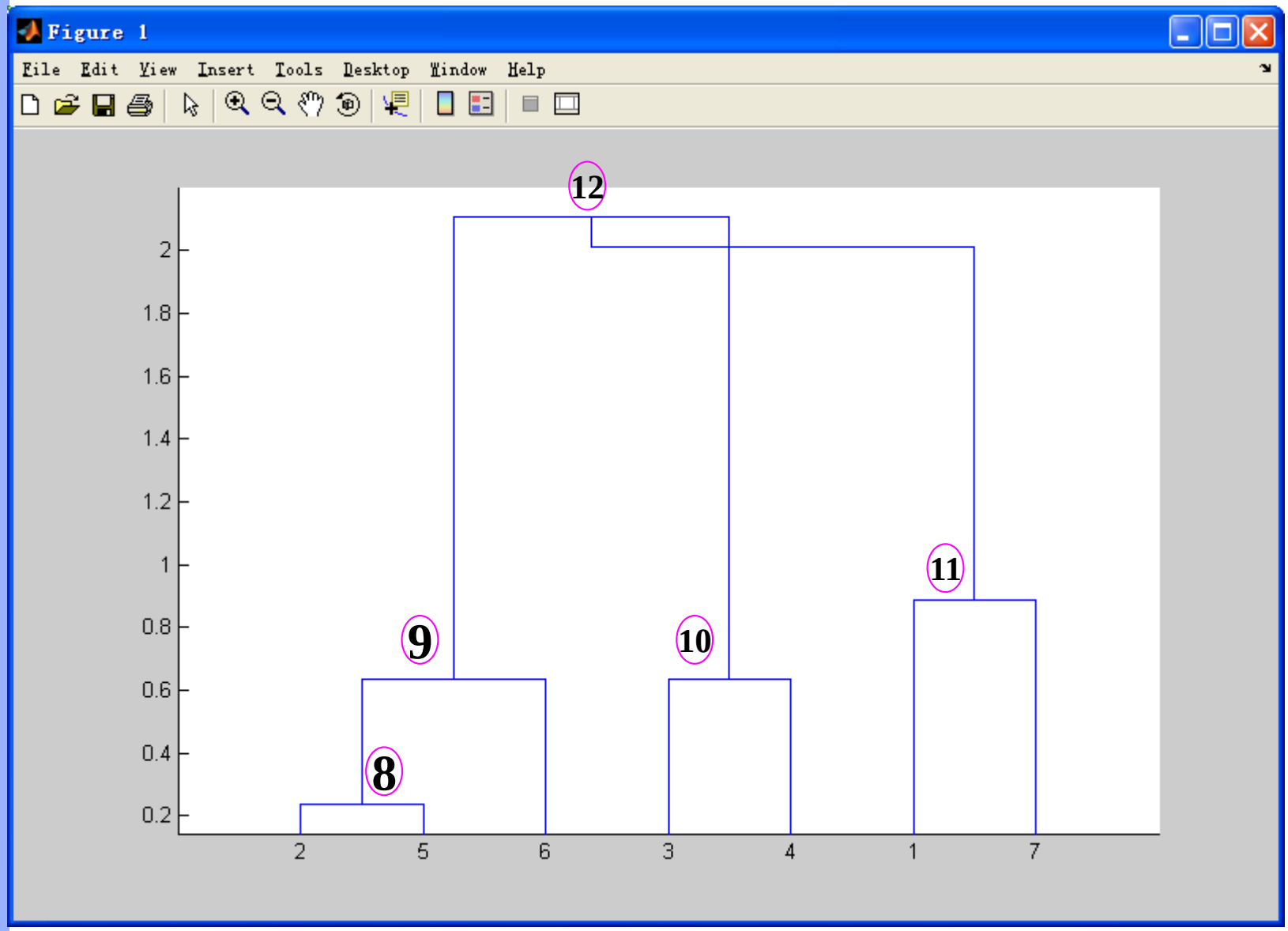
yy =

0	2.3879	2.1983	1.6946	2.1684	2.2284	0.8895
2.3879	0	2.6097	2.0616	0.2378	0.6255	2.3778
2.1983	2.6097	0	0.6353	2.5522	2.0153	2.9890
1.6946	2.0616	0.6353	0	1.9750	1.5106	2.4172
2.1684	0.2378	2.5522	1.9750	0	0.6666	2.1400
2.2284	0.6255	2.0153	1.5106	0.6666	0	2.4517
0.8895	2.3778	2.9890	2.4172	2.1400	2.4517	0

z =

2.0000	5.0000	0.2378
6.0000	8.0000	0.6353
3.0000	4.0000	0.6353
1.0000	7.0000	0.8895
9.0000	10.0000	2.1063
11.0000	12.0000	2.0117

