第5章
数据聚类
Data Clustering

向 世 明
smxiang@nlpr.ia.ac.cn
http://www.escience.cn/people/smxiang/index.html

时空数据分析与学习课题组(STDAL)
中科院自动化研究所 模式识别国家重点实验室

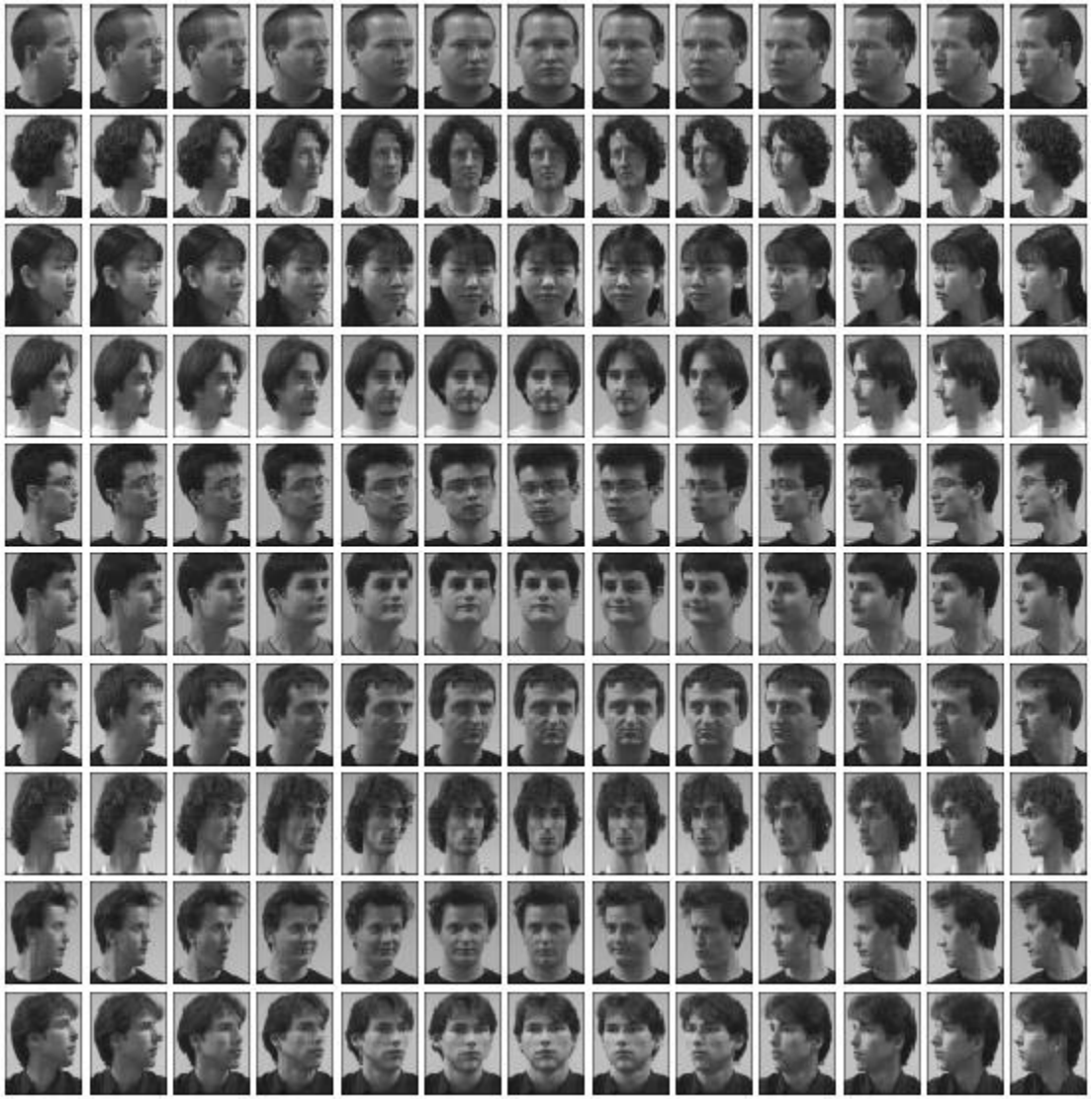
助教： 方深 (shen.fang@nlpr.ia.ac.cn)

内容提要

- 引言
- 聚类性能评价
- 一些挑战性问题
- 距离与相似性度量
- 层次聚类
- K-均值聚类
- 谱聚类

5.1 引言

- **聚类**
  - **物以类聚，人以群分。**
  - **将数据分成多个类别，在同一个类内，对象（实体）之间具有较高的相似性，不同类对象间差异性较大。**
  - 对一批没有类别标签的样本集，按照样本之间的相似程度分类，相似的归为一类，不相似的归为其它类。这种分类称为**聚类分析**，也称为无监督分类。
  - 聚类的质量(或结果)取决于对度量标准的选择。
  - 聚类结果**因不同任务而不同**。



身份识别 vs 姿态估计

5.1 引言

- **聚类任务**
  - 给定一个样本集合 X，给定一种度量样本间相似度或者相异度（距离）的标准。聚类系统的输出是关于样本集 X 的一个划分，即 $D = \{D_1 \cup D_2 \cup \ldots \cup D_k\}$。其中，$D_i$ $(i=1,2,\ldots,k)$是 X 的一个子集，且满足：
    - $D_1 \cup D_2 \cup \ldots \cup D_k = X$
    - $D_i \cap D_j = \varnothing, \quad i \neq j$
  - D 中成员 D_1, D_2 ,…, D_k 叫做类或者簇(cluster)，每个类均通过一些特征来描述：
    - 通过**类中心**或者**类的边界点**来表示；
    - 使用**聚类树**采用图形化方式来表示。

5.1 引言

- **聚类方法分类**
  - **按照不同的技术路线**
    - **模型法(原型聚类）：**为每一个簇引入一个模型，然后对数据进行划分，使其满足各自分派的模型，如K-Means。
    - **层次法：**对给定样本进行层次划分，如**层级聚类**。
    - **密度法：**对数据的密度进行评价，如**混合高斯模型、Mean-Shift方法**。
    - **网格法：**将数据空间划分为有限个单元网络结构，然后基于网络结构进行聚类，如**矢量量化**。

5.1 引言

- **挑战性问题**
  - **可伸缩性**
    - 可伸缩性是指聚类算法无论对于小数据集还是大数据集，都应有效；无论对小类别数据还是大别类数据，都应有效。
  - **具有不同类型的数据处理能力**
    - 既可处理数值型数据，也可处理非数植型数据；既可处理离散数据，也可处理连续域内的数据。比如布尔型、时序型、枚举型、以及这些类型的混合。
  - **能够发现任意形状的聚类**
    - 能够发现任意形状的簇，球状的、位于同一流形上的数据。因此，选择合适的距离度量很关键。

5.2 聚类性能评价

- 有聚类结果参考标准：外部指标（external index）
- 无聚类结果参考标准：内部指标（internal index）

5.2 聚类性能评价

- **外部指标**
  - 假定存在某个参考聚类结果，通过与它比较计算得到外部指标
  - 数据集：$D=\{x_1,x_2,\cdots,x_m\}$
  - 通过聚类给出的簇划分：$C=\{C_1,C_2,\cdots,C_k\}$
  - 参考聚类结果为：$C^*=\{C_1^*,C_2^*,\cdots,C_s^*\}$

a：在C中和C^*中**都属于相同**聚类的样本对个数

b：在C中**属于相同**聚类、但在C^*中**属于不同**聚类的样本对个数

c：在C中**属于不同**聚类、但在C^*中**属于相同**聚类的样本对个数

d：在C中和C^*中**都属于不同**聚类的样本对个数

5.2 聚类性能评价

- **外部指标**

  a：在C中和C^*中**都属于相同**聚类的样本对个数

  b：在C中**属于相同**聚类、但在C^*中**属于不同**聚类的样本对个数

  c：在C中**属于不同**聚类、但在C^*中**属于相同**聚类的样本对个数

  d：在C中和C^*中**都属于不同**聚类的样本对个数

$JC = \frac{a}{a+b+c}$	Jaccard系数（Jaccard Coefficient）
$FMI = \sqrt{\frac{a}{a+b} \cdot \frac{a}{a+c}}$	FM指数（Fowlkes and Mallows Index）
$RI = \frac{2(a+d)}{m(m-1)}$	Rand指数（Rand Index）

5.2 聚类性能评价

- **内部指标**
  - 在无参考聚类结果下，根据**聚类簇样本之间的相互距离**定义一些内部评价指标

$avg(C) = \frac{1}{\lvert C\rvert\lvert C-1\rvert} \sum_{x_i, x_j \in C, x_i \neq x_j} dist(x_i, x_j)$	簇内平均距离
$diam(C) = \max_{x_i, x_j \in C, x_i \neq x_j} dist(x_i, x_j)$	簇内最远距离
$d_{\min}(C_i, C_j) = \min_{x_i \in C_i, x_j \in C_j} dist(x_i, x_j)$	簇间最近距离
$d_{cen}(C_i, C_j) = dist(u_i, u_j), \quad u_i = \frac{1}{\lvert C_i\rvert} \sum_{x_j \in C_i} x_j$	簇间中心距离

5.2 聚类性能评价

- **内部指标**

1、DB指数（Davies-Bouldin Index，DBI）

$$\mathrm{DBI}=\frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\left(\frac{avg(C_i)+avg(C_j)}{d_{cen}(C_i,C_j)}\right)$$

2、Dunn指数（Dunn Index，DI）

$$\mathrm{DI}=\min_{1\le i\le k}\left\{\min_{j\neq i}\left(\frac{d_{\min}(C_i,C_j)}{\max_{1\le l\le k}diam(C_l)}\right)\right\}$$

5.2 聚类性能评价

- **常用指标（标签给定情形）**

1、聚类精度(Accuracy)：

$$\mathrm{Acc}= = \frac{1}{n}\sum_{i=1}^{n}\delta\left(c_i, \mathrm{map}(g_i)\right)$$

其中，c_i 和g_i为第i个样本的聚类标签和真实标签，map()为一个置换函数，样本标签置换后具有最大的匹配程度，$\delta(\cdot,\cdot)$为通常的指标函数。

5.2 聚类性能评价

- **常用指标（标签给定情形）**

2、Normalized Mutual Information (NMI)

$$\mathrm{NMI}=\frac{\sum_{i,j=1}^{c} o_{ij}\log\left((n\times o_{ij})/(c_i\times g_j)\right)}{\sqrt{\sum_{i=1}^{c} c_i\log\left((c_i/n)\right)\cdot\sum_{j=1}^{c} g_j\log\left((g_j/n)\right)}}$$

where c_i is the number of the data points in the i-th cluster C_i obtained by the algorithm, g_j is that of the data points in the j-th ground-truth class G_j , and o_{ij} indicates the number of the data points in $C_i \cap G_j$. In general, the larger the accuracy and NMI are, the better the performance of the algorithm is.

5.3 一些挑战性问题

- **挑战性问题**
  - **能够处理高维数据**
    - 既可处理属性较少的数据，也可处理属性较多的数据。
    - 在高维空间聚类更具挑战性，随着维数的增加，具有相同距离的两个样本其相似程度可以相差很远。对于高维稀疏数据，这一点更突出。
  - **对噪声鲁棒**
    - 在实际中，绝大多数样本集都包含噪声、空缺、部分未知属性、孤立点、甚至错误数据。

5.3 一些挑战性问题

- **挑战性问题**
  - **具有约束的聚类**
    - 在实际应用中，通常需要在某种约束条件下进行聚类，既满足约束条件，以希望有高聚类精度，是一个挑战性问题。
  - **对初始输入参数鲁棒**
    - 具有自适应的簇数判定能力（一直没有解决好）。
    - 对初始聚类中心鲁棒。
  - **能够解决用户的问题**
    - 聚类结果能被用户所理解，并能带来经济效益，特别是在数据挖掘领域。

5.4 距离与相似性度量

- **距离**
  - 设有 d 维空间的三个样本**x**，**y** 和 **z**，记 $d(.,.)$为一个 $R^d \times R^d \rightarrow R$的映射，如满足如下几个条件则称$d(.,.)$为一个距离:
    - $d(\mathbf{x}, \mathbf{y}) \geq 0$ 非负性
    - $d(\mathbf{x}, \mathbf{x}) = 0$ 自相似性
    - $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ 对称性
    - $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y})$ 三角不等式
  - 距离可以描述对点间的相异程度，距离越大，两个点越不相似；距离越小，两个点越相似。

5.4 距离与相似性度量

- 设$\mathbf{x}, \mathbf{y} \in \mathrm{R}^d$，**Minkowski 距离度量**定义如下：

$$d(\mathbf{x},\mathbf{y}) = \left(\sum_{i=1}^{d} |x_i - y_i|^q \right)^{\frac{1}{q}}$$

$$d(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{d} |x_i - y_i|$$

城区距离

曼哈顿距离

$$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{d} |x_i - y_i|^2}$$

欧氏距离

$$d(\mathbf{x},\mathbf{y}) = \max_{1 \le i \le d} |x_i - y_i|$$

切比雪夫距离

5.4 距离与相似性度量

- 设$\mathbf{x}, \mathbf{y} \in \mathrm{R}^d$， **Mahalanobis (马氏)距离**定义如下：

$$d(\mathbf{x},\mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^T \mathbf{M}(\mathbf{x}-\mathbf{y})}$$

其中，**M**是半正定矩阵。

  - **M**为单位矩阵时，退化为欧氏距离度量。
  - **M**为对角矩阵时，退化为特征加权欧氏距离

5.4 距离与相似性度量

- **相似性**
  - 设$\mathbf{x}, \mathbf{y} \in \mathrm{R}^d$， **余弦相似度**定义如下：

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{d} x_i y_i}{\sqrt{\sum_{i=1}^{d} x_i^2} \sqrt{\sum_{i=1}^{d} y_i^2}}$$

（两个模为1的向量之内积）

5.4 距离与相似性度量

- **相似性**
  - 设**x**，**y** $\in R^d$，其每维特征只取{0,1}中的一个值。为了定义数据点之间的距离，通常先计算出如下几个值：

    - f_{00}：样本 **x** 和 **y** 中满足 $x_i=y_i=0$ 的属性的个数
    - f_{10}：样本 **x** 和 **y** 中满足 $x_i=1\&y_i=0$ 的属性的个数
    - f_{01}：样本 **x** 和 **y** 中满足 $x_i=0\&y_i=1$ 的属性的个数
    - f_{11}：样本 **x** 和 **y** 中满足 $x_i=y_i=1$ 的属性的个数

  - 进一步，可定义如下几种类型的相似性度量：

5.4 距离与相似性度量

- 相似性
  - **简单匹配系数**(simple matching coefficient, SMC):

$$s_{SMC}(\mathbf{x},\mathbf{y}) = \frac{f_{00}+f_{11}}{f_{00}+f_{10}+f_{01}+f_{11}}$$

  - **Jaccard 相似系数**:

$$s_J(\mathbf{x},\mathbf{y}) = \frac{f_{11}}{f_{10}+f_{01}+f_{11}}$$

  - **Tanimoto系数**:

$$s_T(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}^T\mathbf{y}}{\mathbf{x}^T\mathbf{x}+\mathbf{y}^T\mathbf{y}-\mathbf{x}^T\mathbf{y}} = \frac{f_{11}}{\mathbf{x}^T\mathbf{x}+\mathbf{y}^T\mathbf{y}-f_{11}}$$

$\mathbf{x}^T\mathbf{x}$: $\mathbf{x}$取1的个数　　$\mathbf{y}^T\mathbf{y}$: $\mathbf{y}$取1的个数

5.4 距离与相似性度量

- **类间距离：**
  - **最短距离法**：定义两个类中最近的两个样本的距离为类间距离。

$$d(D_a, D_b) = \min\{d(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \in D_a, \mathbf{y} \in D_b\}$$

