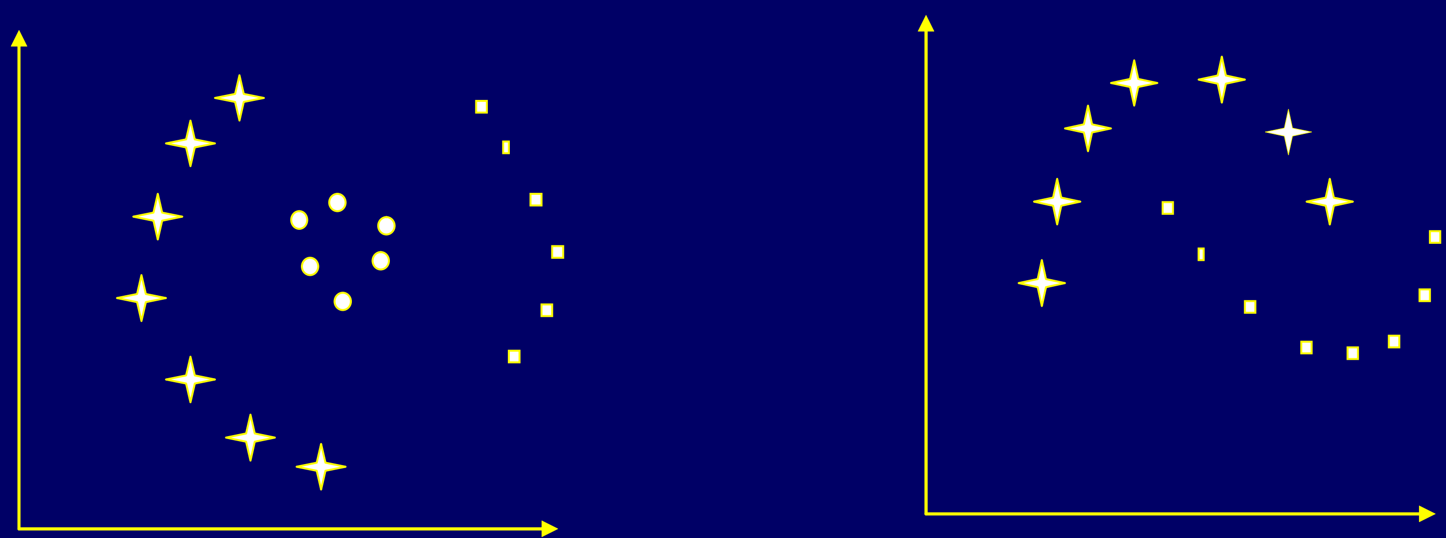
第二章　聚类分析

2.5.4 近邻函数法

在C—均值法中，当类内样本非球状散布时，用样本的均值矢量作为类的代表，一般聚类结果不佳。如果我们有类的模式分布的某些先验知识，可以构造能反映类的模式分布情况的核函数，那么就以核函数来代表类。

如果实际中不能确定核函数或不能用简单的函数表示核函数时，可以采用近邻函数法。这种算法特别适用于类的模式分布是条状或线状的情况。

2.5.4 近邻函数法



类的各种线状分布

2.5 聚类的算法

2.5.4 近邻函数法

近邻函数：

对于一个样本集中的任意两个样本 $\vec{x}_i$和 $\vec{x}_j$，如果$\vec{x}_i$是 $\vec{x}_j$ 的第Ⅰ个近邻点，则定义$\vec{x}_i$对 $\vec{x}_j$ 的近邻系数为Ⅰ，记为$d(i,j)=I$；

同理，如果 $\vec{x}_j$ 是$\vec{x}_i$的第J个近邻点，则定义$\vec{x}_j$对 $\vec{x}_i$ 的近邻系数为J，记为$d(j,i)=J$ 。

