第8章第2讲
数据聚类
Data Clustering

向 世 明
smxiang@nlpr.ia.ac.cn

助教：杨学行(xhyang@nlpr.ia.ac.cn); 吴一超(yichao.wu@nlpr.ia.ac.cn)

8.9 分级聚类(Hierarchical Clustering)

- 生物学上的物种分类
  - 门、纲、目、科、属、种
  - 最相似的物种被分为“种”
  - 这种分层次归类方法对生物学研究发挥着巨大的作用
    - 生物学意义
    - 生物学研究意义：解决分岐、发现新的物种。
  - 这种思想也可以自然地应用到聚类分析之中，称为**分级聚类、层次聚类**或者**系统聚类**。

8.9 分级聚类

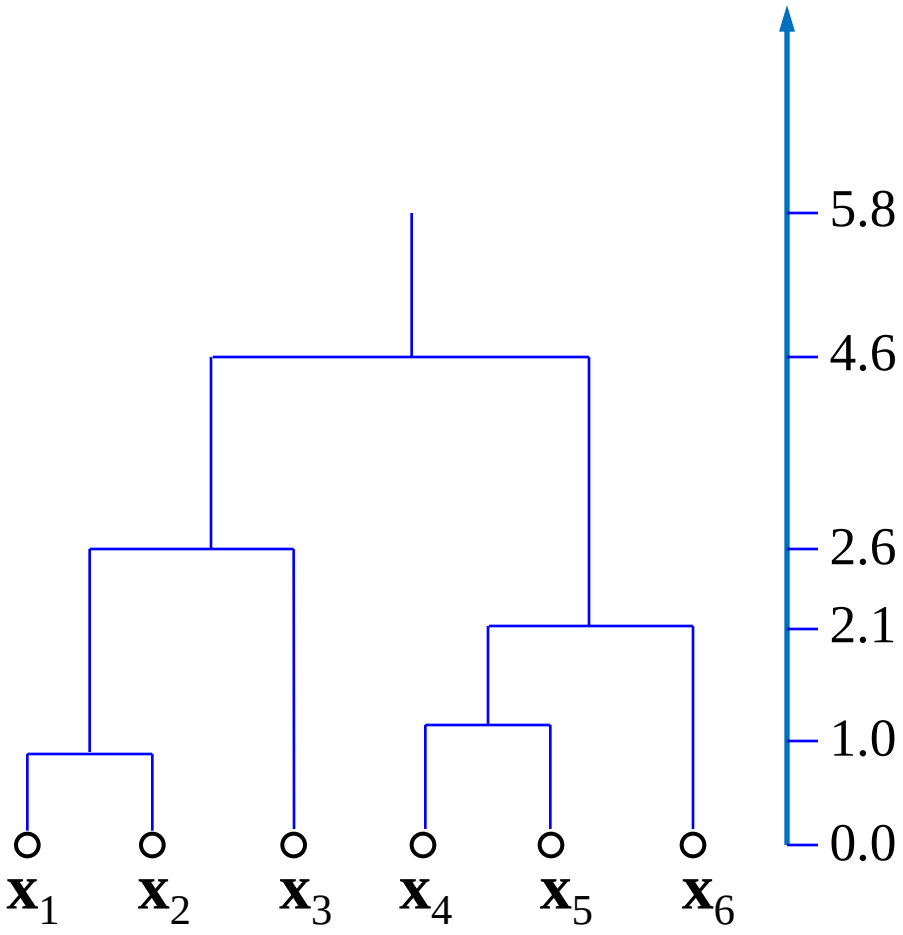
- 分级聚类思想
  - 对于 n 个样本，极端的情况下，最多可以将数据分成 n 类；最少可以只分成一类，即全部样本都归为一类。
  - **凝聚的层次聚类（自底向上）**
    - 将每个样本作为一个簇，然后根据给定的规则**逐渐合并**一些样本，形成更大的簇，直到所有的样本都被分到一个合适的簇中。
  - **分裂的层次聚类（自顶向下）**
    - 将所有的样本置于一个簇中，然后根据给定的规则**逐渐细分**样本，得到越来越小的簇，直到某个终止条件得到满足。