  - **最长距离法**：定义两个类中最远的两个样本的距离为类间距离。

$$d(D_a, D_b) = \max\{d(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \in D_a, \mathbf{y} \in D_b\}$$

- **类直径：**类直径反映类中样本之间的差异，可定义为类中各样本至**类中心点**的欧氏距离平方和：

$$r(D_a) = \sum_{\mathbf{x} \in D_a} (\mathbf{x} - \bar{\mathbf{x}})^T (\mathbf{x} - \bar{\mathbf{x}}))$$

5.5 层次聚类(Hierarchical Clustering)

- **生物学上的物种分类**
  - 界、门、纲、目、科、属、种
  - 最相似的物种被分为“种”
  - 相似的“种”被分为“属”
  - ……

这种思想是一种典型的聚类分析思想，称为**层次聚类**，也叫**分级聚类**或者**系统聚类**。

5.5 层次聚类

- 对于 n 个样本，极端的情况下，最多可以将数据分成 n类；最少可以只分成一类，即全部样本都归为一类。
- **凝聚的层次聚类（自底向上）**

  将每个样本作为一个簇，然后根据给定的规则**逐渐合并**一些样本，形成更大的簇，直到所有的样本都被分到一个簇中。
- **分裂的层次聚类（自顶向下）**

  将所有的样本置于一个簇中，然后根据给定的规则**逐渐细分**样本，得到越来越小的簇，直到某个终止条件得到满足。

5.5 层次聚类

- **自底向上的层次聚类步骤**

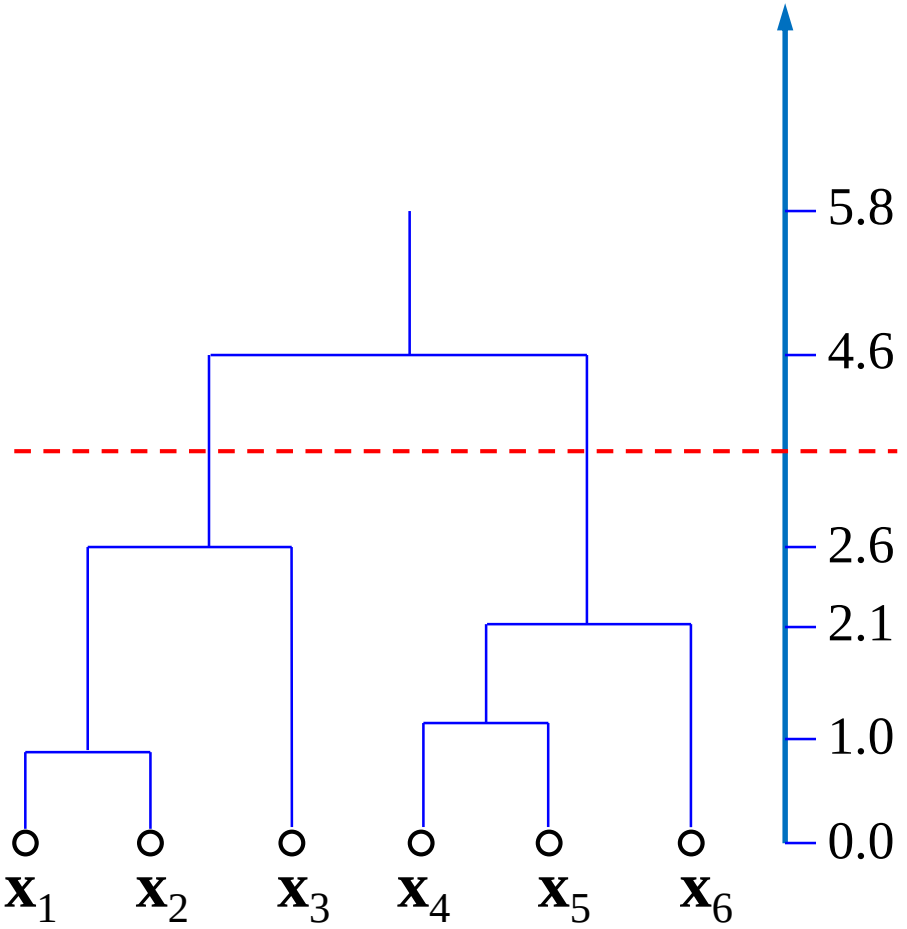
  (1) 初始化：每个样本形成一个类

  (2) 合并：计算任意两个类之间的距离（或相似性），将距离最小（或相似性最大）的两个类合并为一个类，记录下这两个类之间的距离（或相似性），其余类不变。

  (3) 重复步骤 (2)，直到所有样本被合并到一个类之中。

5.5 层次聚类

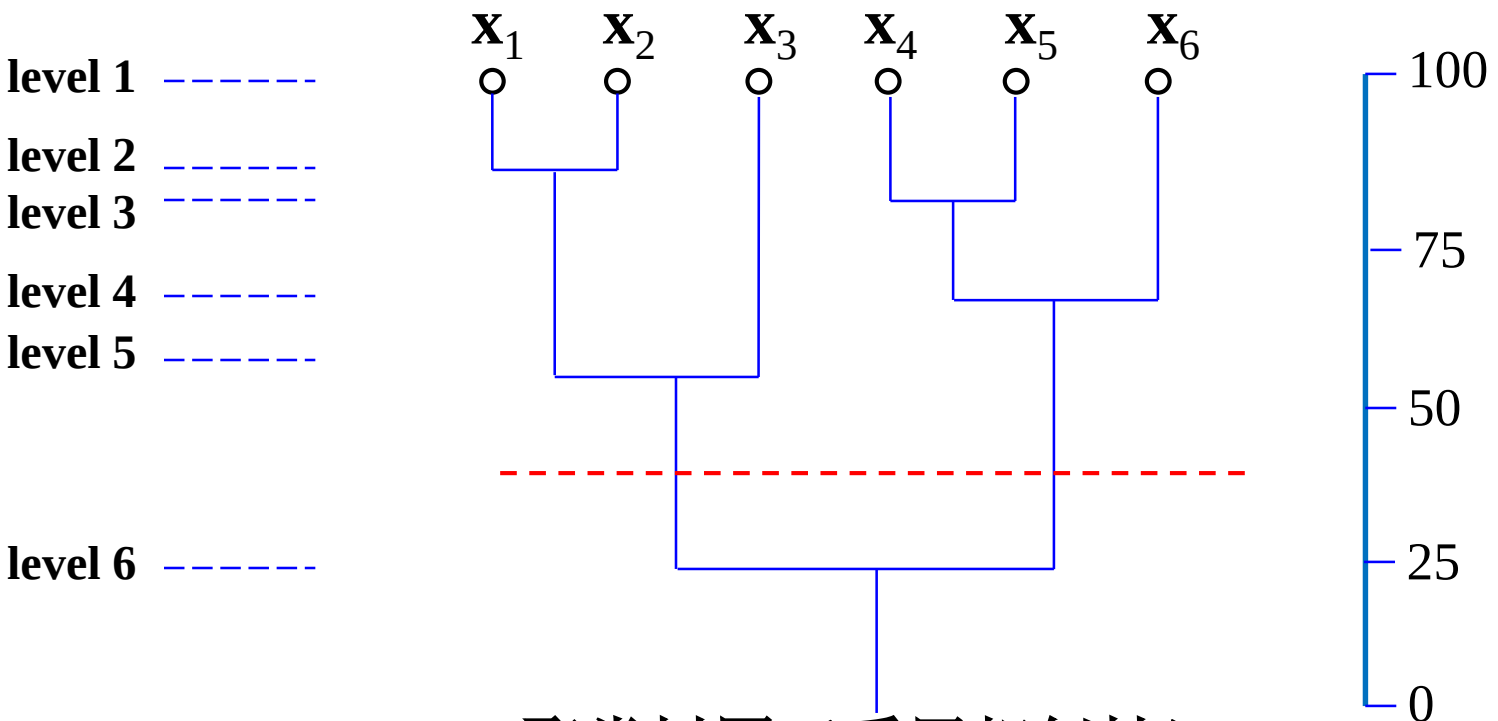
- **聚类树**
  - 层次聚类结果用一棵树来表示，称为**聚类树**(dendrogram)，或**系统树图**。
  - 如右图，最底层的每个节点表示一个样本，**采用树枝连接两个合并的样本**，树枝的长度反映两个节点之间的距离（或相似性）。



聚类树图（采用距离）

5.5 层次聚类

- **聚类树另一种表示：采用“水平”来表示分级聚类过程的不同阶段。**开始为水平1，每个样本为一类，然后每执行一次“合并”增加一个水平。



聚类树图（采用相似性）

5.5 层次聚类

(1) 初始化：每个样本形成一个类
(2) 合并：**计算任意两个类之间的距离**（或相似性），将距离最小（或相似性最大）的两个类合并为一个类，记录下这两个类之间的距离（或相似性），其余类不变。
(3) 重复步骤（2），直到所有样本被合并到一个类之中。

- **两个核心问题：**
  - 如何度量**样本之间**的距离或相似性
  - 如何度量**两个类（簇）之间**的距离或者相似性

5.6 K-均值聚类

- **算法基本思想**

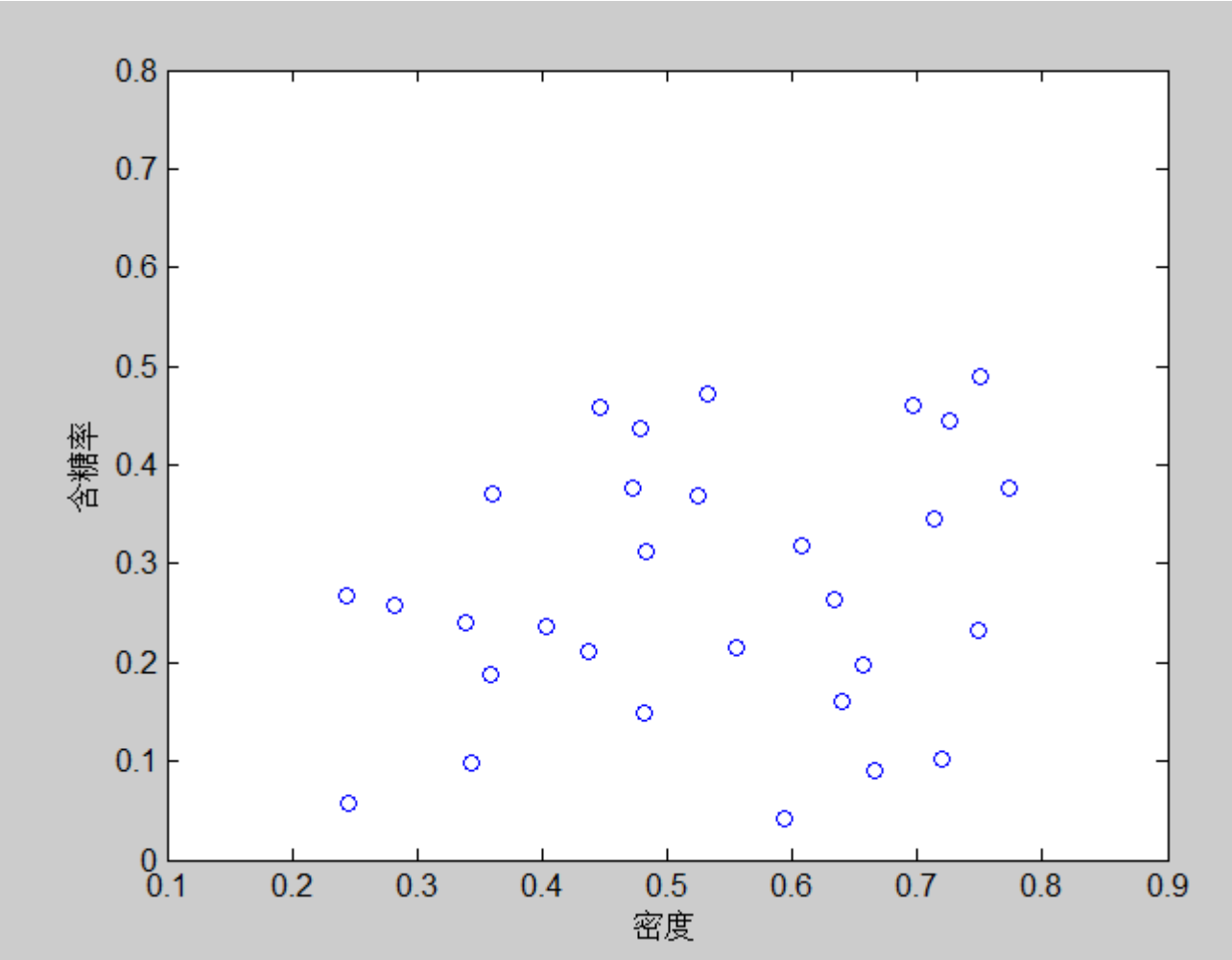
K-Means Clustering—Algorithm 1

1 begin initialization n, c, μ_1, μ_2,…, μ_c.
2 do classify n samples according to nearest μ_i
3 re-compute μ_i
4 until no change in μ_i
5 return μ_1, μ_2,…, μ_c

5.6 K-均值聚类

任务：对30个西瓜根据其密度和含糖率进行聚类。

编号	密度	含糖率	编号	密度	含糖率	编号	密度	含糖率
1	0.697	0.460	11	0.245	0.057	21	0.748	0.232
2	0.774	0.376	12	0.343	0.099	22	0.714	0.346
3	0.634	0.264	13	0.639	0.161	23	0.483	0.312
4	0.608	0.318	14	0.657	0.198	24	0.478	0.437
5	0.556	0.215	15	0.360	0.370	25	0.525	0.369
6	0.403	0.237	16	0.593	0.042	26	0.751	0.489
7	0.481	0.149	17	0.719	0.103	27	0.532	0.472
8	0.437	0.211	18	0.359	0.188	28	0.473	0.376
9	0.666	0.091	19	0.339	0.241	29	0.725	0.445
10	0.243	0.267	20	0.282	0.257	30	0.446	0.459


0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
含糖率
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
密度