于是，$\vec{x}_i$ 和 $\vec{x}_j$ 之间的近邻函数值定义为:

$$\alpha_{ij} = d(i, j) + d(j, i) - 2 = I + J - 2$$

2·5 聚类的算法

近邻函数值：

$$\alpha_{ij} = d(i, j) + d(j, i) - 2 = I + J - 2$$

当 $\vec{x}_i$ 和 $\vec{x}_j$ 互为最近邻时，有 $\alpha_{ij} = 0$ 。

显然，样本间的近邻函数值越小，说明它们彼此越近，意味着它们越相似。

如果样本集包含N个样本，那么近邻系数总是小于或等于N-1，因此，$\vec{x}_i$ 和 $\vec{x}_j$ 之间的近邻函数值满足：

$$\alpha_{ij} \le 2N - 4$$

2·5 聚类的算法

连接损失：

在聚类过程中，如果 $\vec{x}_i$ 和 $\vec{x}_j$ 被聚为一类，就称 $\vec{x}_i$ 与 $\vec{x}_j$ 是相互连接的。

对于每个连接，都应定义一个指标，用以刻划这两个样本是否适于连接，称其为<u>连接损失。</u>

由两样本的近邻函数值 α_{ij} 的定义可知，α_{ij} 越小，表明它们越相似。若把它们连接起来，损失也就越小。因此可以<u>将近邻函数值 α_{ij} 作为 $\vec{x}_i$ 和 $\vec{x}_j$ 之间的连接损失</u>。

2.5.4 近邻函数法

连接损失：

在聚类过程中，当考虑样本 $\vec{x}_i$ 时，计算它与其它各样本间的近邻函数值，如果

$$\alpha_{ik} = \min_j \left[\alpha_{ij} \right]$$

则把 $\vec{x}_i$ 和 $\vec{x}_k$ 连接起来，并有连接损失 α_{ik}。

若 $\vec{x}_i$与 $\vec{x}_j$不实际连接，则不存在连接损失，即：

$$\alpha_{ij} \triangleq 0$$

近邻聚类准则函数：

在定义了两样本间的连接损失之后，还要区分出类内连接损失和类间连接损失。

设共有c类：$\omega_p(p=1,2,\cdots,c)$，总的类内连接损失定义为：

$$L_W=\sum_{p=1}^{c}\sum_{\substack{\vec{x}_i\in\omega_p\\ \vec{x}_j\in\omega_p}}\alpha_{ij}$$

记ω_p类的类内最大近邻函数值：

$$\alpha_{p\max}=\max_{\substack{\vec{x}_i\in\omega_p\\ \vec{x}_j\in\omega_p}}\left[\alpha_{ij}\right]$$

2·5 聚类的算法

设聚类ω_p和ω_q $(p, q = 1,2,\cdots,c; \quad p \neq q)$ 的样本之间的最小近邻函数值为γ_{pq}，即

$$\gamma_{pq} = \min_{\substack{\vec{x}_i \in \omega_p \\ \vec{x}_j \in \omega_q}} \left[\alpha_{ij}\right]$$

设γ_{pk} 为聚类ω_p与其它各聚类 ω_q $(q = 1,2,\cdots,c; \quad q \neq p)$ 的最小近邻函数值的最小值，即

$$\gamma_{pk} = \min_{\substack{q \\ q \neq p}} \left[\gamma_{pq}\right] \quad (p = 1,2,\cdots,c)$$