8.9 分级聚类

- **自底向上的分级聚类步骤**
  - (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两个类之中。

8.9 分级聚类

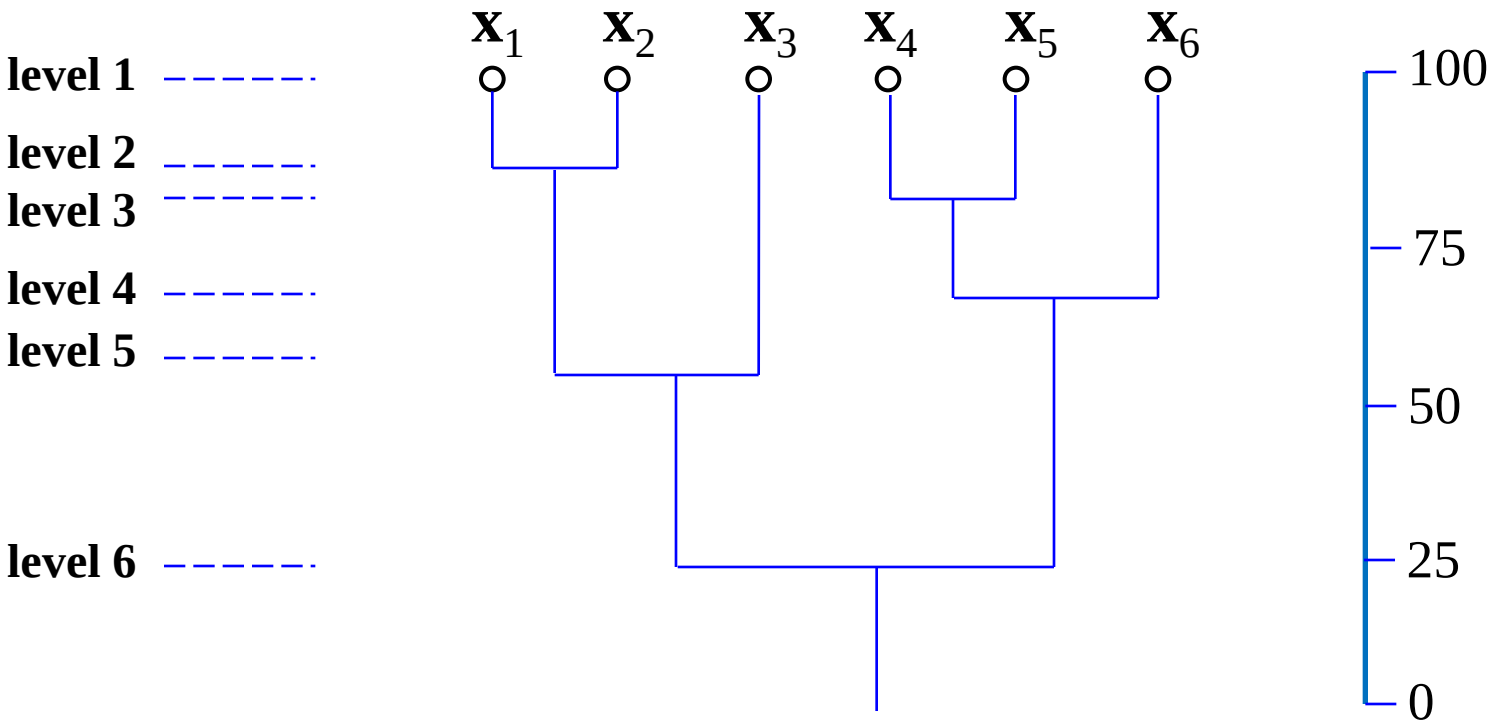
- 系统树
  - 分级聚类是一种典型的**系统聚类法**。用一棵树来描述分级聚类结果，称为**聚类树**(dendrogram)，或**系统树图**。
  - 如右图，最底层的每个节点表示一个样本，**采用树枝连接两个合并的样本**，树枝的长度反映两个节点之间的距离（或相似性）。
  - 为方便，**采用“水平”来表示分级聚类过程的不同阶段。**开始为水平1，每个样本为一类，然后每执行一次“合并”增加一个水平。