5.6 K-均值聚类

假定聚类簇数为3

初始化：随机选取3个样本作为均值向量。

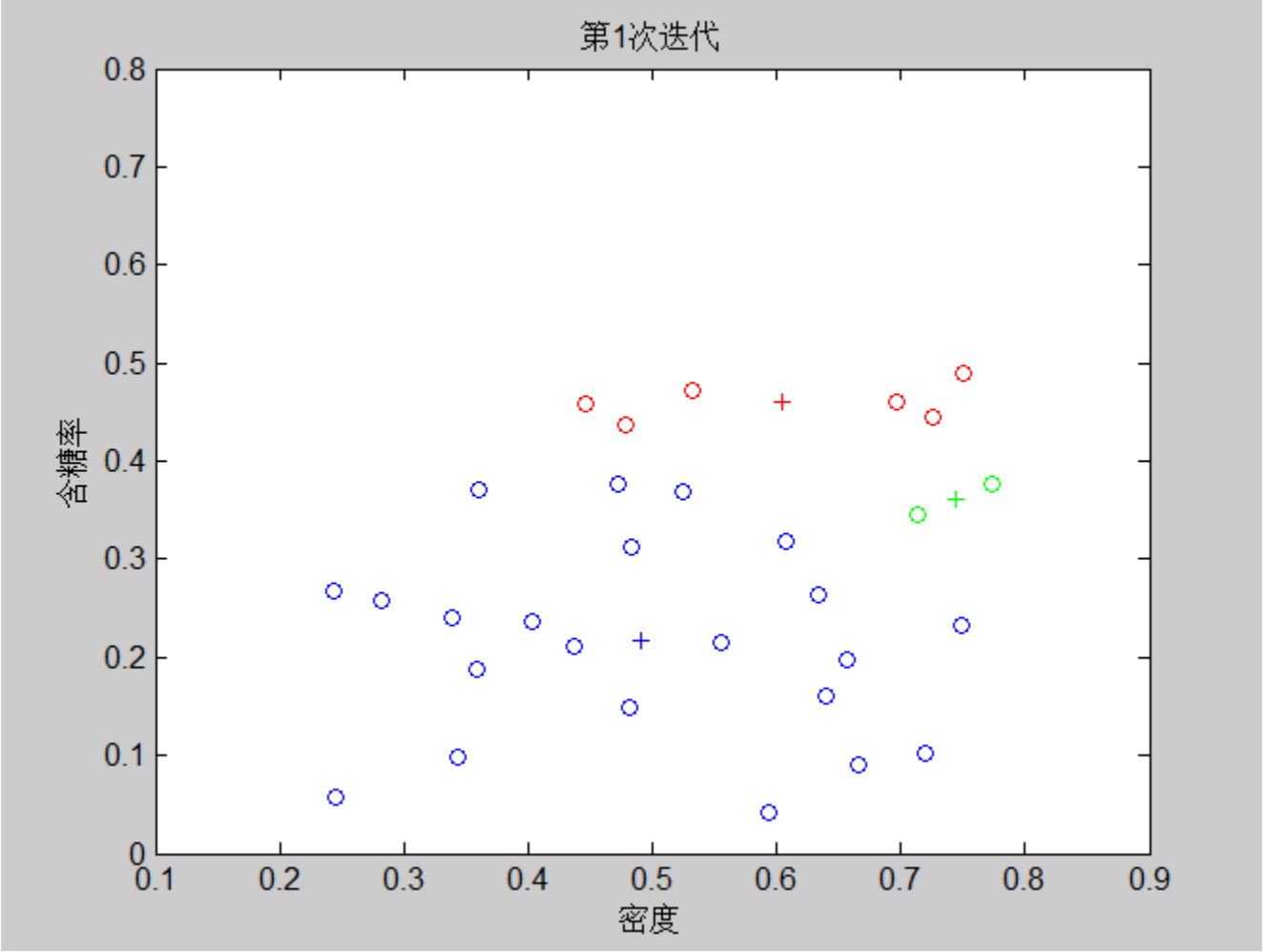
$$\mu_1 = x_1 = (0.697, 0.460); \mu_2 = x_2 = (0.774, 0.376); \mu_3 = x_3 = (0.634, 0.264)$$