上式表明，除ω_p类内样本外，只有ω_k中的某一个样本与 ω_p中某一个样本最近邻，近邻函数值为 γ_{pk}。

2·5 聚类的算法

ω_p 类与 ω_k 类的类间最小连接损失有如下四种情况：

$$\beta_p = \begin{cases} (\alpha_{p\max} - \gamma_{pk}) + (\alpha_{k\max} - \gamma_{pk}) , & 若 \begin{cases} \gamma_{pk} > \alpha_{p\max} \\ \gamma_{pk} > \alpha_{k\max} \end{cases} \\ \alpha_{p\max} - \gamma_{pk} , & 若 \begin{cases} \gamma_{pk} \le \alpha_{p\max} \\ \gamma_{pk} > \alpha_{k\max} \end{cases} \\ \alpha_{k\max} - \gamma_{pk} , & 若 \begin{cases} \gamma_{pk} > \alpha_{p\max} \\ \gamma_{pk} \le \alpha_{k\max} \end{cases} \\ \alpha_{p\max} + \alpha_{k\max} - \gamma_{pk} , & 若 \begin{cases} \gamma_{pk} \le \alpha_{p\max} \\ \gamma_{pk} \le \alpha_{k\max} \end{cases} \end{cases}$$

2·5 聚类的算法

ω_p 类与 ω_k 类的类间最小连接损失有如下四种情况：

$$\beta_p = \begin{cases} (\alpha_{p\max} - \gamma_{pk}) + (\alpha_{k\max} - \gamma_{pk}), & 若 \begin{cases} \gamma_{pk} > \alpha_{p\max} \\ \gamma_{pk} > \alpha_{k\max} \end{cases} \\ \alpha_{p\max} + \alpha_{k\max} - \gamma_{pk}, & 若 \begin{cases} \gamma_{pk} \le \alpha_{p\max} \\ \gamma_{pk} \le \alpha_{k\max} \end{cases} \end{cases}$$

第一种情况：类内最大近邻函数值小于类间最小近邻值，说明聚类是合理的，这种情况不应付出代价，β_p可赋予负值，即损失为负数。

2·5 聚类的算法

ω_p 类与 ω_k 类的类间最小连接损失有如下四种情况：

$$\beta_p = \begin{cases} (\alpha_{p\max} - \gamma_{pk}) + (\alpha_{k\max} - \gamma_{pk}) \ , & 若 \begin{cases} \gamma_{pk} > \alpha_{p\max} \\ \gamma_{pk} > \alpha_{k\max} \end{cases} \\ \alpha_{p\max} - \gamma_{pk} \ , & 若 \begin{cases} \gamma_{pk} \le \alpha_{p\max} \\ \gamma_{pk} > \alpha_{k\max} \end{cases} \\ \alpha_{k\max} - \gamma_{p} & 若 \begin{cases} \gamma_{pk} > \alpha_{p\max} \end{cases} \\ \alpha \end{cases}$$

第二种情况：ω_p类的类内最大近邻函数值大于类间最小近邻值，说明这两类合并应付出代价，β_p赋予一个正值，即损失为：α_{pmax}-γ_{pk}。

2·5 聚类的算法

ω_p 类与 ω_k 类的类间最小连接损失有如下四种情况：

第三种情况：ω_k类的类内最大近邻函数值大于类间最小近邻值，说明这两类合并也应付出代价，β_p赋予一个正值而损失为：$\alpha_{kmax}-\gamma_{pk}$。

$$\beta_p = \begin{cases} \cdots \\ \alpha_{k\max} - \gamma_{pk}, & \text{若} \begin{cases} \gamma_{pk} > \alpha_{p\max} \\ \gamma_{pk} \le \alpha_{k\max} \end{cases} \\ \alpha_{p\max} + \alpha_{k\max} - \gamma_{pk}, & \text{若} \begin{cases} \gamma_{pk} \le \alpha_{p\max} \\ \gamma_{pk} \le \alpha_{k\max} \end{cases} \end{cases}$$

2·5 聚类的算法

ω_p 类与 ω_k 类的类间最小连接损失有如下四种情况：

第四种情况：ω_p和ω_k类的类内最大近邻函数值都大于类间最小近邻值，说明这两类合并应付出的代价更大，β_p赋予连接损失为：$a_{pmax}+\alpha_{kmax}-\gamma_{pk}$。

$$\beta_p = \begin{cases} \cdots \\ \alpha_{k\max} - \gamma_{pk} & 若 \begin{cases} \gamma_{pk} > \alpha_{p\max} \\ \gamma_{pk} \le \alpha_{k\max} \end{cases} \\ \alpha_{p\max} + \alpha_{k\max} - \gamma_{pk}, & 若 \begin{cases} \gamma_{pk} \le \alpha_{p\max} \\ \gamma_{pk} \le \alpha_{k\max} \end{cases} \end{cases}$$