按重心距离法得到的系统聚类树如下：



MATLAB聚类分析方法：

1.根据系统聚类树创建聚类

根据已求出的系统聚类树z，使用cluster函数构造聚类。

例2： 根据系统聚类树创建聚类。

```
%Filename:ex6_2.m
x=[3 1.7;1 1;2 3;2 2.5;1.2 1;1.1 1.5;3 1];
y=pdist(x,'mahal');
yy=squareform(y)
z=linkage(y,‘single’) %用最近距离法创建系统聚类树
h=dendrogram(z)     %生成树状图（这里即为聚类树状图）
t=cluster(z,3)          %分成三个聚类
```

输出结果：

t =

3
1
2
2
1
1
3

左边输出结果的含义：第1、第7样本点为第3类，第2、第5、第6样本点为第1类，第3、第4样本点为第2类。

2.根据原始数据创建聚类

在MATLAB中，可使用clusterdata函数对原始数据进行聚类，格式有两种：

(1)clusterdata(x,a),其中0<a<1,表示在系统聚类树中距离小于a的样本点归结为一类；

(2)clusterdata(x,b),其中b>1且为整数，表示将原始数据x分为b类。

说明：利用clusterdata 函数对数据样本进行一次聚类的方法简洁方便，但使用范围较窄(不能由用户根据自身需要设定参数以及不能更改距离的计算方法)。

例3：由原始数据创建聚类。

```
%Filename:ex6_3.m
%根据原始数据创建聚类
x=[3 1.7;1 1;2 3;2 2.5;1.2 1;1.1 1.5;3 1];
t=clusterdata(x,0.5)    %距离小于0.5的样本点归结为一类
z=clusterdata(x,3)      %将原始数据x分为3类
```

t =

4
3
2
2
3
1
4

t的输出结果共有四类，第1类：样本点6；第2类：样本点3和4；第3类：样本点3和5；第4类：样本点1和7。

z =

2
3
1
1
3
3
2

z的结果约定将原始数据分为三类。Z的输出结果为，第1类：样本点3和4；第2类：样本点1和7；第3类：样本点2、5和6。

3.分步聚类法

设有样本数据矩阵x:

Step 1:对于不同的距离，利用**pdist**函数计算样本点之间的距离。

```
y1=pdist(x)   %欧氏距离
y2=pdist(x,'mahal')
y3=pdist(x,'minkowski',1)  %p=1为绝对距离/城市距离
```

Step 2:计算系统聚类树及相关信息。

```
z1=linkage(y1)
z2=linkage(y2)
z3=linkage(y3)
```

Step 3:利用cophenet函数计算聚类树信息与原始数据的距离之间的相关性，该值越大越好(利用cophenet函数评价聚类信息)。

t1=cophenet(z1,y1)

t2=cophenet(z2,y2)

t3=cophenet(z3,y3)

注意：z在前y在后，顺序不能颠倒。

Step 4:选择具有最大的cophenet值的距离进行聚类。

利用函数clusterdata(x,a)对数据进行聚类，其中0<a<1，表示系统聚类树距离小于a的样本点归为一类。

例4：分步聚类法示例。

```
%Filename:ex6_4.m
x=[3 1.7;1 1;2 3;2 2.5;1.2 1;1.1 1.5;3 1];
y1=pdist(x);                    %欧氏距离
y2=pdist(x,'mahal');
%y3=pdist(x,'minkowski',1);  %p=1为绝对距离/城市距离
y3=pdist(x,'cityblock');
z1=linkage(y1);
z2=linkage(y2);
z3=linkage(y3);
t1=cophenet(z1,y1)
t2=cophenet(z1,y2)
t3=cophenet(z1,y3)
```

```
t1 =
    0.9291
t2 =
    0.9103
t3 =
    0.9161
```