迭代：

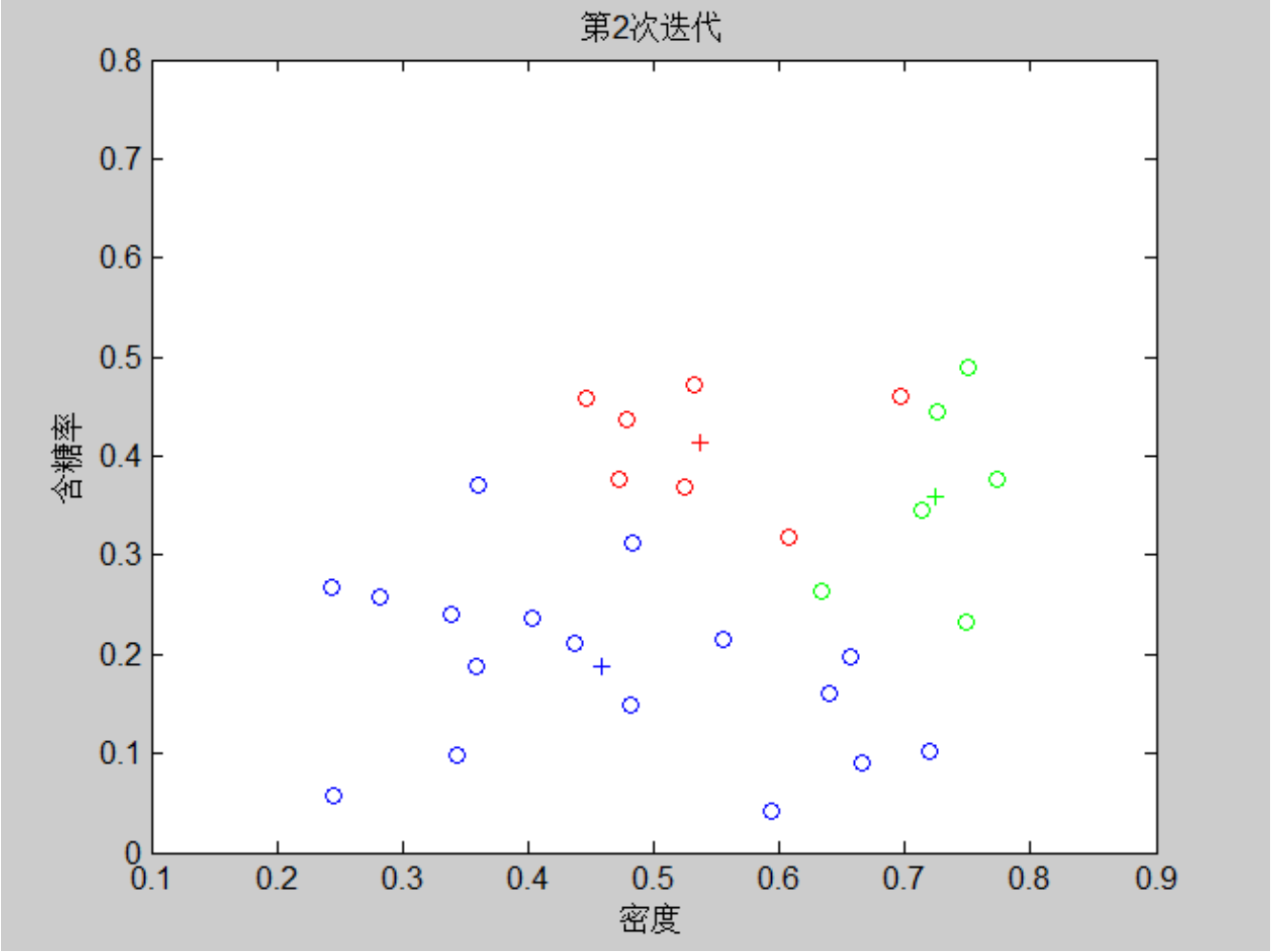
① 对每一个样本，计算其到3个均值向量的距离，并分配至距离最近的簇；

② 根据新产生的簇重新计算3个均值向量

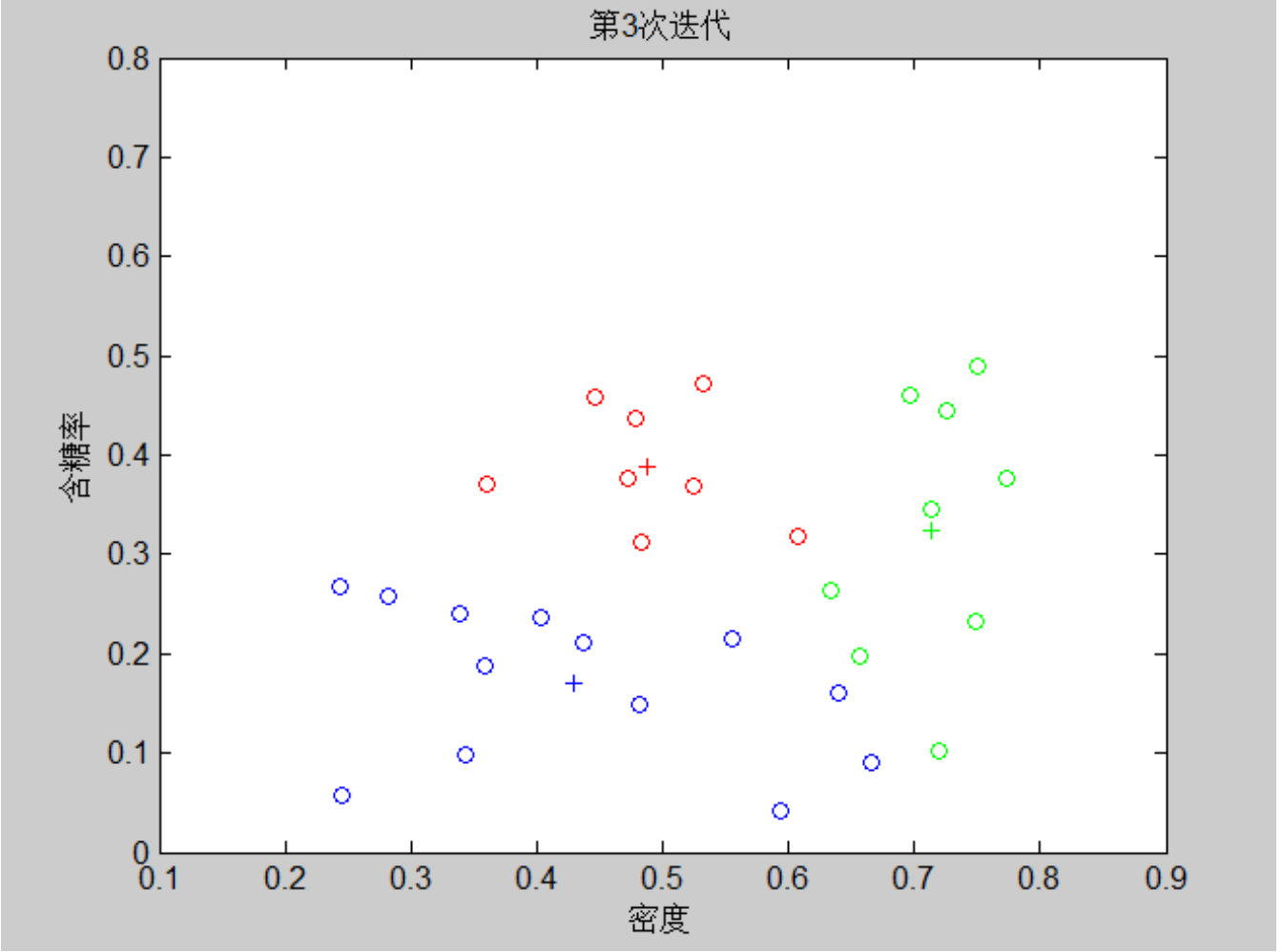
第一次迭代后结果



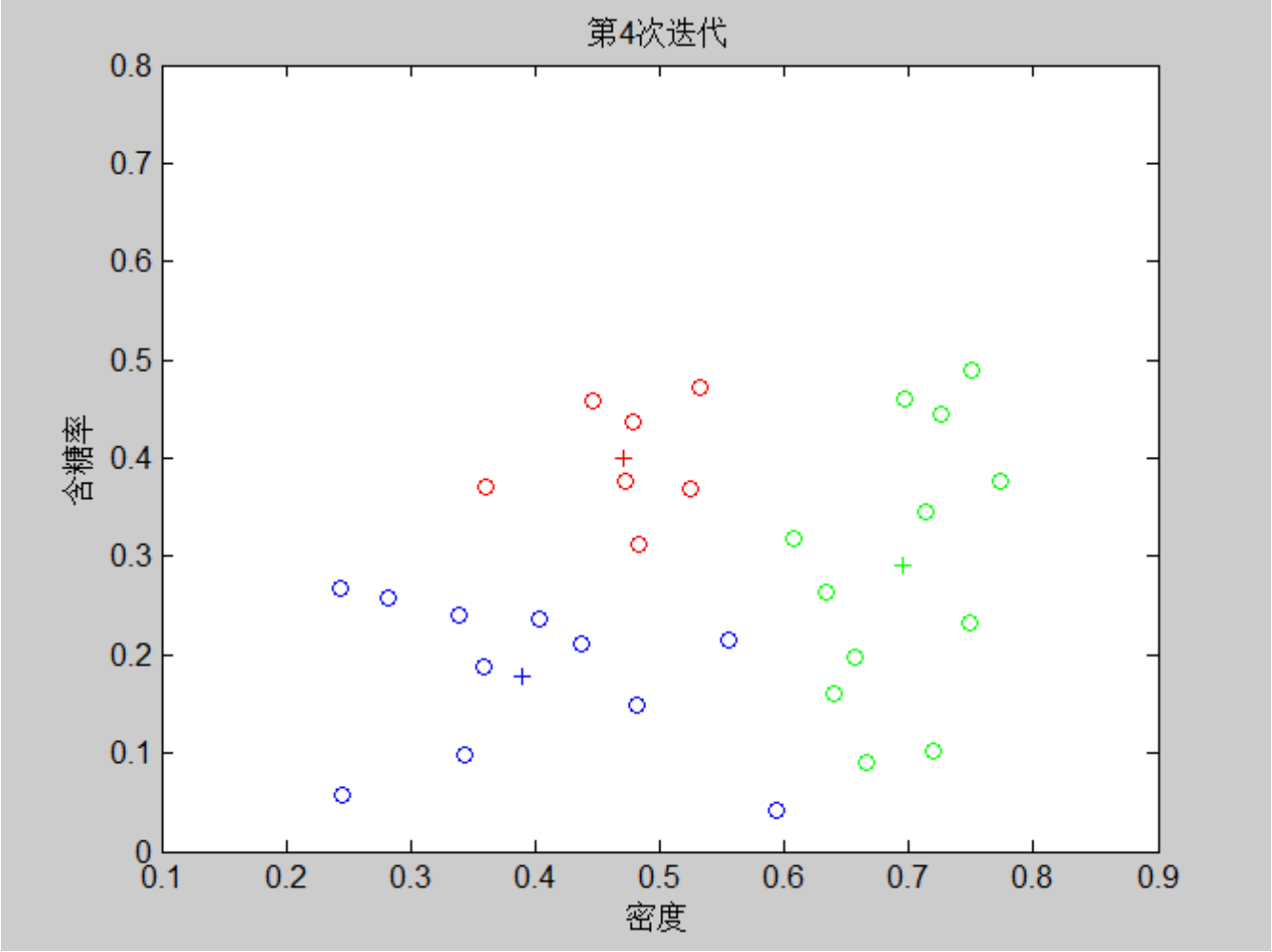
第二次迭代后结果



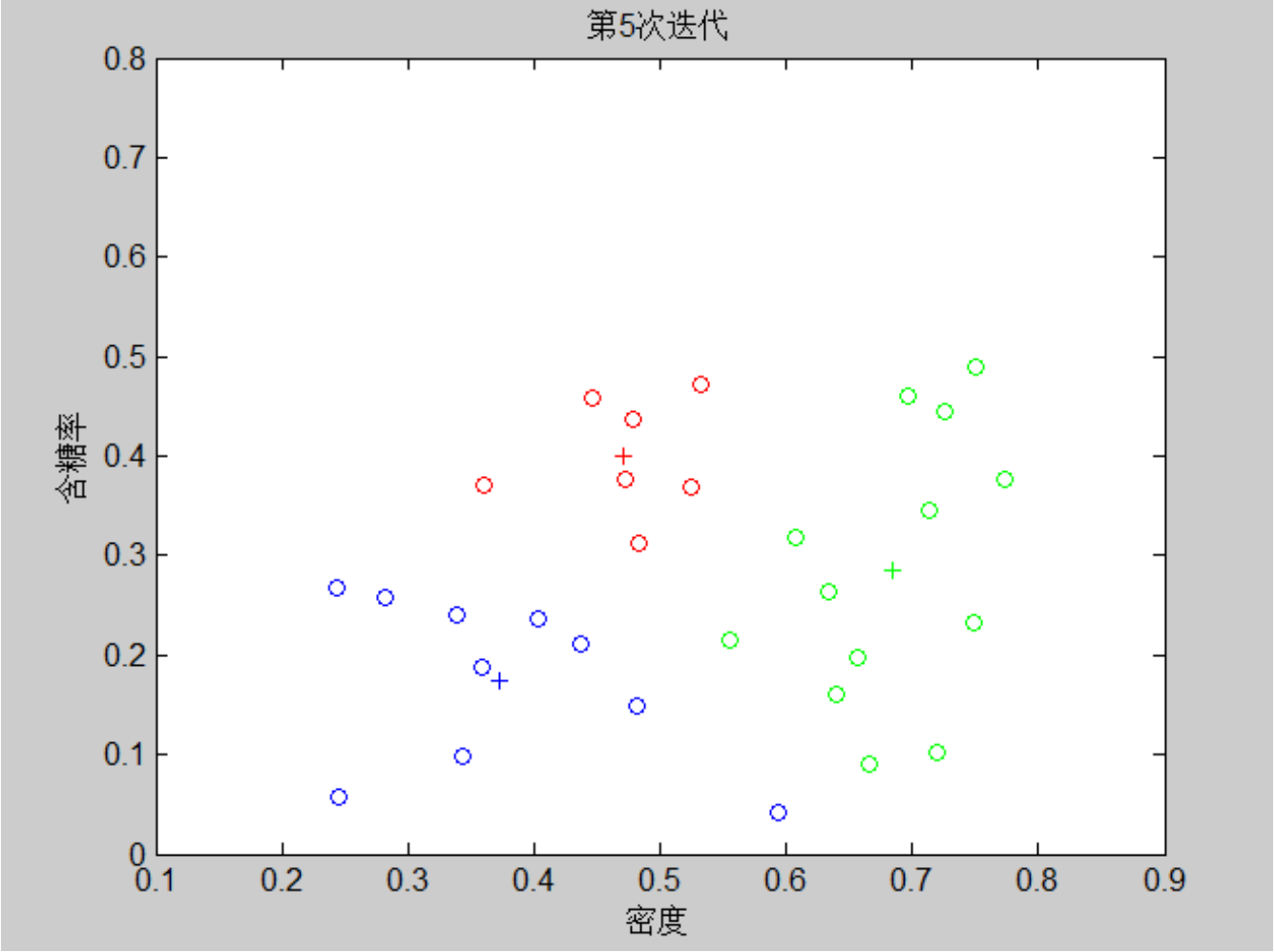
第三次迭代后结果



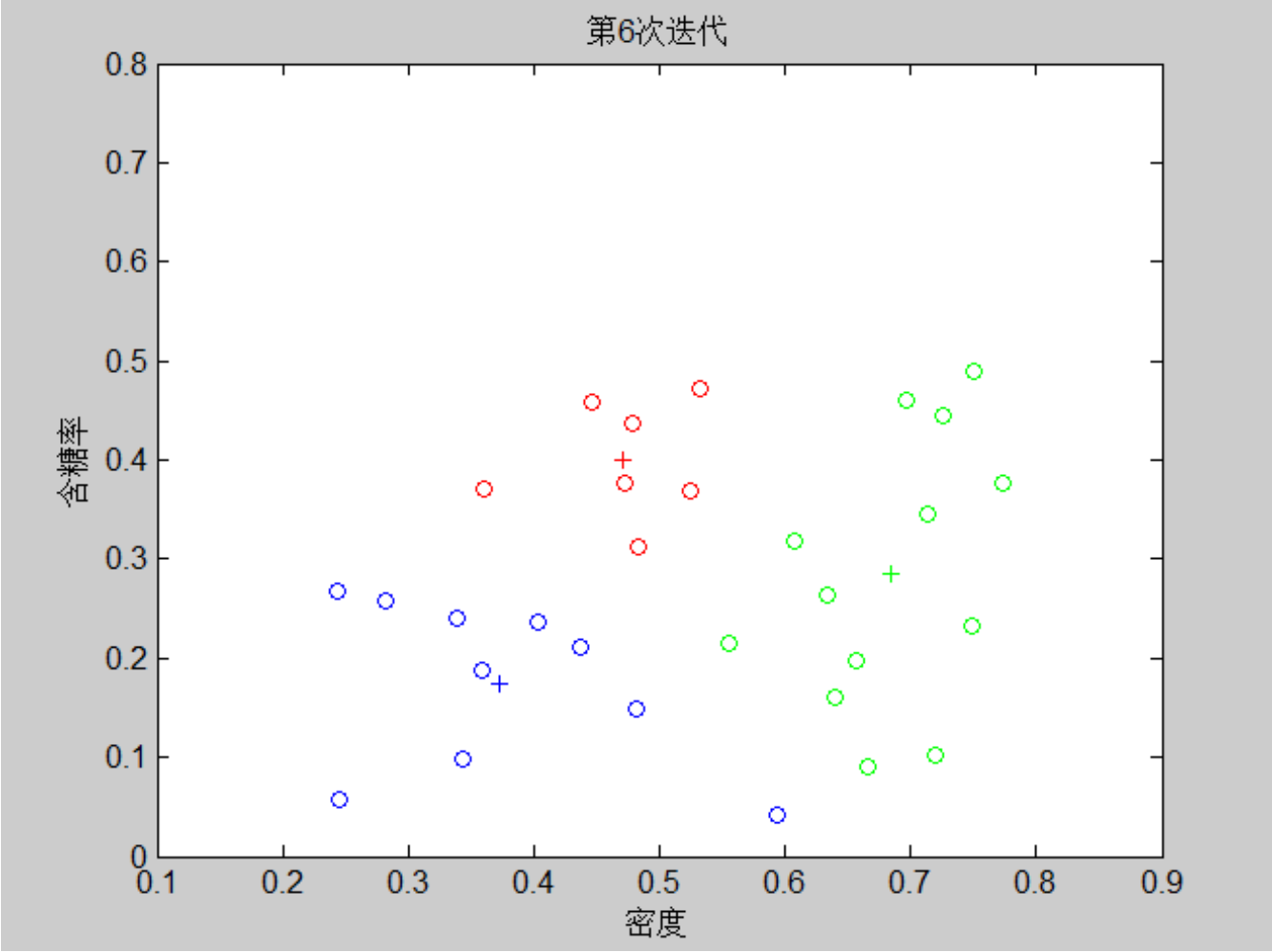
第四次迭代后结果



第五次迭代后结果



第六次迭代后结果



5.6 K-均值聚类

- K-均值聚类是**误差平方和最小准则**下的聚类方法
  - 设 n_i 表示属于 ω_i 类样本的个数，$\mathbf{m}_i$ 是这些样本的均值（注：这里将 $\boldsymbol{\mu}_i$ 换成 $\mathbf{m}_i$）：

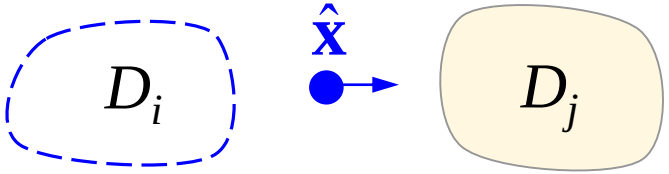
$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in D_i} \mathbf{x}$$

  - 考虑对所有样本的一个划分，计算划分后的样本与均值的误差平方和，得到如下“误差平方和”聚类准则：

$$J_e = \sum_{i=1}^{c} J_i, \quad \text{其中，} \quad J_i = \sum_{\mathbf{x} \in D_i} \left\| \mathbf{x} - \mathbf{m}_i \right\|^2$$

  - 对于不同的划分（聚类），会得到不同的 $\mathbf{m}_i$。因此 J_e 的值也是不同的。使 J_e 最小的聚类就是误差平方和准则下的最优结果。因此，称这类聚类方法为**最小方差划分法**。

5.6 K-均值聚类



- **迭代过程中的样本调整：**
  - 假设样本 $\hat{\mathbf{x}}$ 从类 D_i 移动到 D_j，此时，两个类中心将同时进行变化：

$$\mathbf{m}_j^* = \mathbf{m}_j + \frac{\hat{\mathbf{x}} - \mathbf{m}_j}{n_j + 1}, \qquad \mathbf{m}_i^* = \mathbf{m}_i - \frac{\hat{\mathbf{x}} - \mathbf{m}_i}{n_i - 1}$$

  - **属于第 j 类的样本点引起的误差平方和将增加为：**

$$J_j^* = \sum_{\mathbf{x} \in D_j} \left\| \mathbf{x} - \mathbf{m}_j^* \right\|^2 + \left\| \hat{\mathbf{x}} - \mathbf{m}_j \right\|^2 = J_j + \frac{n_j \left\| \hat{\mathbf{x}} - \mathbf{m}_j \right\|^2}{n_j + 1}$$

属于第 j 类的样本点引起的误差平方和将增加，推导如左。

$$J_j^* = \sum_{\mathbf{x}\in D_j} \left\|\mathbf{x}-\mathbf{m}_j^*\right\|^2 + \left\|\hat{\mathbf{x}}-\mathbf{m}_j^*\right\|^2$$

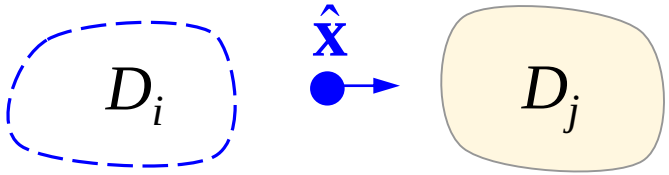
$$= \sum_{\mathbf{x}\in D_j} \left\|\mathbf{x}-\mathbf{m}_j - \frac{\hat{\mathbf{x}}-\mathbf{m}_j}{n_j+1}\right\|^2 + \left\|\frac{n_j}{n_j+1}(\hat{\mathbf{x}}-\mathbf{m}_j)\right\|^2$$

$$= \sum_{\mathbf{x}\in D_j} \left(\left\|\mathbf{x}-\mathbf{m}_j\right\|^2 - \frac{2}{n_j+1}(\mathbf{x}-\mathbf{m}_j)^T(\hat{\mathbf{x}}-\mathbf{m}_j) + \frac{\left\|\hat{\mathbf{x}}-\mathbf{m}_j\right\|^2}{(n_j+1)^2} \right) + \left\|\frac{n_j}{n_j+1}(\hat{\mathbf{x}}-\mathbf{m}_j)\right\|^2$$

$$= J_j - \frac{2}{n_j+1}(\hat{\mathbf{x}}-\mathbf{m}_j)^T \left(\sum_{\mathbf{x}\in D_j}\mathbf{x} - \sum_{\mathbf{x}\in D_j}\mathbf{m}_j \right) + \frac{n_j\left\|\hat{\mathbf{x}}-\mathbf{m}_j\right\|^2}{(n_j+1)^2} + \left\|\frac{n_j}{n_j+1}(\hat{\mathbf{x}}-\mathbf{m}_j)\right\|^2$$

$$= J_j - \frac{2}{n_j+1}(\hat{\mathbf{x}}-\mathbf{m}_j)^T \left(n_j\mathbf{m}_j - n_j\mathbf{m}_j \right) + \frac{n_j\left\|\hat{\mathbf{x}}-\mathbf{m}_j\right\|^2}{n_j+1)}$$

$$= J_j + \frac{n_j\left\|\hat{\mathbf{x}}-\mathbf{m}_j\right\|^2}{n_j+1}$$



- **迭代过程中的样本调整：**
  - **属于第 i 类的样本点引起的误差平方和将减少为：**

$$J_i^* = J_i - \frac{n_i \left\| \hat{\mathbf{x}} - \mathbf{m}_i \right\|^2}{n_i - 1}$$

  - 如果**减少量大于增加量**，因此**鼓励这种移动，即**将样本 $\hat{\mathbf{x}}$ 从类 D_i 移动到 D_j 会减少总体误差：

$$\frac{n_j \left\| \hat{\mathbf{x}} - \mathbf{m}_j \right\|^2}{n_j + 1} < \frac{n_i \left\| \hat{\mathbf{x}} - \mathbf{m}_i \right\|^2}{n_i - 1}$$

如果减少量大于增加量，此时鼓励移动

从一个类引出样本会减少该类均方误差；但移入样本至一个类会增加该类均方误差。如果**减少量大于增加量**，对这样的样本进行移动是有利于总体误差减少的。

K-Means Clustering—Algorithm2 (minimum squared error clustering)

1 begin initialization n, c, $\mathbf{m}_1$, $\mathbf{m}_2$,…, $\mathbf{m}_c$.

2 do randomly select a sample $\hat{\mathbf{x}}$

3 $i \leftarrow \arg\min_{i'} \| \mathbf{m}_{i'} - \hat{\mathbf{x}} \|$ // classify $\hat{\mathbf{x}}$

4 if $n_i \neq 0$, then compute

$$\rho_j = \begin{cases} \frac{n_j}{n_j+1} \| \hat{\mathbf{x}} - \mathbf{m}_j \|, & j \neq i \\ \frac{n_j}{n_j-1} \| \hat{\mathbf{x}} - \mathbf{m}_j \|, & j = i \end{cases}$$

5 find the minimum ρ_k among all ρ_j, j=1,2,…,c

6 if $\rho_k \leq \rho_j$ for all j, then transfer $\hat{\mathbf{x}}$ to D_k

7 re-compute J_e, $\mathbf{m}_i$, $\mathbf{m}_k$

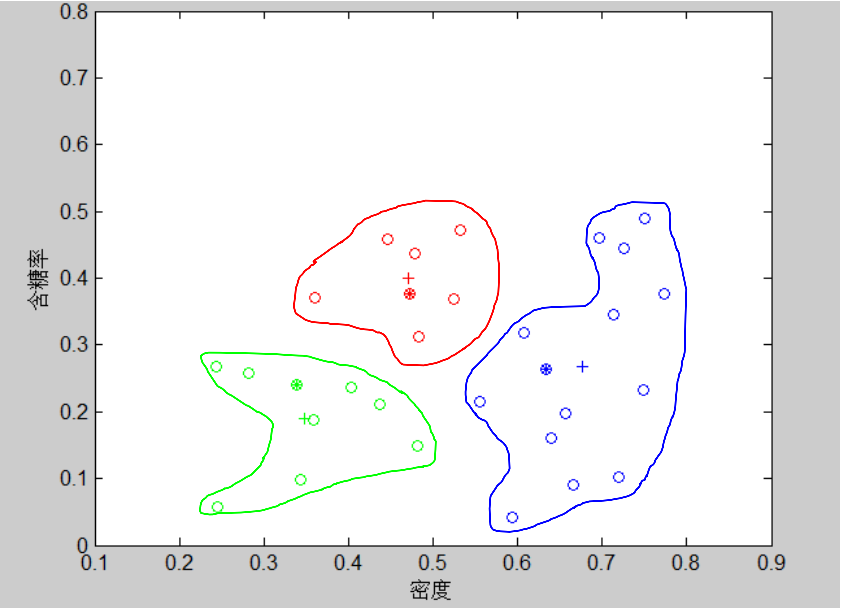
8 until no change in J_e for all n samples