2.5.4 近邻函数法

近邻聚类准则函数：

在上述描述基础上，定义总的类间损失：$L_B = \sum_{p=1}^{c} \beta_p$

聚类的目标是使各 γ_{pk} 尽可能地大，使各 $\alpha_{p\max}$ 尽可能地小，因而构造聚类的准则函数为：

$$J_L = L_W + L_B \Rightarrow \min$$

有了上述的准则函数后，可以用迭代方法得出近邻聚类的具体实现。

2.5 聚类的算法

近邻函数法算法步骤：

(1) 对于给定的待分类样本集 $X=\{\vec{x}_1,\vec{x}_2,\cdots,\vec{x}_N\}$，计算距离矩阵 D, D 的阵元： $D_{ij}=d(\vec{x}_i,\vec{x}_j)\quad(i,j=1,2,\cdots,N)$

$d(\vec{x}_i,\vec{x}_j)$ 表示样本 $\vec{x}_i$ 和 $\vec{x}_j$ 间的距离；

(2) 利用矩阵 D，计算近邻矩阵 M，其元素 M_{ij} 为样本 $\vec{x}_i$ 对 $\vec{x}_j$ 的近邻系数；

(3) 生成近邻函数矩阵 L，其阵元为 $L_{ij}=M_{ij}+M_{ji}-2$

置矩阵 L 的主对角线上阵元 $L_{ii}=2N$（$i=1, 2, \ldots, N$），

如果 $\vec{x}_i$ 和 $\vec{x}_j$ 有连接，则 L_{ij} 给出它们非零近邻函数值，即连接损失；

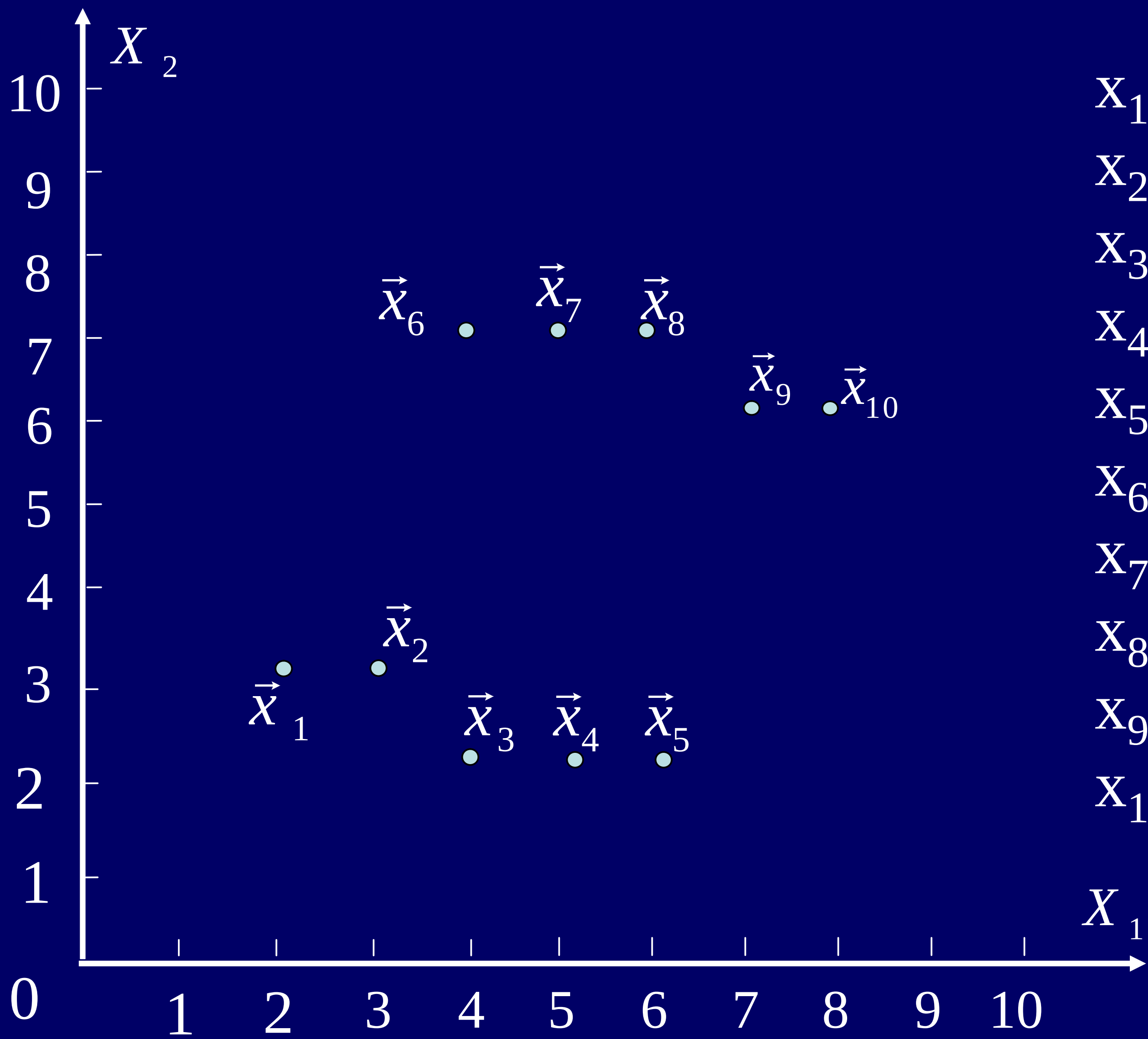
2.5 聚类的算法

(4) 搜索矩阵L，将每个点与和它有最小近邻函数值的点连接起来，形成初始聚类；