由于t1的值最大，可见此例利用欧氏距离最合适。于是，可编写另一个文件名为ex6_4a.m的文件。

```
%Filename:ex6_4a.m
x=[3 1.7;1 1;2 3;2 2.5;1.2 1;1.1 1.5;3 1];
y1=pdist(x);        %欧氏距离
z1=linkage(y1)
```

z1 =

2.0000	5.0000	0.2000
3.0000	4.0000	0.5000
6.0000	8.0000	0.5099
1.0000	7.0000	0.7000
9.0000	11.0000	1.2806
10.0000	12.0000	1.3454

矩阵**z1**的第**1**行表示样本点**2**、**5**为一类，在系统聚类树上的距离为了**0.2**，其他类推。考察矩阵的第**3**列，系统聚类树上的**6**个距离，可以选择**0.5**作为聚类分界阈值。于是，可编写另一个文件名为**ex6_4b.m**的文件。

```
%Filename:ex6_4b.m
x=[3 1.7;1 1;2 3;2 2.5;1.2 1;1.1 1.5;3 1];
y1=pdist(x);          %欧氏距离
z1=linkage(y1)
t=cluster(z1,0.5)
%h=dendrogram(z1)
```

输出结果：

```
t =
     4
     3
     2
     2
     3
     1
     4
```

即分为四类，第1类：样本点6；第2类样本点3和4；第3类：样本点2和5；第4类：样本点1和7。

思考题：根据调查得到某地10所学校的数据(见下表)，试采用分步聚类法编写程序，将这些学校按三种类别聚类。

学校	占地面积/m^2	建筑面积/m^2	教师总数	学生总数
学校1	2088	562.05	42	434
学校2	10344.8	4755	76	1279
学校3	2700	4100	56	820
学校4	3967	3751	67	990
学校5	5850.24	6173.25	78	1240
学校6	1803.26	5224.99	72	1180
学校7	2268	8011	56	800
学校8	32000	18000	200	2000
学校9	100000	30000	200	1100
学校10	173333	60000	420	2552

参考程序：

```
%Filename:tk6_1.m
%分步聚类法思考题
x=[2088,562.05,42,434
   10344.8,4755,76,1279
   2700,4100,56,820
   3967,3751,67,990
   5850.24,6173.25,78,1240
   1803.26,5224.99,72,1180
   2268,8011,56,800
   32000,18000,200,2000
   100000,30000,200,1100
   173333,60000,420,2552];
y1=pdist(x); %欧氏距离
y2=pdist(x,'mahal');
%y3=pdist(x,'minkowski',1);  %p=1为绝对距离/城市距离
y3=pdist(x,'cityblock');
```

```
z1=linkage(y1);
z2=linkage(y2);
z3=linkage(y3);
t1=cophenet(z1,y1)
t2=cophenet(z1,y2)
t3=cophenet(z1,y3)
```

t1 =

0.9409

t2 =

0.4736

t3 =

0.9364

t1最大，说明采用欧氏距离最好。

编写名为tk6_1a.m的另一个文件：

```
%Filename:tk6_1a.m
%分步聚类法思考题
x=[2088,562.05,42,434
   10344.8,4755,76,1279
   2700,4100,56,820
   3967,3751,67,990
   5850.24,6173.25,78,1240
   1803.26,5224.99,72,1180
   2268,8011,56,800
   32000,18000,200,2000
   100000,30000,200,1100
   173333,60000,420,2552];
```