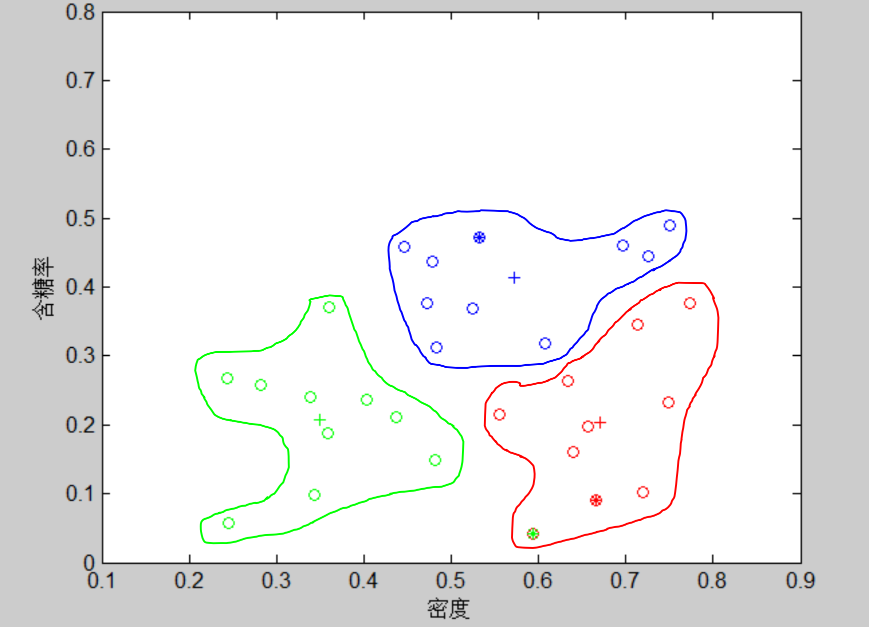
9 return $\mathbf{m}_1$, $\mathbf{m}_2$,…, $\mathbf{m}_c$

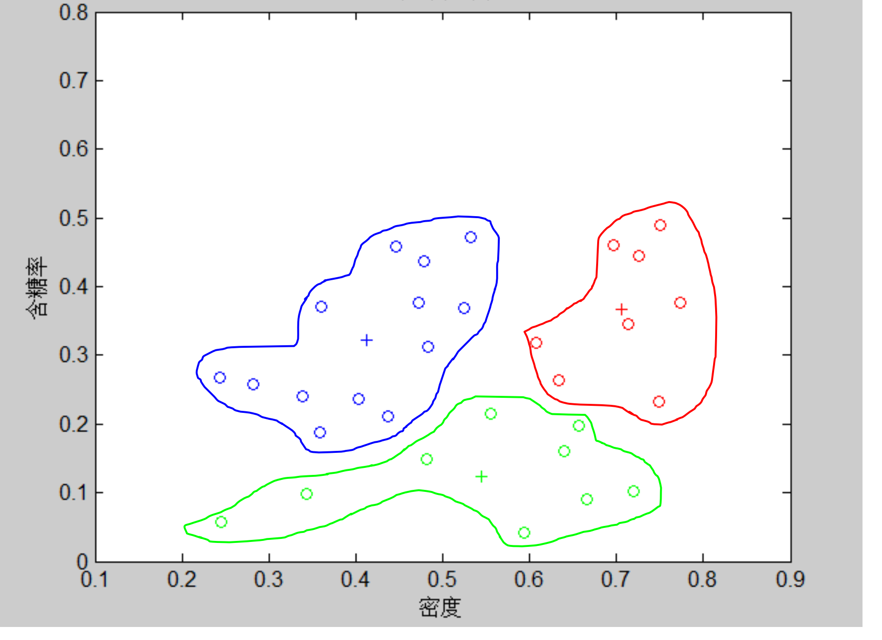
5.6 K-均值聚类

- K-均值聚类算法是一种典型的**动态聚类方法**，具有如下三个要点：
  - (1) 选择欧氏距离度量作为样本间的相似性度量。
  - (2) 采用最大似然估计或最小均方误差作为评价聚类的准则函数。
  - (3) 给定某个初始分类，然后采用迭代算法寻找准则函数的极值。
- **优点：**
  - 是解决聚类问题的一种经典算法，简单、快速。
  - 对处理大数据集，该算法仍可保持其高效率。
  - 对于密集簇，聚类效果很好。
- **缺点：**
  - 必须事先给定簇的个数，且对初始值敏感。
  - 不适合于发现非凸曲面的簇以及大小相差很大的簇。
  - 对噪声、孤立数据点、野点很敏感。

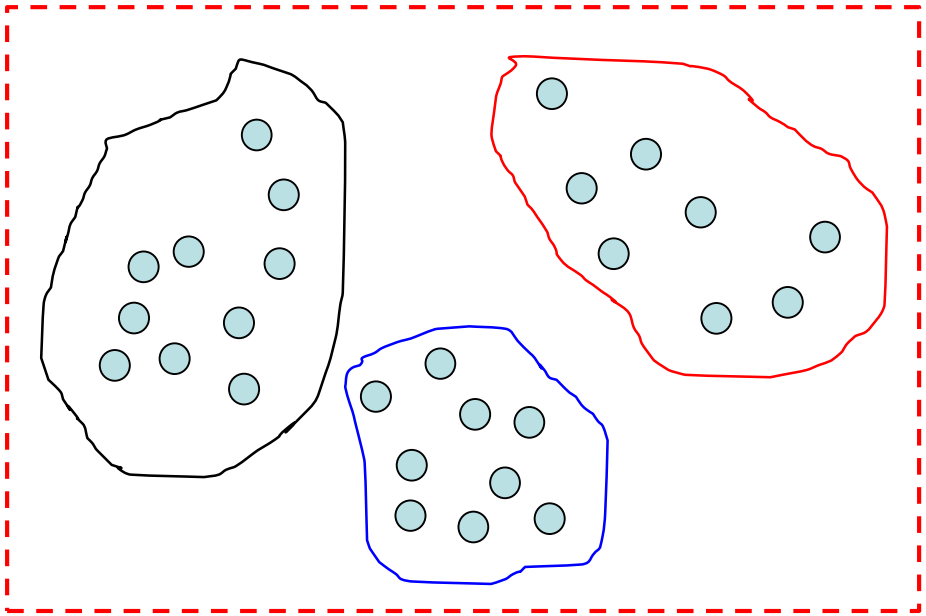
不同初始值可能会得到不同的聚类结果



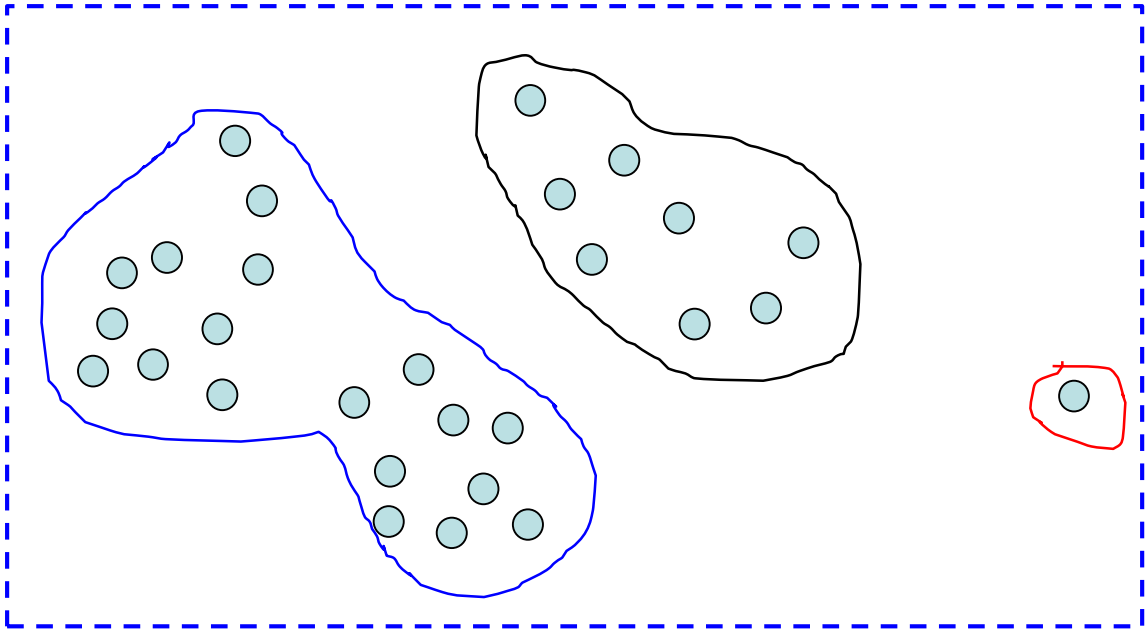




中国科学院大学 University of Chinese Academy of Sciences



孤立数据点对聚类结果影响很大

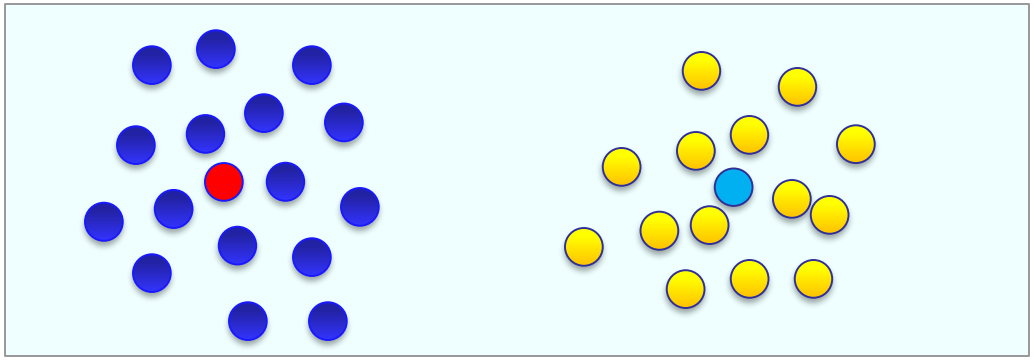


5.7 谱聚类

- **谱聚类**
  - 从图切割的角度，聚类就是要找到一种合理的分割图的方法，**分割后能形成若干个子图**。连接不同子图的边的权重尽可能小，子图内部边权重尽可能大。
  - 谱聚类算法建立在**图论中的谱图理论**基础之上，其本质是将聚类问题转化为一个**图上的关于顶点划分的最优问题**。
  - 谱聚类算法建立在**点对亲和性**基础之上，理论上能对任意分布形状的样本空间进行聚类。
  - 最早关于谱聚类的研究始于1973年，主要用于计算机视觉和VLSI设计领域。从2000年开始，谱聚类逐渐成为机器学习领域中的一个研究热点。

5.7.1 引言

- **回忆K-means聚类**

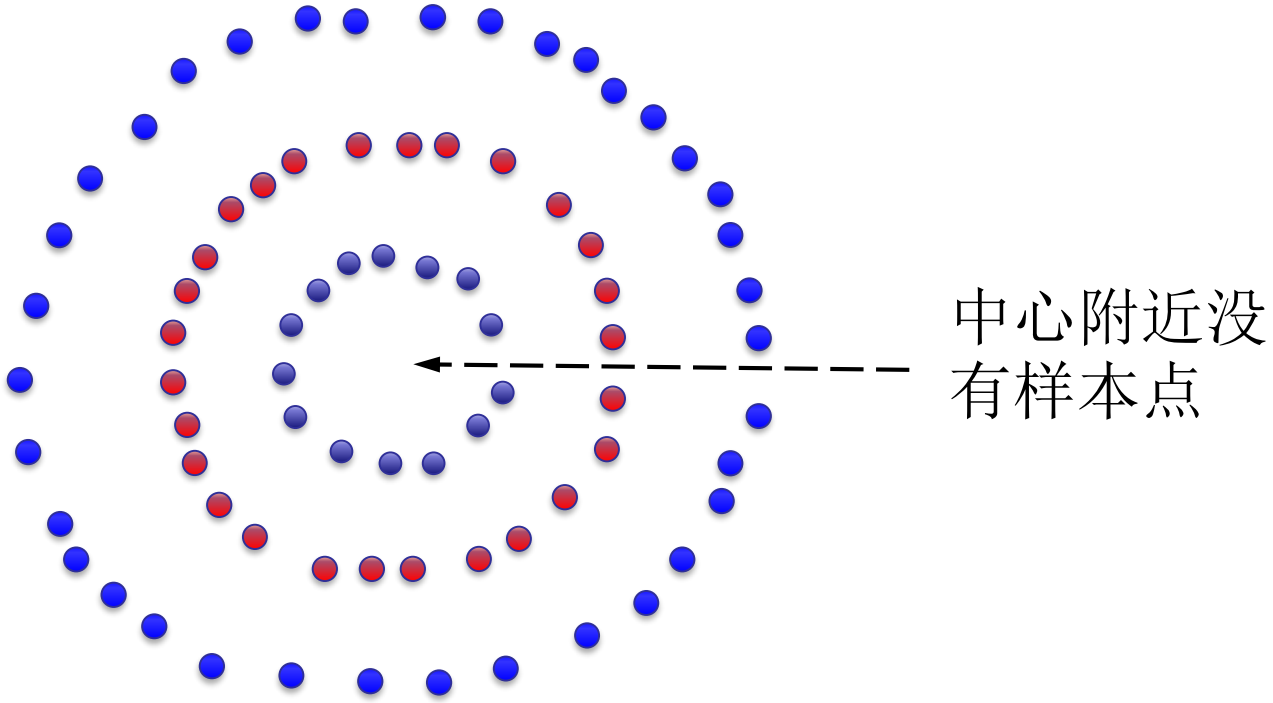


每个簇均可以用中心点来表示

(特别适合于单个簇符合高斯分布的情形)

5.7.1 引言

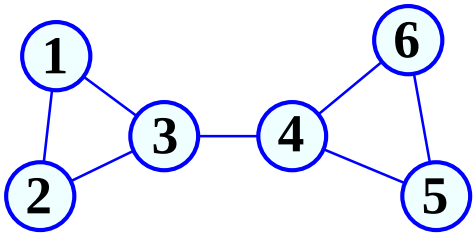
- 对于其它分布情形



Spectral clustering allows us to address these sorts of clusters!