(5) 对于(4)所形成的聚类，计算γ_{pk}、α_{pm}、α_{km}。若γ_{pk}小于或等于α_{pm}或α_{km}，则合并ω_k和ω_p，它们的样本间建立连接关系，转至(5)；否则结束。

上述迭代过程，最终将使准则函数J_L达极小。

例：已知有10个样本，每个样本有2个特征，使用近邻函数法实现样本分类。



$x_1(2,3)$
$x_2(3,3)$
$x_3(4,2)$
$x_4(5,2)$
$x_5(6,2)$
$x_6(4,7)$
$x_7(5,7)$
$x_8(6,7)$
$x_9(7,6)$
$x_{10}(8,6)$

2·5 聚类的算法

(1) 计算距离矩阵*D*

	X1	x2	x3	x4	x5	x6	x7	x8	x9	x10
x1	0	1.0	2.2	3.2	4.1	4.5	5.0	5.7	5.8	6.7
x2	1.0	0	1.4	2.2	3.2	4.1	4.4	5.0	5.0	5.8
x3	2.2	1.4	0	1.0	2.0	5.0	5.1	5.4	5.0	5.7
x4	3.2	2.2	1.0	0	1.0	5.1	5.0	5.1	4.4	5.0
x5	4.1	3.2	2.0	1.0	0	5.4	5.1	5.0	4.1	4.5
x6	4.5	4.1	5.0	5.1	5.4	0	1.0	2.0	3.2	4.1
x7	5.0	4.4	5.1	5.0	5.1	1.0	0	1.0	2.2	3.2
x8	5.7	5.0	5.4	5.1	5.0	2.0	1.0	0	1.4	2.2
x9	5.8	5.0	5.0	4.4	4.1	3.2	2.2	1.4	0	1.0
x10	6.7	5.8	5.7	5.0	4.5	4.1	3.2	2.2	1.0	0