系统树图（采用距离）

8.9 分级聚类

- **系统树：** 分级聚类的另一种表示



系统树图（采用相似性）

8.9 分级聚类

- **分级聚类两个核心问题：**
  - 如何度量样本之间的距离或相似性
  - 如何度量两个簇之间的距离或者相似性
- **样本之间的距离与相似性：**
  - 欧氏距离、马式距离、城区距离、匹配距离、…
  - 相关系数、高斯相似性函数、余弦、距离倒数、…

8.9 分级聚类

- 簇与簇之间的距离

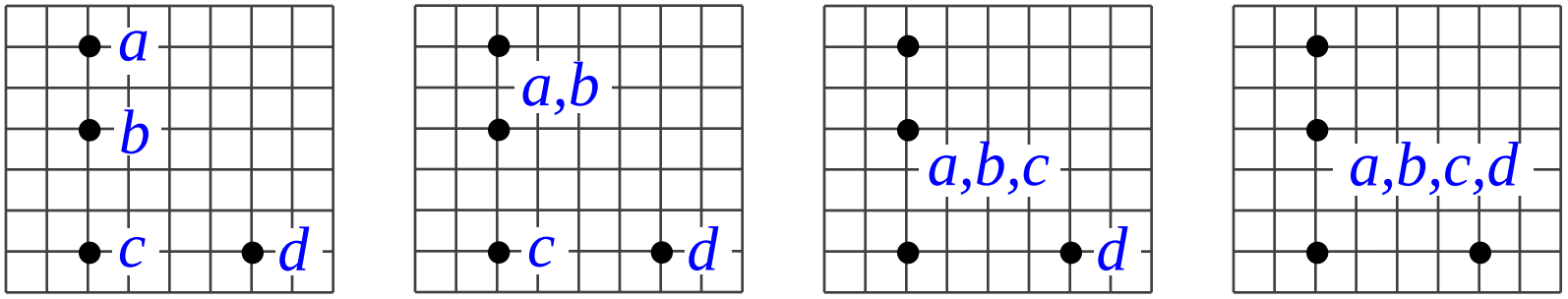
$d_{\min}(D_i, D_j) = \min_{\substack{\mathbf{x}\in D_i \\ \mathbf{x}'\in D_j}} \|\mathbf{x}-\mathbf{x}'\|$	最小距离
$d_{\max}(D_i, D_j) = \max_{\substack{\mathbf{x}\in D_i \\ \mathbf{x}'\in D_j}} \|\mathbf{x}-\mathbf{x}'\|$	最大距离
$d_{avg}(D_i, D_j) = \frac{1}{n_i n_j} \sum_{\mathbf{x}\in D_i} \sum_{\mathbf{x}'\in D_j} \|\mathbf{x}-\mathbf{x}'\|$	平均距离
$d_{mean}(D_i, D_j) = \|\mathbf{m}_i - \mathbf{m}_j\|$	中心距离