5.7.2 图论基本概念

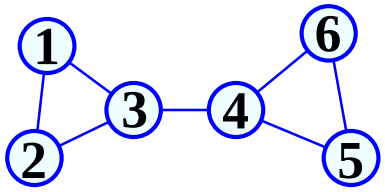
- **图 G：** 由顶点集 V 和边集 E 所构成，记为 G(V,E)。根据边是否有向，可以分为无向图或者有向图。
- **图 G 的邻接矩阵 W：**
  - 行数和列数等于矩阵顶点的个数；
  - 矩阵元素为 0 或 1。1 表示对应的一对顶点有边相连，0表示没有边相连。



$$\mathbf{W}=\begin{pmatrix}0&1&1&0&0&0\\1&0&1&0&0&0\\1&1&0&1&0&0\\0&0&1&0&1&1\\0&0&0&1&0&1\\0&0&0&1&1&0\end{pmatrix}$$

5.7.2 图论基本概念

- **顶点的度：等于与顶点相连接的边的条数。**
- **度矩阵：　为一个对角矩阵。**将邻接矩阵各行元素累加至对应的主对角元素，可得到**度矩阵 D。**

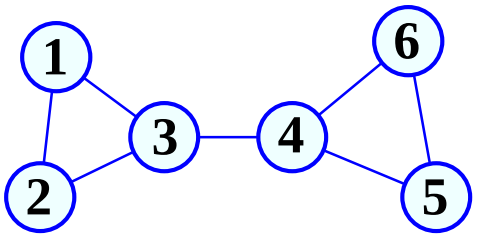


$$\mathbf{W}=\begin{pmatrix}0&1&1&0&0&0\\1&0&1&0&0&0\\1&1&0&1&0&0\\0&0&1&0&1&1\\0&0&0&1&0&1\\0&0&0&1&1&0\end{pmatrix},\quad \mathbf{D}=\begin{pmatrix}2&&&&&\\&2&&&&\\&&3&&&\\&&&3&&\\&&&&2&\\&&&&&2\end{pmatrix}$$

5.7.2 图论基本概念

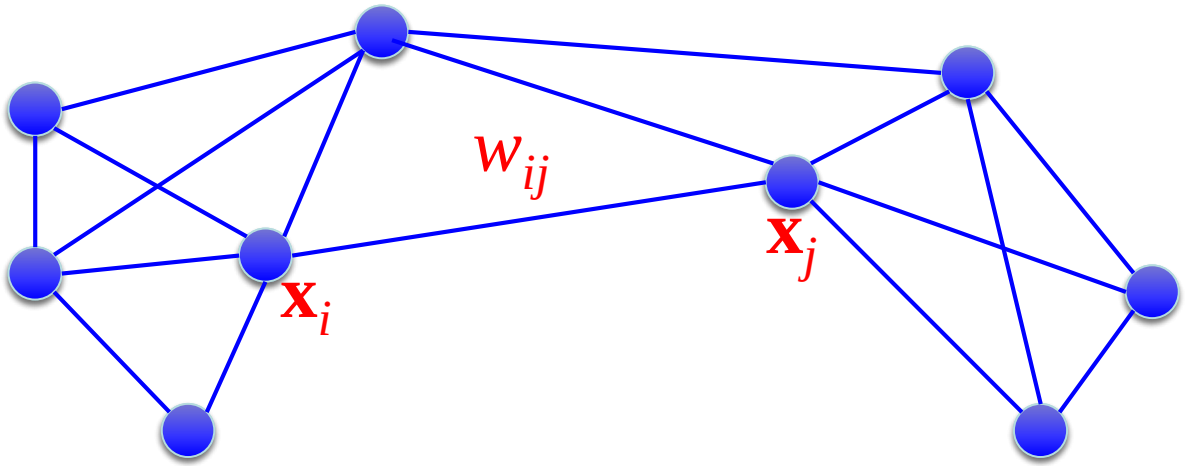
- **拉普拉斯矩阵**
  - **度矩阵**减去**邻接矩阵**得到**拉普拉斯矩阵 L**。

$$\mathbf{D}=\begin{pmatrix}2&&&&&\\&2&&&&\\&&3&&&\\&&&3&&\\&&&&2&\\&&&&&2\end{pmatrix},\quad \mathbf{L}=\mathbf{D}-\mathbf{W}=\begin{pmatrix}2&-1&-1&0&0&0\\-1&2&1&0&0&0\\-1&-1&3&-1&0&0\\0&0&-1&3&-1&-1\\0&0&0&-1&2&-1\\0&0&0&-1&-1&2\end{pmatrix}$$



5.7.2 图论基本概念

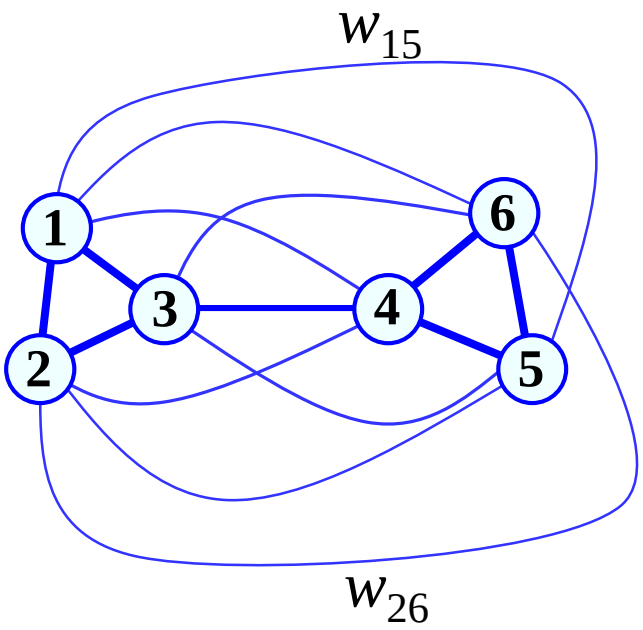
- **基于数据集的图构造**
  - 将每一个数据点视为图的一个顶点，顶点之间可以有边相连。每条边上加上一些权重，用来反映点对亲和性（即相似性）。



比如，采用高斯函数计算点对亲和性：$w_{ij} = \exp\left(-\frac{\| \mathbf{x}_i - \mathbf{x}_j \|_2^2}{2\sigma^2}\right)$

5.7.2 图论基本概念

- **图构造 G(V, E)**
  - 根据某种测度构建点对相似度矩阵



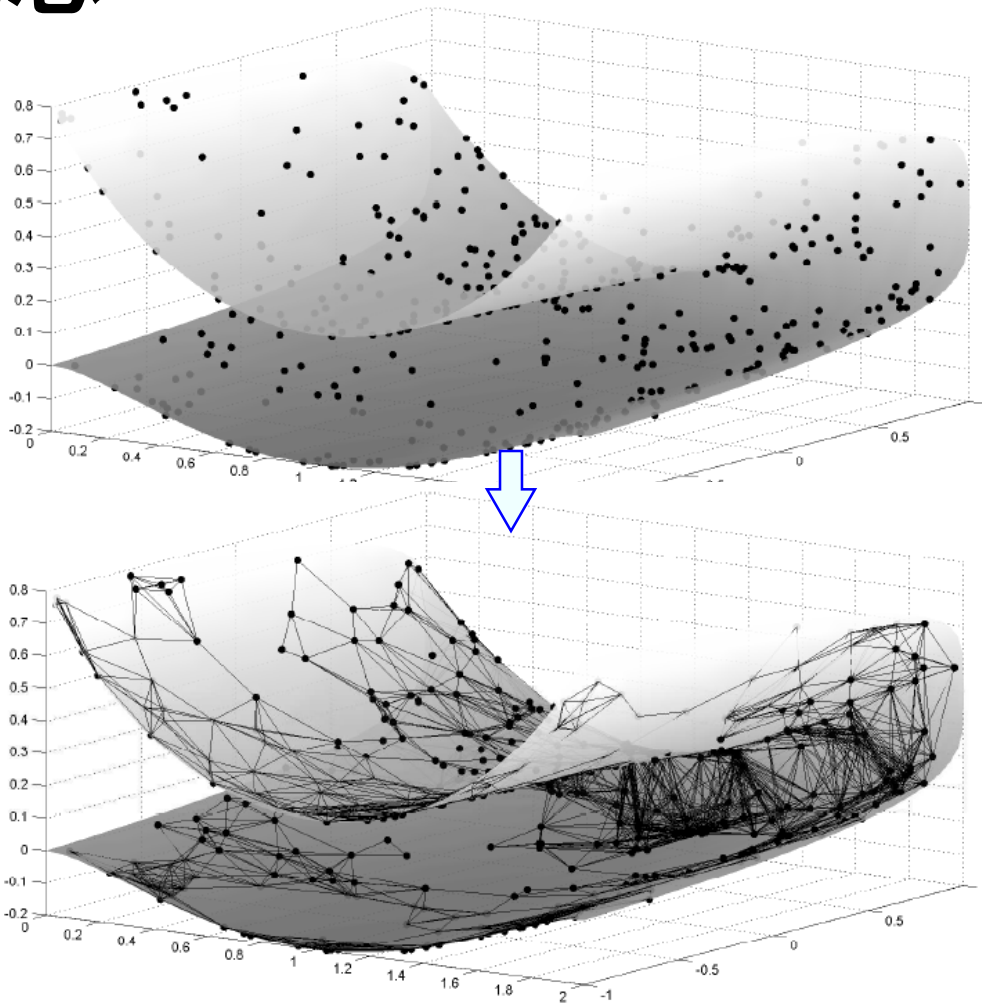
$$\begin{pmatrix} 0 & w_{12} & w_{13} & w_{14} & w_{15} & w_{16} \\ & 0 & w_{23} & w_{24} & w_{25} & w_{26} \\ & & 0 & w_{34} & w_{35} & w_{36} \\ & & & 0 & w_{45} & w_{46} \\ & & & & 0 & w_{56} \\ & & & & & 0 \end{pmatrix}$$

对称

点对相似度矩阵

5.7.2 图论基本概念

- 图构造 G(V, E)
  - 全连接
  - 局部连接
    - k - 近邻
    - ε - 半径

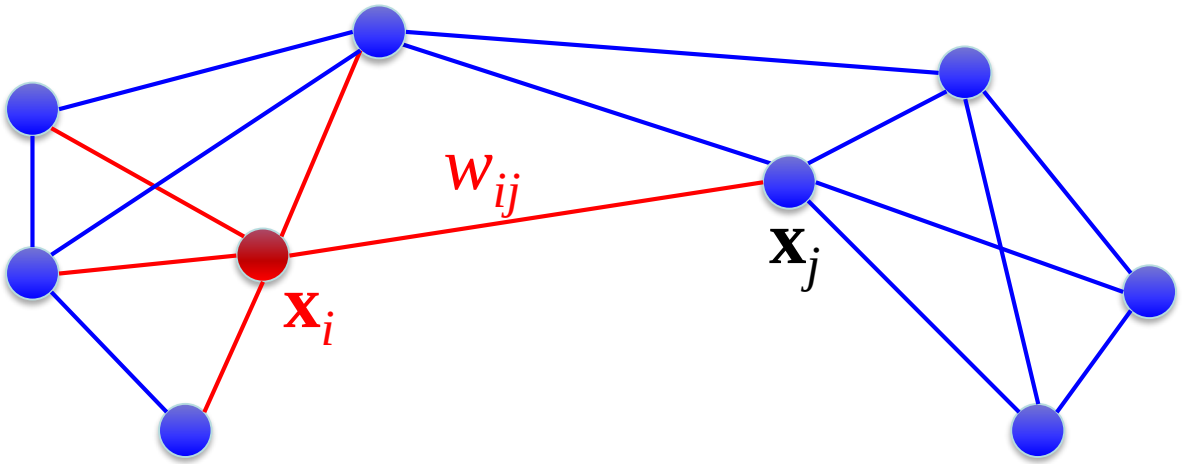


***k*-近邻：** 对每个数据点 $\mathbf{x}_i$，首先在所有样本中找出不包含 $\mathbf{x}_i$ 的 k 个最邻近的样本点，然后 $\mathbf{x}_i$ 与每个邻近样本点均有一条边相连，从而完成图构造。

为了保证W矩阵的对称性，可以令$\mathbf{W}=(\mathbf{W}^T+\mathbf{W})/2$

5.7.2 图论基本概念

- **顶点的度：**所有与该顶点相连接的边的权重之和。



$$d_i = \sum_{j \in V} w_{ij}$$

（如果顶点 $\mathbf{x}_j$ 不与 $\mathbf{x}_i$ 相边接，则 $w_{ij}= 0$。）

5.7.2 图论基本概念

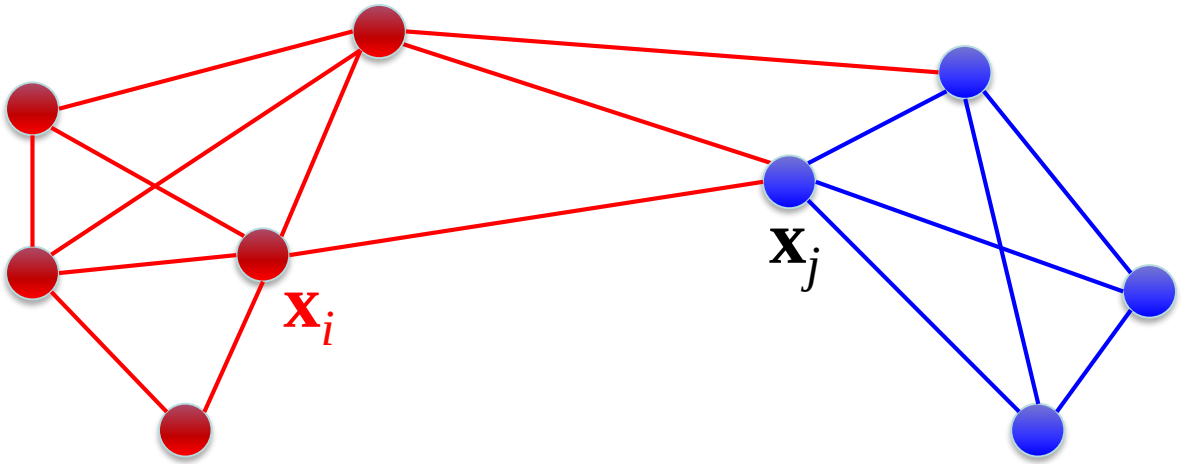
- **拉普拉斯矩阵(Laplacian matrix)**
  - 拉普拉斯矩阵是描述图的一种矩阵。给定一个具有 n 个顶点的图，其拉普拉斯矩阵描述为：

$$\mathbf{L} = \mathbf{D} - \mathbf{W}$$

  - 其中，$\mathbf{D}$ 为一个对角矩阵，主对角元素表示顶点的度。$\mathbf{W}$ 为亲和度矩阵，其元素 w_{ij} 表示顶点 $\mathbf{x}_i$ 与 $\mathbf{x}_j$ 之间的亲和程度（即相似度）。

5.7.2 图论基本概念

- **子图 A⊂V 的势 |A|：** 等于其所包含的顶点个数。
- **子图 A⊂V 的体积 *vol*(*A*)：** 等于其中所有顶点的度之和。



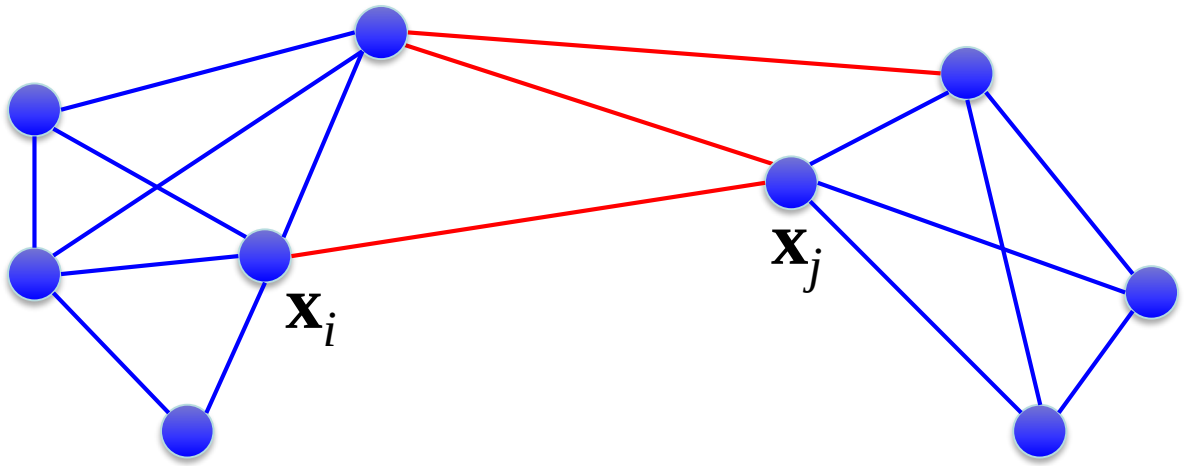
$$vol(A) = \sum_{i \in A} d_i$$

5.7.2 图论基本概念

- **子图A的补图：** V 中去掉 A 的顶点所构成的子图 $\bar{A}$ ：

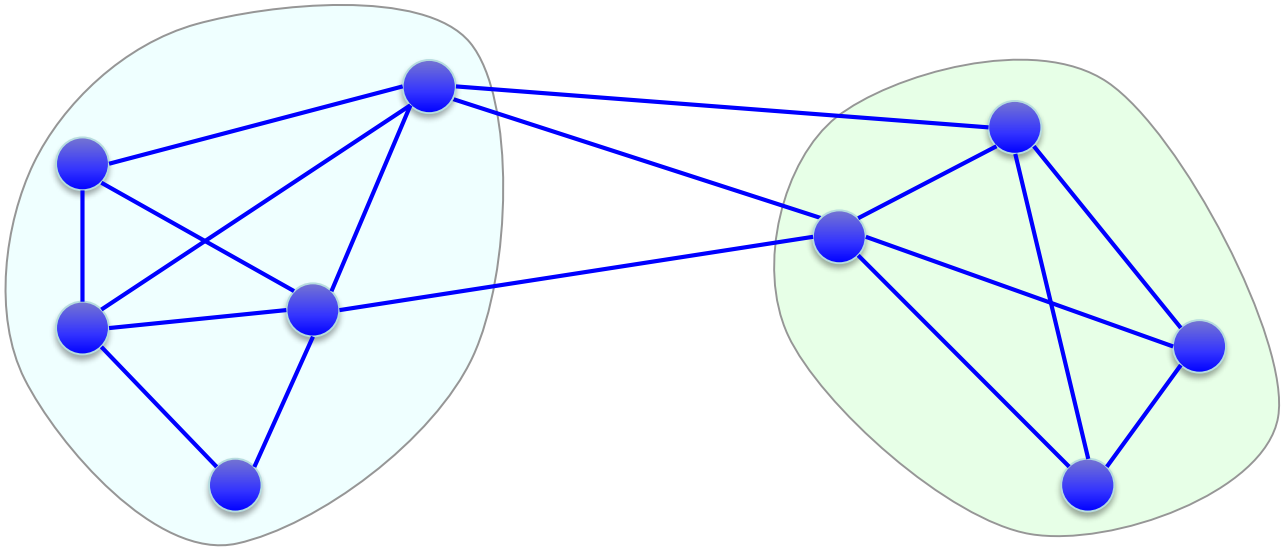
$$\bar{A} = V - A$$

- **边割：** 指边 E 的一个子集，去掉该子集中的边，图就变成两个两通子图。



5.7.2 图论基本概念

- **图切割：**
  - 设 A_1 , A_2 , …, A_k 为顶点集合 A 的非空连通子集，如果 $A_i \cap A_j = \varnothing, i \neq j$, 且 $A_1 \cup A_2 \cup \ldots \cup A_k = V$, 则称 A_1 , A_2 , …, A_k 为图 G 的一个分割。