2·5 聚类的算法

(1) 计算近邻系数矩阵*M*

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
x1	0	1	2	3	4	5	6	7	8	9
x2	1	0	2	3	4	5	6	7	7	8
x3	4	2	0	1	3	5	6	7	5	8
x4	3	2	1	0	1	6	5	6	4	5
x5	4	3	2	1	0	8	7	6	4	5
x6	5	4	6	7	8	0	1	2	3	4
x7	5	4	6	5	6	1	0	1	2	3
x8	8	5	7	6	5	3	1	0	2	4
x9	8	7	7	6	5	4	3	2	0	1
x10	9	8	7	6	5	4	3	2	1	0

2·5 聚类的算法

(1) 计算近邻函数矩阵L

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
x1	20	0	4	4	6	8	9	13	14	16
x2	0	20	2	3	5	7	8	10	12	14
x3	4	2	20	0	3	9	10	12	10	13
x4	4	3	0	20	0	11	8	10	8	9
x5	6	5	3	0	20	14	11	9	7	8
x6	8	7	9	11	14	20	0	3	5	6
x7	9	8	10	8	11	0	20	0	3	4
x8	13	10	12	10	9	3	0	20	2	4
x9	14	12	10	8	7	5	3	2	20	0
x10	16	14	13	9	8	6	4	4	0	20

2·5 聚类的算法

形成初始聚类

1： x1,x2

2： x3,x4,x5

3： x6,x7,x8

4： x9,x10

计算$\alpha_{p\max}$ 和 γ_{pk}

$\alpha_{1\max}=0$

$\alpha_{2\max}=3$

$\alpha_{3\max}=3$

$\alpha_{4\max}=0$

2.5 聚类的算法

(1) 计算近邻函数矩阵L

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
x1	20	0	4	4	6	8	9	13	14	16
x2	0	20	2	3	5	7	8	10	12	14
x3	4	2	20	0	3	9	10	12	10	13
x4	4	3	0	20	0	11	8	10	8	9
x5	6	5	3	0	20	14	11	9	7	8
x6	8	7	9	11	14	20	0	3	5	6
x7	9	8	10	8	11	0	20	0	3	4
x8	13	10	12	10	9	3	0	20	2	4
x9	14	12	10	8	7	5	3	2	20	0
x10	16	14	13	9	8	6	4	4	0	20

2·5 聚类的算法

形成初始聚类	计算 $\alpha_{p\max}$ 和 γ_{pk}			
1：x1,x2	$\alpha_{1\max}=0$	$\gamma_{12}=2$		
2：x3,x4,x5	$\alpha_{2\max}=3$	$\gamma_{13}=7$	$\gamma_{23}=8$	
3：x6,x7,x8	$\alpha_{3\max}=3$	$\gamma_{14}=12$	$\gamma_{24}=7$	$\gamma_{34}=2$
4：x9,x10	$\alpha_{4\max}=0$			

2·5 聚类的算法

形成初始聚类	计算 $\alpha_{p\max}$ 和 γ_{pk}
1：x1,x2 2：x3,x4,x5 3：x6,x7,x8 4：x9,x10	$\alpha_{1\max}=0$　$\gamma_{12}=2$ $\alpha_{2\max}=3$　$\gamma_{13}=7$　$\gamma_{23}=8$ $\alpha_{3\max}=3$　$\gamma_{14}=12$　$\gamma_{24}=7$　$\gamma_{34}=2$ $\alpha_{4\max}=0$
合并1，2 1：x1,x2,x3,x4,x5 2：x6,x7,x8 3：x9,x10	

2·5 聚类的算法

(1) 计算近邻函数矩阵*L*

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
x1	20	0	4	4	6	8	9	13	14	16
x2	0	20	2	3	5	7	8	10	12	14
x3	4	2	20	0	3	9	10	12	10	13
x4	4	3	0	20	0	11	8	10	8	9
x5	6	5	3	0	20	14	11	9	7	8
x6	8	7	9	11	14	20	0	3	5	6
x7	9	8	10	8	11	0	20	0	3	4
x8	13	10	12	10	9	3	0	20	2	4
x9	14	12	10	8	7	5	3	2	20	0
x10	16	14	13	9	8	6	4	4	0	20