8.9 分级聚类

- 实践
  - 根据数据特性、聚类目标的不同，通常需要采用不同的簇间距离。
  - 对同一数据集，采用不同的簇间连接通常会得到不同的聚类结果。

- 例子1：采用最近距离连接簇

(注：*a*到*d*之间的距离取整数）



	b	c	d
a	2	5	6
b		3	5
c			4

	b	c	d
a	2	5	6
b		3	5
c			4

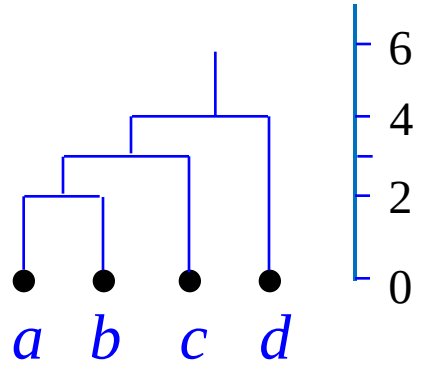
	c	d
a,b	3	5
c		4

	d
a,b,c	4

	c	d
a,b	3	5
c		4

	d
a,b,c	4

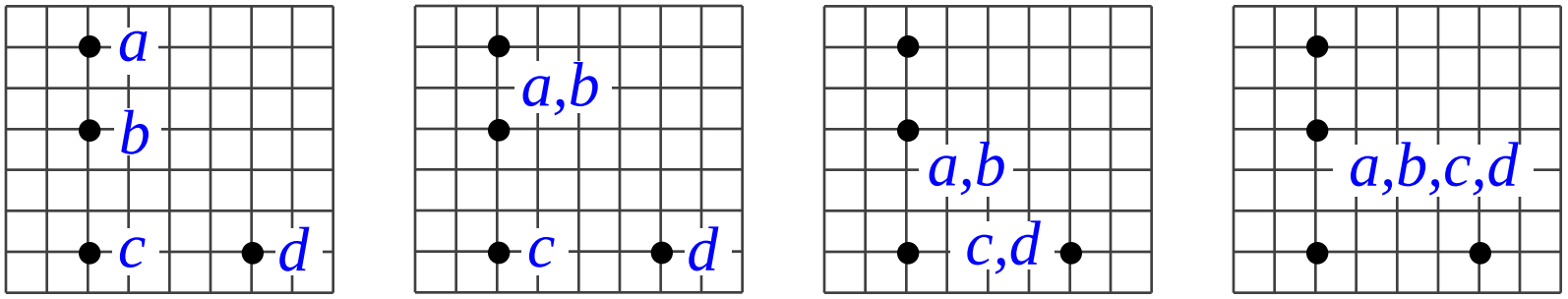
重新计算距离



簇与样本之间的距离采用最近距离

- **例子2：采用最远距离连接簇** (最大最小准则)

(注：*a*到*d*之间的距离取整数)



	b	c	d
a	2	5	6
b		3	5
c			4

	b	c	d
a	2	5	6
b		3	5
c			4

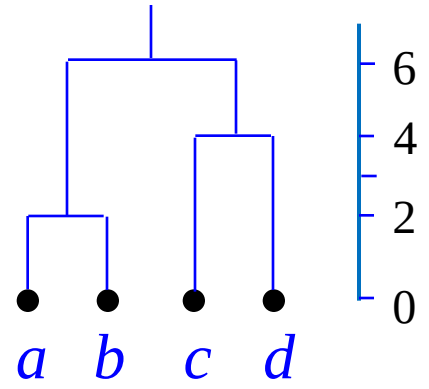
	c	d
a,b	5	6
c		4

	c,d
a,b	6

	c	d
a,b	5	6
c		4

	c,d
a,b	6

重新计算距离



簇与样本之间的距离采用最远距离

- 例子3：请按最小距离准则对如下6个样本进行分级聚类：

$$\mathbf{x}_1 = (0, 3, 1, 2, 0)$$
$$\mathbf{x}_2 = (1, 3, 0, 1, 0)$$
$$\mathbf{x}_3 = (3, 3, 0, 0, 1)$$
$$\mathbf{x}_4 = (1, 1, 0, 2, 0)$$
$$\mathbf{x}_5 = (3, 2, 1, 2, 1)$$
$$\mathbf{x}_6 = (4, 1, 1, 1, 0)$$