5.7.2 图论基本概念

- **子图相似度：子图 A 与子图 B 的相似度**定义为连接两个子图所有边的权重之和：

$$W(A,B)=\sum_{i\in A,\, j\in B} w_{ij}$$

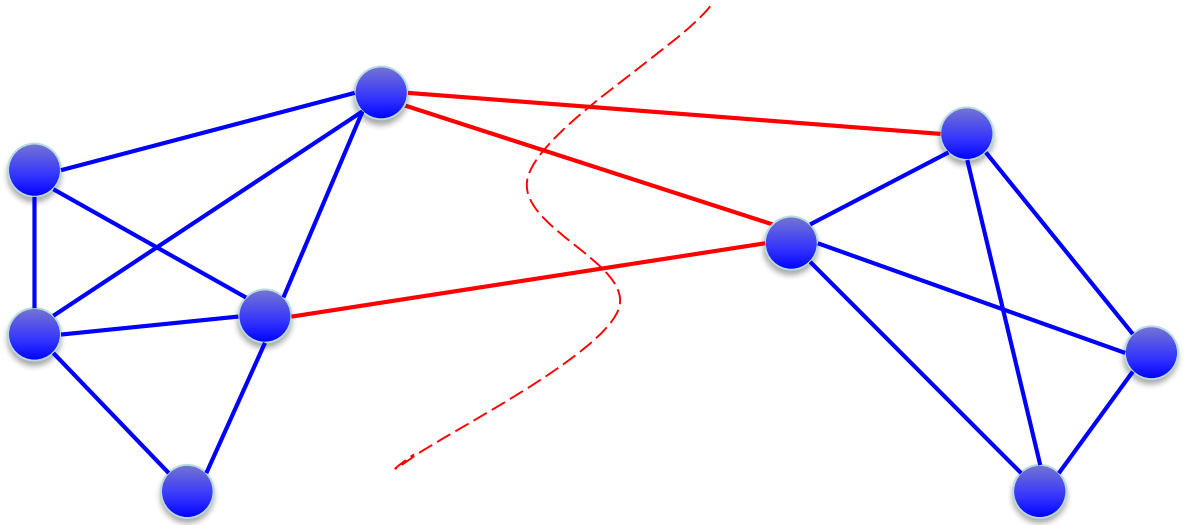
- **子图之间的切割：子图 A 与子图 B 的切割定义：**

$$cut(A,B)=W(A,B)=\sum_{i\in A,\, j\in B} w_{ij}$$

注：如果两个顶点不相连，则权重为零。

5.7.3 图切割

- **最小二分切割 (Minimum bipartitional cut)**
  - 在所有的图切割中，找一个最小代价的切割，将图分为两个不连通的子图。也就是说，切开之后，两个子图之间的相似性要最小。



5.7.3 图切割

- **最小二分切割**
  - 在所有的图切割中，找一个最小代价的切割，将图分为两个不连通的子图。也就是说，切开之后，两个子图之间的相似性要最小。**最优化问题如下：**

$$\min_{A} \quad \mathrm{cut}(A, \bar{A}) := W(A, \bar{A}) = \sum_{i \in A,\, j \in \bar{A}} w_{ij}$$

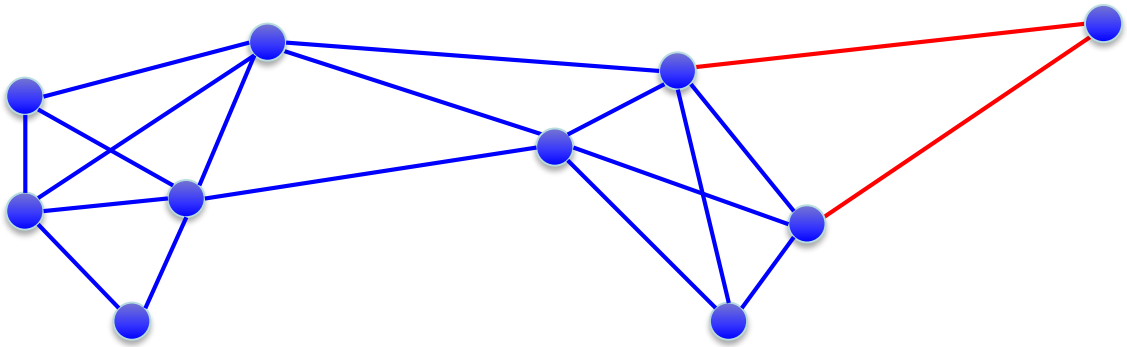
$$s.t. \quad A \neq \varnothing,$$

$$A \cap \bar{A} = \varnothing,$$

$$A \cup \bar{A} = V$$

5.7.3 图切割

- 最小二分切割
  - **在实践中，上述目标函数通常将一个点(比如野点)从其余各点中分离出来**。从聚类的角度看，这并不是我们所期望的。



5.7.3 图切割

- **归一化最小二分切割**
  - 出现上述问题的原因在于**对子图的规模没有加以限制。**
  - 一个基本的假设是希望两个子图的**规模不要相差太大。**
  - 一个基本的做法是采用**子图的势或者体积**来对切割进行归一化，即定义如下目标函数：
    - 采用子图的势：

$$\mathrm{Ratiocut}(A,\bar{A}) := \frac{1}{2}\left(\frac{cut(A,\bar{A})}{|A|}+\frac{cut(A,\bar{A})}{|\bar{A}|}\right)$$

    - 采用子图的体积：

$$\mathrm{Ncut}(A,\bar{A}) := \frac{1}{2}\left(\frac{cut(A,\bar{A})}{vol(A)}+\frac{cut(A,\bar{A})}{vol(\bar{A})}\right)$$

5.7.3 图切割

- **K-切割 (k > 2)：**
  - 考虑将图分成 k 个子图：A_1，A_2，…，A_k。一种直观的方法是将图切割问题理解为多个二分切割问题的综合。
- **未归一化切割目标函数：**

$$\mathrm{cut}(A_1, A_2, \cdots, A_k) := \frac{1}{2}\sum_{i=1}^{k} W(A_i, \bar{A}_i)$$

- **比例切割目标函数：**

$$\mathrm{Ratiocut}(A_1, A_2, \cdots, A_k) := \frac{1}{2}\sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{|A_i|} = \sum_{i=1}^{k} \frac{\mathrm{cut}(A_i, \bar{A}_i)}{|A_i|}$$

- **归一化切割目标函数：**

$$\mathrm{Ncut}(A_1, A_2, \cdots, A_k) := \frac{1}{2}\sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{\mathrm{vol}(A_i)} = \sum_{i=1}^{k} \frac{\mathrm{cut}(A_i, \bar{A}_i)}{\mathrm{vol}(A_i)}$$

5.7.4 拉普拉斯矩阵的性质

- 拉普拉斯矩阵： $\mathbf{L} = \mathbf{D} - \mathbf{W}$
- 性质：
  - **L 的行和为零：**
  - **L 有一个特征值为零**，其对应的特征向量为一个元素全为 1 的向量：
  - **L 有 *n* 个非负的特征值， *n* 为图的顶点个数**。
  - **L 是半正定矩阵。**

5.7.4 拉普拉斯矩阵的性质

- 性质(续)：
  - **L** 是半正定矩阵，对任意向量 $\mathbf{f} = [f_1, f_2, \ldots, f_n]^T$, 有：

$$\mathbf{f}^T\mathbf{L}\mathbf{f} = \mathbf{f}^T\mathbf{D}\mathbf{f} - \mathbf{f}^T\mathbf{W}\mathbf{f} = \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} f_i f_j w_{ij}$$

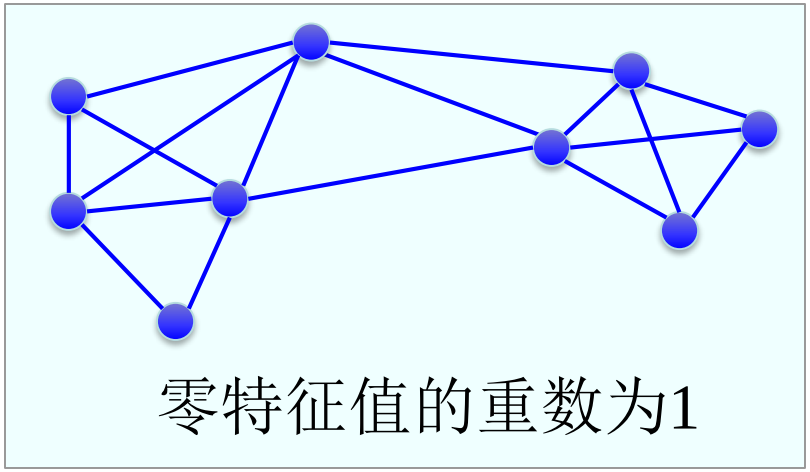
$$= \frac{1}{2}\left(\sum_{i=1}^{n} d_i f_i^2 - 2\sum_{i,j=1}^{n} f_i f_j w_{ij} + \sum_{j=1}^{n} d_j f_j^2 \right)$$

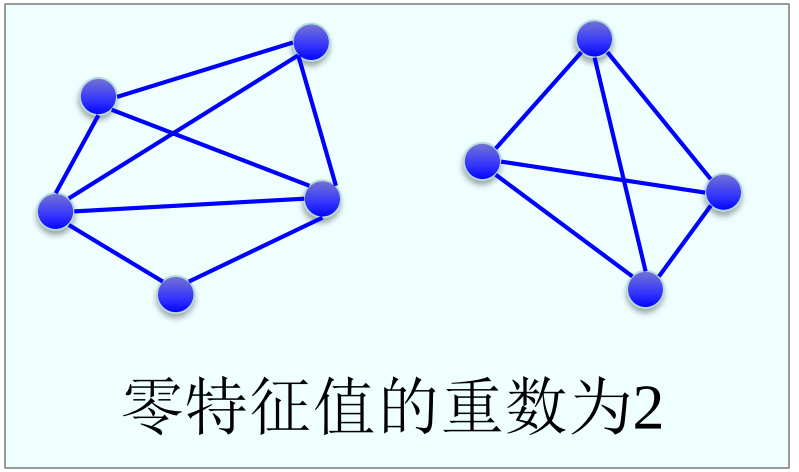
$$= \frac{1}{2}\left(\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2 \right)$$

$$\geq 0$$

5.7.4 拉普拉斯矩阵的性质

- 性质(续)：
  - 图的连通子图与拉普拉斯矩阵 L 的特征值的关系：
    - 设 G 为一个具有非负连接权重的无向图，由图 G 导出的拉普拉斯矩阵 L 的零特征值的重数等于图 G 的连通子图的个数 k。





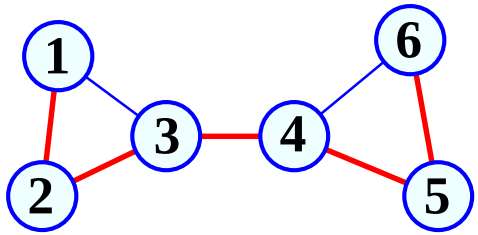
5.7.4 拉普拉斯矩阵的性质

- **证明：**
  - 首先考虑图 G 是连通的，即 $k = 1$。对此情形，需要证明 **L** 矩阵只有一个特征值为 0，且对应的特征向量由元素全为 1 的向量所构成。
  - 假定 **f** 为特征值 0 对应的特征向量，于是有：

$$0 = \mathbf{f}^T 0\mathbf{f} = \mathbf{f}^T \mathbf{L}\mathbf{f} = \frac{1}{2}\left(\sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2 \right)$$

  - 由于 w_{ij} 非负，要求 $(f_i - f_j)^2$ 项必须等于零，**这就意味着 f_i 必须等于 f_j 。**

- **证明（续）**
  - 进一步，由于图是连通的，根据图论相关知识，一定**存在一条路径将所有的顶点连接起来**。这样，f_i 与 f_j 的相等关系就得以在整个路径上传播。所以 $\mathbf{f}$ 向量的所有分量均相等。这就意味着 $\mathbf{f}$ 是一个分量全为1的特征向量(只差一个任意的系数）。它可以构成特征向量空间的基。
  - 现在需要证明的问题，有没有第二重特征向量，其对应的特征值为零，但其分量并不全相等。
    - 反证法：假如存在两个分量不相等，但是，根据上面同样的分析，由于此时 $\mathbf{f}^T\mathbf{L}\mathbf{f}$ 仍然为零，在图连通的情形下必须推导出这两个分量相等。这就是一个矛盾。所以特征值零并不存在第二特征向量。



Red is a path

证明（续）

- 接下来证明 $k > 1$的情形，即连接子图多于一个。
- 不失一般性，假定样本点均按连通子图逐个排序。这样，由于连通子图之间不存在边相连，所以图G 的拉普拉斯矩阵具有分块连通的结构：

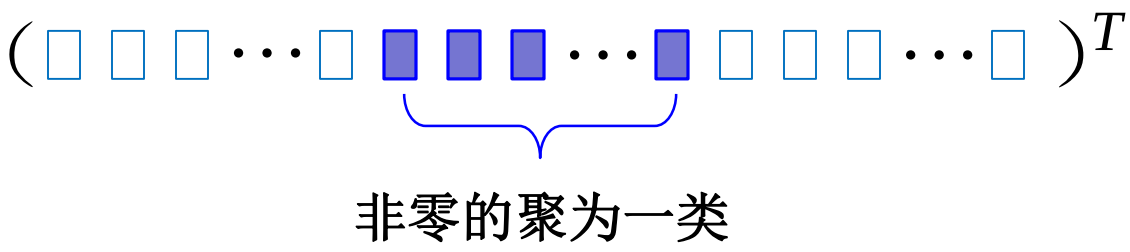
$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_1 & & & \\ & \mathbf{L}_2 & & \\ & & \ddots & \\ & & & \mathbf{L}_k \end{pmatrix}$$