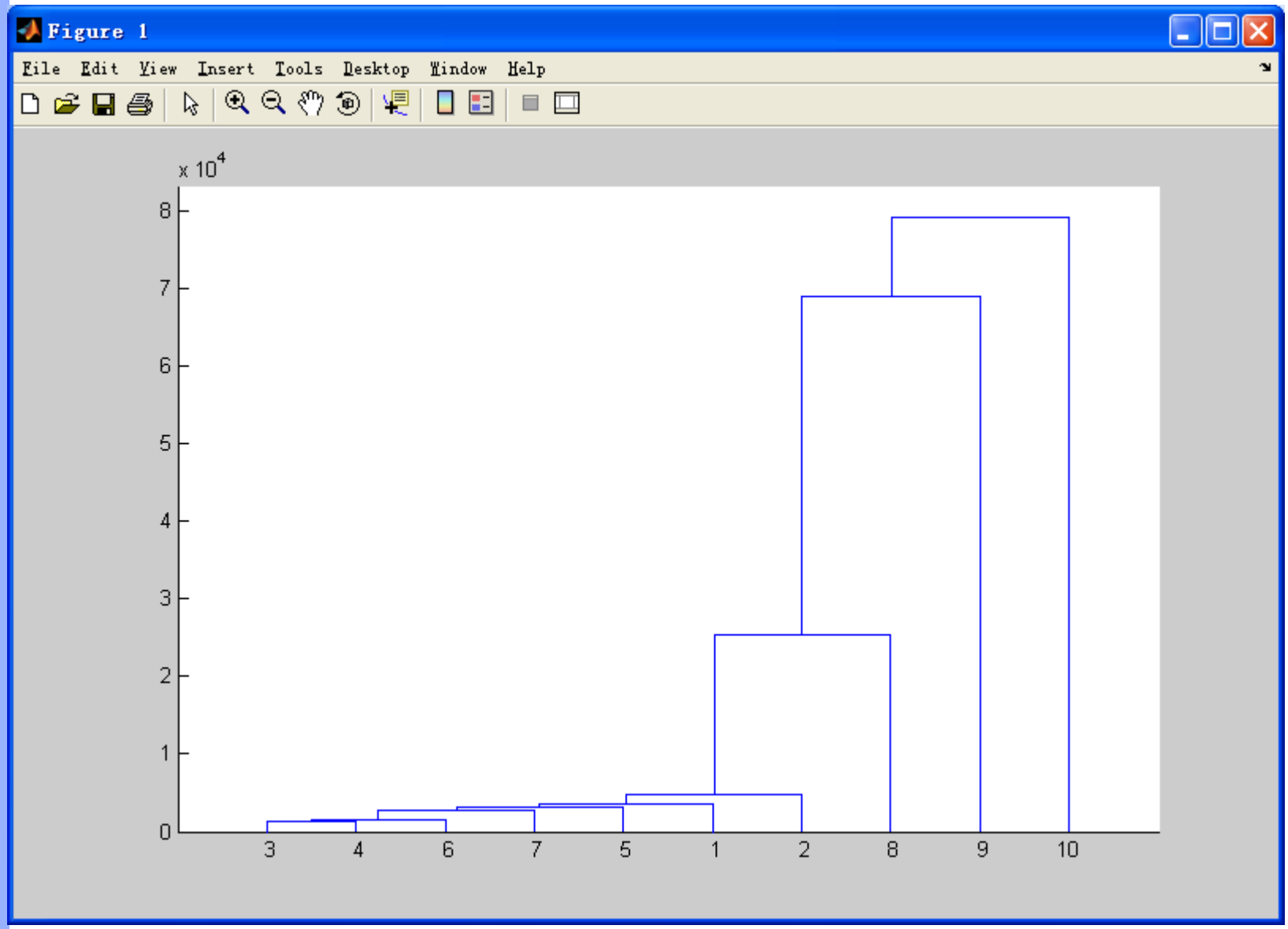
```
y1=pdist(x); %欧氏距离
z1=linkage(y1);
t=cluster(z1,3)
h=dendrogram(z1)
```

t =

2
2
2
2
2
2
2
2
1
3

t输出结果表明：学校9分为第1类；学校1-8分为第2类；学校10分为第3类。

思考题的系统聚类树状图：



举例2：运用分解聚类法对经济差距进行分类统计与决策(一篇小型论文)。

论文来源：
万树平. 运用分解聚类法对经济差距的分类统计与决策，统计与决策， 2007年第5期，P139。
(可在学校图书馆网站进“中国期刊网”或“重庆维普”检索“运用分解聚类法对经济差距的分类”，然后下载文件即可)

具体内容详见“运用分解聚类法对经济差距的分类.PDF”文档。