解：将每个样本单独看成一类，计算各点对之间的距离，见下表：

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$
$\mathbf{x}_1$	0					
$\mathbf{x}_2$	$\sqrt{3}$	0				
$\mathbf{x}_3$	$\sqrt{15}$	$\sqrt{6}$	0			
$\mathbf{x}_4$	$\sqrt{6}$	$\sqrt{5}$	$\sqrt{13}$	0		
$\mathbf{x}_5$	$\sqrt{11}$	$\sqrt{8}$	$\sqrt{6}$	$\sqrt{7}$	0	
$\mathbf{x}_6$	$\sqrt{21}$	$\sqrt{14}$	$\sqrt{8}$	$\sqrt{11}$	$\sqrt{4}$	0

解 (续):

基于上述矩阵，根据最小距离准则，应将 $\mathbf{x}_1$和$\mathbf{x}_2$合并为一类，得到 $G_1=\{\mathbf{x}_1,\mathbf{x}_2\}$，$\{\mathbf{x}_3\}$，$\{\mathbf{x}_4\}$，$\{\mathbf{x}_5\}$，$\{\mathbf{x}_6\}$。**按最小距离原则**重新计算各类之间的距离，见下表：

	G_1	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$
G_1	0				
$\mathbf{x}_3$	$\sqrt{6}$	0			
$\mathbf{x}_4$	$\sqrt{5}$	$\sqrt{13}$	0		
$\mathbf{x}_5$	$\sqrt{8}$	$\sqrt{6}$	$\sqrt{7}$	0	
$\mathbf{x}_6$	$\sqrt{14}$	$\sqrt{8}$	$\sqrt{11}$	$\sqrt{4}$	0

解释：比如，类G_1与$\{\mathbf{x}_3\}$之间的距离，按最小距离准则，计算为$\mathbf{x}_2$与$\mathbf{x}_3$的距离

解 (续):

基于上述矩阵，根据最小距离准则，应将 $\mathbf{x}_5$和$\mathbf{x}_6$合并为一类，得到 $G_1=\{\mathbf{x}_1,\mathbf{x}_2\}$，$\{\mathbf{x}_3\}$，$\{\mathbf{x}_4\}$，$G_2=\{\mathbf{x}_5,\mathbf{x}_6\}$。**按最小距离原则**重新计算各类之间的距离，见下表：

	G_1	$\mathbf{x}_3$	$\mathbf{x}_4$	G_2
G_1	0			
$\mathbf{x}_3$	$\sqrt{6}$	0		
$\mathbf{x}_4$	$\sqrt{5}$	$\sqrt{13}$	0	
G_2	$\sqrt{8}$	$\sqrt{6}$	$\sqrt{7}$	0