2.5 聚类的算法

形成初始聚类

1： x1,x2

2： x3,x4,x5

3： x6,x7,x8

4： x9,x10

计算 $\alpha_{p\max}$ 和 γ_{pk}

$\alpha_{1\max}=0$ $\gamma_{12}=2$

$\alpha_{2\max}=3$ $\gamma_{13}=7$ $\gamma_{23}=8$

$\alpha_{3\max}=3$ $\gamma_{14}=12$ $\gamma_{24}=7$ $\gamma_{34}=2$

$\alpha_{4\max}=0$

合并1，2

1： x1,x2,x3,x4,x5

2： x6,x7,x8

3： x9,x10

$\alpha_{1\max}=6$ $\gamma_{12}=7$

$\alpha_{2\max}=3$ $\gamma_{13}=7$ $\gamma_{23}=2$

$\alpha_{3\max}=0$

2·5 聚类的算法

1： x1,x2,x3,x4,x5

2： x6,x7,x8

3： x9,x10

$\alpha_{1\max}=6$ $\gamma_{12}=7$

$\alpha_{2\max}=3$ $\gamma_{13}=7$ $\gamma_{23}=2$

$\alpha_{3\max}=0$

合并2，3

1： x1,x2,x3,x4,x5

2： x6,x7,x8,x9,x10

计算$\alpha_{p\max}$ 和 γ_{pk}

$\alpha_{1\max}=6$ $\gamma_{12}=7$

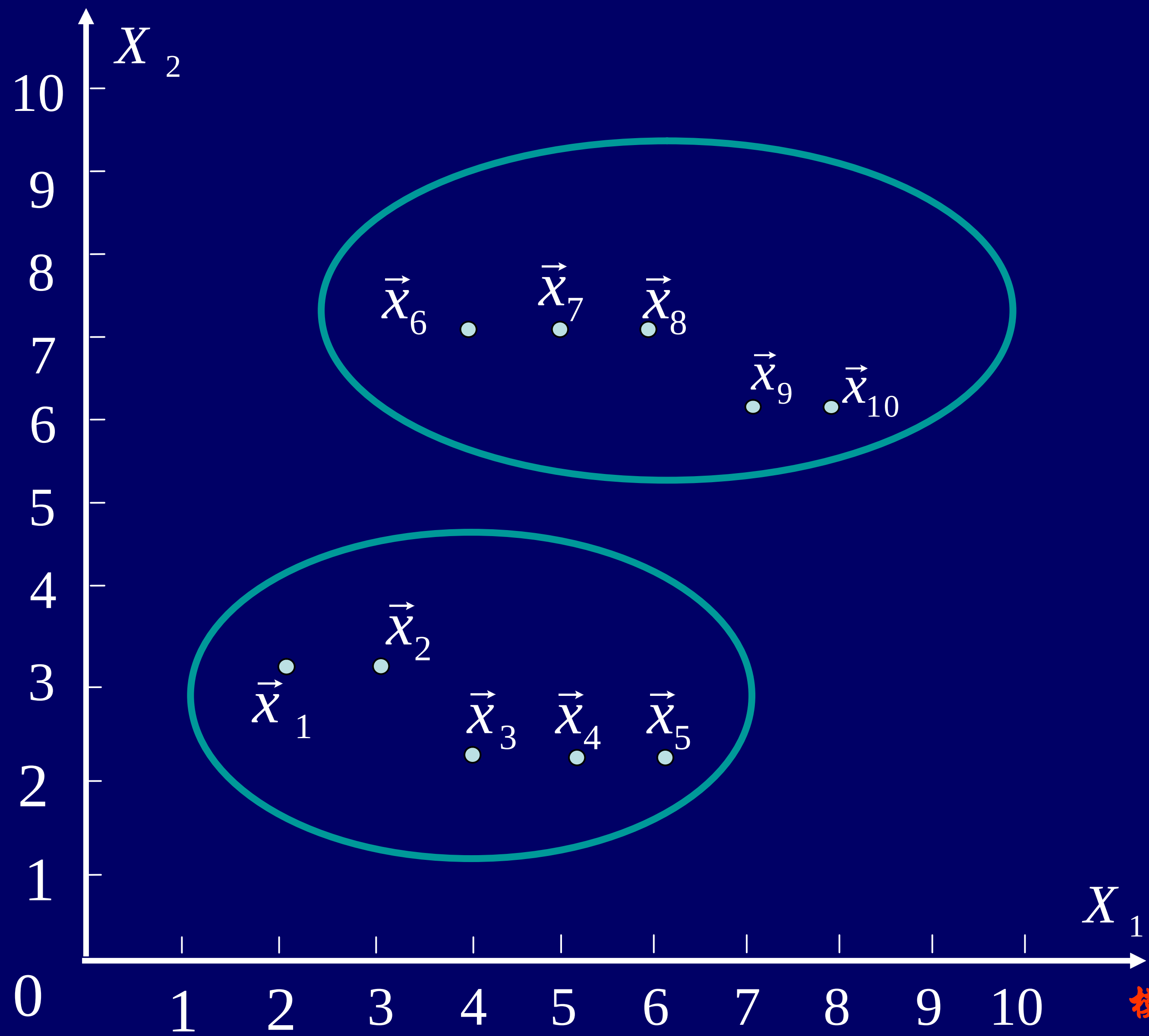
$\alpha_{2\max}=6$

2·5 聚类的算法

(1) 计算近邻函数矩阵L

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
x1	20	0	4	4	6	8	9	13	14	16
x2	0	20	2	3	5	7	8	10	12	14