- 且每一个 $\mathbf{L}_i$ 均为一个拉普拉斯矩阵，对应一个连通的子图。所以我们一共可以构造 k 个非零特征向量 (均为特征值0所对应的），而且不可能构造多于 k 个非零特征向量使特征向量空间的基大于 k:

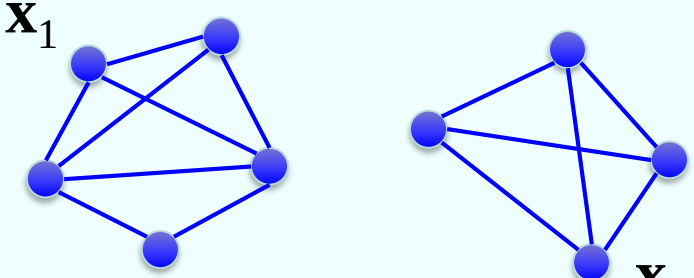
$$\mathbf{e}_{A_1} = [1,1,\cdots,1,0,0,\cdots,0]^T \in R^n \quad \cdots$$
$$\mathbf{e}_{A_2} = [0,0,\cdots,0,1,1,\cdots,1,0,0,\cdots,0]^T \in R^n,$$
$$\mathbf{e}_{A_k} = [0,0,\cdots,0,1,1,\cdots,1]^T \in R^n$$

5.7.4 拉普拉斯矩阵的性质

- 拉普拉斯矩阵性质引出的重要结论：
  - 如果图 G 具有 k 个连通子图，若每个连通子图为一个聚类，那么采用其拉普拉斯矩阵的零特征值对应的特征向量可以将这些子图分离开来。
  - 这是因为这些特征值对应的特征向量具有分块非零等值的结构。因此可以自然地将数据点分开。
  - 因此，求解 **L** 矩阵零特征值对应的特征向量，这正是聚类所期待的。

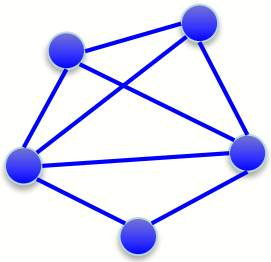


进一步解释：

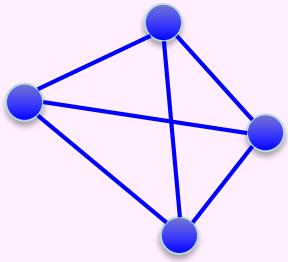


$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_1 & \\ & \mathbf{L}_2 \end{pmatrix}$$

$$\mathbf{L}\begin{pmatrix}1\\1\\1\\1\\1\\0\\0\\0\\0\end{pmatrix} = \mathbf{0}, \text{ 且有 } \mathbf{L}\begin{pmatrix}0\\0\\0\\0\\0\\1\\1\\1\\1\end{pmatrix} = \mathbf{0}.$$



$$\mathbf{L}_1 = \mathbf{D}_1 - \mathbf{W}_1 \in R^{5\times5}, \quad \mathbf{L}_1\begin{pmatrix}1\\1\\1\\1\\1\end{pmatrix} = \mathbf{0},$$



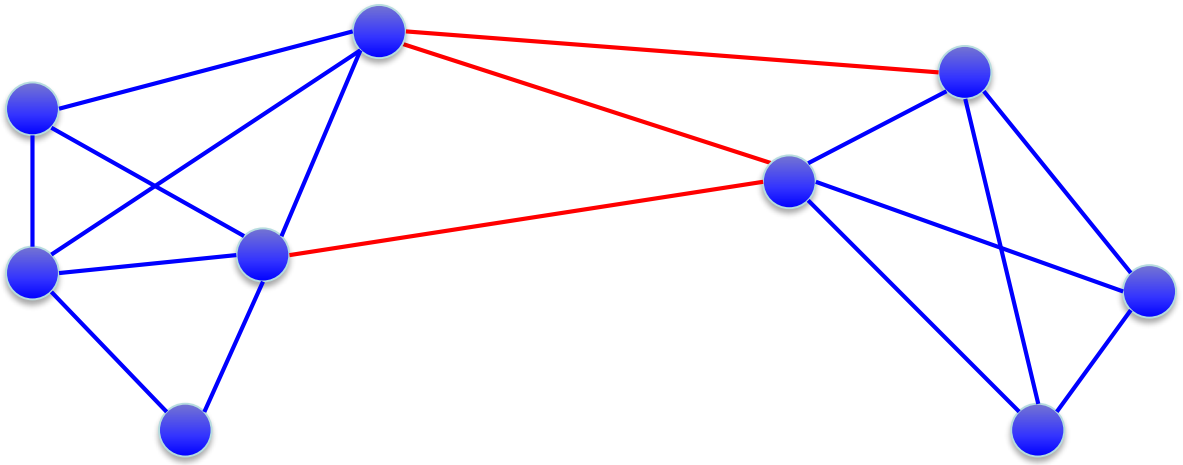
$$\mathbf{L}_2 = \mathbf{D}_2 - \mathbf{W}_2 \in R^{4\times4}, \quad \mathbf{L}_2\begin{pmatrix}1\\1\\1\\1\end{pmatrix} = \mathbf{0}$$

$$\begin{matrix}\mathbf{x}_1 \to \\ \mathbf{x}_2 \to \\ \\ \vdots \\ \\ \mathbf{x}_6 \to \\ \\ \vdots \\ \mathbf{x}_9 \to\end{matrix}\begin{pmatrix}1 & 0\\1 & 0\\1 & 0\\1 & 0\\1 & 0\\0 & 1\\0 & 1\\0 & 1\\0 & 1\end{pmatrix}$$

如果以该矩阵的每一行为样本点的新特征，则直接可以得出聚类结果！

5.7.4 拉普拉斯矩阵的性质

- 但是，实际应用中，数据簇之间并非是完全分离的。这就是说，图可能仍然是连通的。在此情形下，自然地，**可以考察拉普拉斯矩阵最小的特征值对应的特征向量，并由这些特征向量组成新的特征空间。**



5.7.5 谱聚类算法

- **谱聚类技术路线**
  - 图的连通子图与 L 矩阵特征值的关系：
    - 设 G 为一个具有非负连接权重的无向图，由图 G 导出的拉普拉斯矩阵 L 的零特征值的重数等于图 G 的连通子图的个数。
  - **该定理告诉我们：**
    - 需要考察 L 矩阵零特征值对应的特征向量。
    - 实际中，数据簇之间可能相互混杂、重叠，所以 L 矩阵通常并不具有分块形状（无论怎样调整样本顺序）。因此，可以考察其最小的几个特征值对应的特征向量。

5.7.5 谱聚类算法

- **谱聚类技术路线**
  - 一旦拉普拉斯矩阵得到构造，由其最小的几个特征对应的特征向量所构成的空间就得到确定。因此，**构造拉普拉斯矩阵是至关重要的一步。**
  - 构造拉普拉斯矩阵本质上取决于**对数据图的描述，即图构造。**

5.7.5 谱聚类算法

- **归一化图拉普拉斯** (Graph Laplacian)
  - 有两种构造归一化图拉普拉斯矩阵的方法
    - 对称型：

$$\mathbf{L}_{sym} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$$

    - 随机游走型 (random walk)：

$$\mathbf{L}_{rw} = \mathbf{D}^{-1}\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}$$

5.7.5 谱聚类算法

- **谱聚类算法**
  - **根据不同的图拉普拉斯构造方法，可以得到不同的谱聚类算法形式。**
  - **但是，这些算法的核心步骤都是相同的：**
    - 利用点对之间的相似性，构建亲和度矩阵；
    - 构建拉普拉斯矩阵；
    - 求解拉普拉斯矩阵最小的特征值对应的特征向量（通常舍弃零特征所对应的分量全相等的特征向量）；
    - 由这些特征向量构成样本点的新特征，采用K-means等聚类方法完成最后的聚类。

Un-normalized (classical) Spectral Clustering—Algorithm 1

1 input: similarity matrix $\mathbf{W}$, number k of clusters

2 compute the un-normalized Laplacian matrix $\mathbf{L=D-W}$

3 compute the first k eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$ of the $\mathbf{L}$

4 let $\mathbf{U} \in \mathrm{R}^{n \times k}$ be the matrix containing the vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$, namely, $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k] \in \mathrm{R}^{n \times k}$

5 for $i = 1, 2, \ldots, n$, let $\mathbf{y}_i \in \mathrm{R}^k$ be the vector corresponding to the i-th row of $\mathbf{U}$.

6 cluster the points $\{\mathbf{y}_i\}_{i=1,\ldots,n}$ in R^k with k-means algorithm into clusters $\mathrm{A}_1, \mathrm{A}_2, \ldots, \mathrm{A}_k$

7 output $\mathrm{A}_1, \mathrm{A}_2, \ldots, \mathrm{A}_k$.

Normalized Spectral Clustering—Algorithm 2 (Shi 算法）

1 input: similarity matrix $\mathbf{W}$, number k of clusters

2 compute the unnormalized Laplacian matrix $\mathbf{L=D-W}$

3 compute the first k generalized eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$ of the generalized eigen-problem $\mathbf{Lu} = \lambda \mathbf{Du}$

4 Let $\mathbf{U} \in R^{n \times k}$ be the matrix containing the vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$, namely, $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k] \in R^{n \times k}$

5 for $i = 1, 2, \ldots, n$, let $\mathbf{y}_i \in R^k$ be the vector corresponding to the i–th row of $\mathbf{U}$.

6 cluster the points $\{\mathbf{y}_i\}_{i=1,\ldots,n}$ in R^k with k-means algorithm into clusters $A_1 , A_2 , \ldots, A_k$

7 output $A_1 , A_2 , \ldots, A_k$.

J. Shi, and J. Malik. Normalized cuts and image segmentation. TPAMI 2000

Normalized Spectral Clustering—Algorithm 3 (Ng算法)

1 input: similarity matrix $\mathbf{W}$, number k of clusters

2 compute $\mathbf{L}_{sym} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$

3 compute the first k eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$ of $\mathbf{L}_{sym}$

4 Let $\mathbf{U} \in R^{n\times k}$ be the matrix containing the vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$, namely, $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k] \in R^{n\times k}$

5 form the matrix $\mathbf{T} \in R^{n\times k}$ from $\mathbf{U}$ by normalizing the rows to norm 1, namely, set $t_{ij} = u_{ij} \Big/ \sqrt{\sum_{m=1}^{n} u_{im}^2}$

6 for $i = 1, 2, \ldots, n$, let $\mathbf{y}_i \in R^k$ be the vector corresponding to the i–th row of $\mathbf{T}$.

7 cluster the points $\{\mathbf{y}_i\}_{i=1,\ldots,n}$ in R^k with k-means algorithm into clusters A_1 , A_2 , …, A_k

8 output A_1 , A_2 , …, A_k .

On spectral clustering: analysis and an algorithm, NIPS, 2002.

5.7.5 谱聚类算法

- **解释**
  - 上述三个算法的核心都是求解一个类似的学习模型：
  - 算法1 (classical)

$$\min_{\mathbf{H}\in R^{n\times k}} tr\left(\mathbf{H}^T\mathbf{L}\mathbf{H}\right), \quad s.t. \quad \mathbf{H}^T\mathbf{H}=\mathbf{I}$$

  - 算法2 (Ncut)

$$\min_{\mathbf{T}\in R^{n\times k}} tr\left(\mathbf{T}^T\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}\mathbf{T}\right), \quad s.t. \ \mathbf{T}^T\mathbf{T}=\mathbf{I} \quad + \quad \mathbf{H}=\mathbf{D}^{-1/2}\mathbf{T}$$

  - 算法3 (Ng's Algorithm)

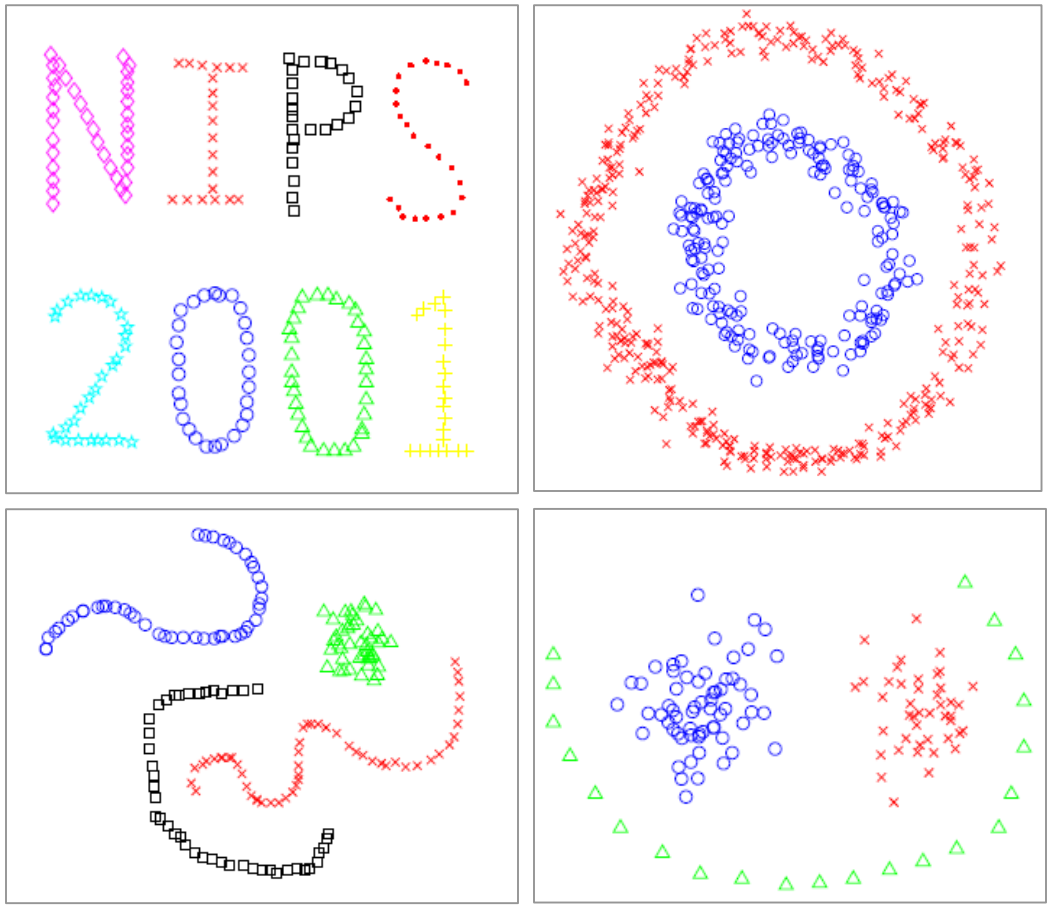
$$\min_{\mathbf{H}\in R^{n\times k}} tr\left(\mathbf{H}^T\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}\mathbf{H}\right), \quad s.t. \ \mathbf{H}^T\mathbf{H}=\mathbf{I}$$

5.7.5 谱聚类算法

- **算法细节**
  - 核心问题是图构造
    - **局部连接 k 近邻 (ε-半径) 取多大？**
    - **点对权值如何计算？**
  - 特征值分解问题：对于超大型矩阵，计算仍然不稳定，可能会引起结果很差。
  - 最后采用 K-means 聚类问题，也可能会影响聚类结果。
  - 当然，聚类数目的多少是一个open problem。

5.7.5 谱聚类算法

- 一些例子



A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: analysis and an algorithm. NIPS, pp. 849-856, 2002.

5.7.5 谱聚类算法

- 采用谱聚类将图像的前景目标分割出来

如何构造近邻图？

Thank All of You!

(Questions?)

向世明

smxiang@nlpr.ia.ac.cn

http://www.escience.cn/people/smxiang

时空数据分析与学习课题组(STDAL)

中科院自动化研究所· 模式识别国家重点实验室