x3	4	2	20	0	3	9	10	12	10	13
x4	4	3	0	20	0	11	8	10	8	9
x5	6	5	3	0	20	14	11	9	7	8
x6	8	7	9	11	14	20	0	3	5	6
x7	9	8	10	8	11	0	20	0	3	4
x8	13	10	12	10	9	3	0	20	2	4
x9	14	12	10	8	7	5	3	2	20	0
x10	16	14	13	9	8	6	4	4	0	20

2·5 聚类的算法



X_2
10
9
8
7
6
5
4
3
2
1
0
$\vec{x}_6$
$\vec{x}_7$
$\vec{x}_8$
$\vec{x}_9$
$\vec{x}_{10}$
$\vec{x}_1$
$\vec{x}_2$
$\vec{x}_3$
$\vec{x}_4$
$\vec{x}_5$
X_1
1
2
3
4
5
6
7
8
9
10

2.5 聚类的算法

形成初始聚类

1： x1,x2

2： x3,x4,x5

3： x6,x7,x8

4： x9,x10

计算 $\alpha_{p\max}$ 和 γ_{pk}

$\alpha_{1\max}=0$ $\gamma_{12}=2$

$\alpha_{2\max}=3$ $\gamma_{13}=$

$\alpha_{3\max}=3$

$\alpha_{4\max}=0$

合并1，2

1： x1,x2,x3,x4,x5

2： x6,x7,x8

3： x9,x10

$\alpha_{1\max}=3$ $\gamma_{12}=$

$\alpha_{2\max}=3$ $\gamma_{13}=$

$\alpha_{3\max}=0$

对于链状的模式分布时，合并后的类内最大连接损失不应重新计算，而是取$\alpha_{1\max}$,$\alpha_{2\max}$,γ_{12}的最大值。

聚类算法小结：

- 基于相似阈值和最小距离原则的简单聚类方法
- 最大最小距离方法
- 谱系聚类方法
- C均值聚类方法
- 近邻函数聚类方法

谢 谢！