解释：比如，类G_1与类G_2之间的距离，按最小距离准则，计算为$\mathbf{x}_2$与$\mathbf{x}_5$的距离

解 (续)：

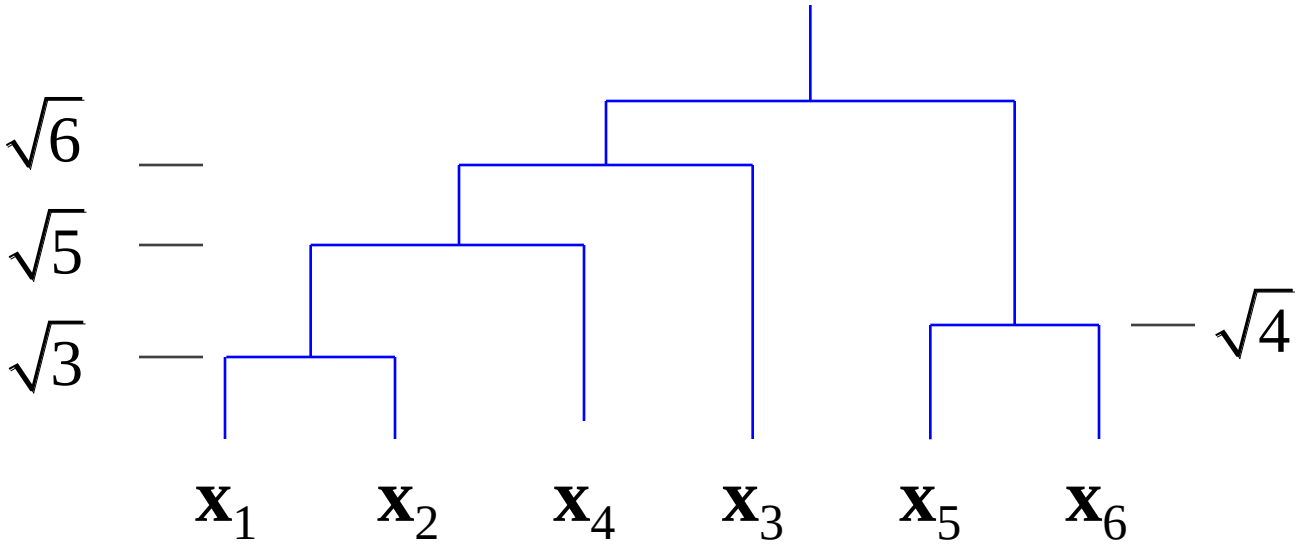
基于上述矩阵，根据最小距离准则，应将 G_1和$\mathbf{x}_3$合并为一类，得到 $G_3=G_1\cup\{\mathbf{x}_4\}$，$\{\mathbf{x}_3\}$，$G_2=\{\mathbf{x}_5,\mathbf{x}_6\}$。**按最小距离原则**重新计算各类之间的距离，见下表：

	G_3	$\mathbf{x}_3$	G_2
G_3	0		
$\mathbf{x}_3$	$\sqrt{6}$	0	
G_2	$\sqrt{7}$	$\sqrt{6}$	0

解释：比如，类G_3与$\{\mathbf{x}_3\}$之间的距离，按最小距离准则，计算为$\mathbf{x}_2$与$\mathbf{x}_3$的距离

解 (续):

基于上述矩阵，根据最小距离准则，应将 G_3和$\mathbf{x}_3$合并为一类，当然也可以将 G_2 和 $\mathbf{x}_3$合并为一类。如选择前者，得到 $G_4=G_3 \cup \{\mathbf{x}_3\}$，$G_2=\{\mathbf{x}_5,\mathbf{x}_6\}$。最后，$G_3$和$G_2$ 合并为一类，这样所有数据将被合并在一起。



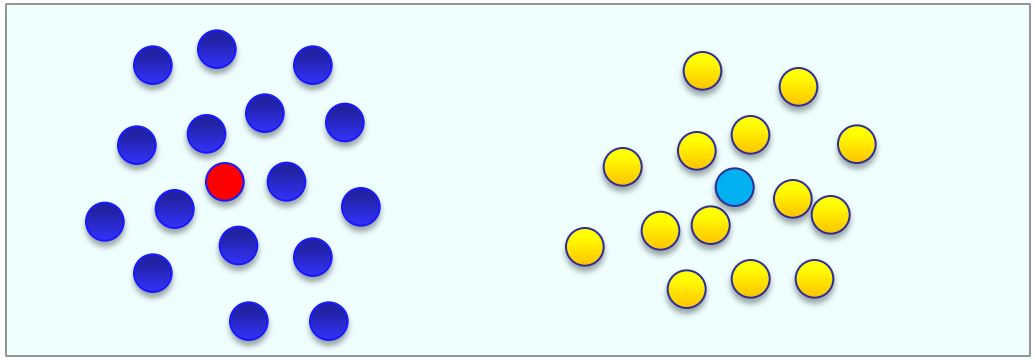
系统树图

第10节　谱聚类

(Spectral Clustering)

8.10.1 引言

- 回忆K-means聚类

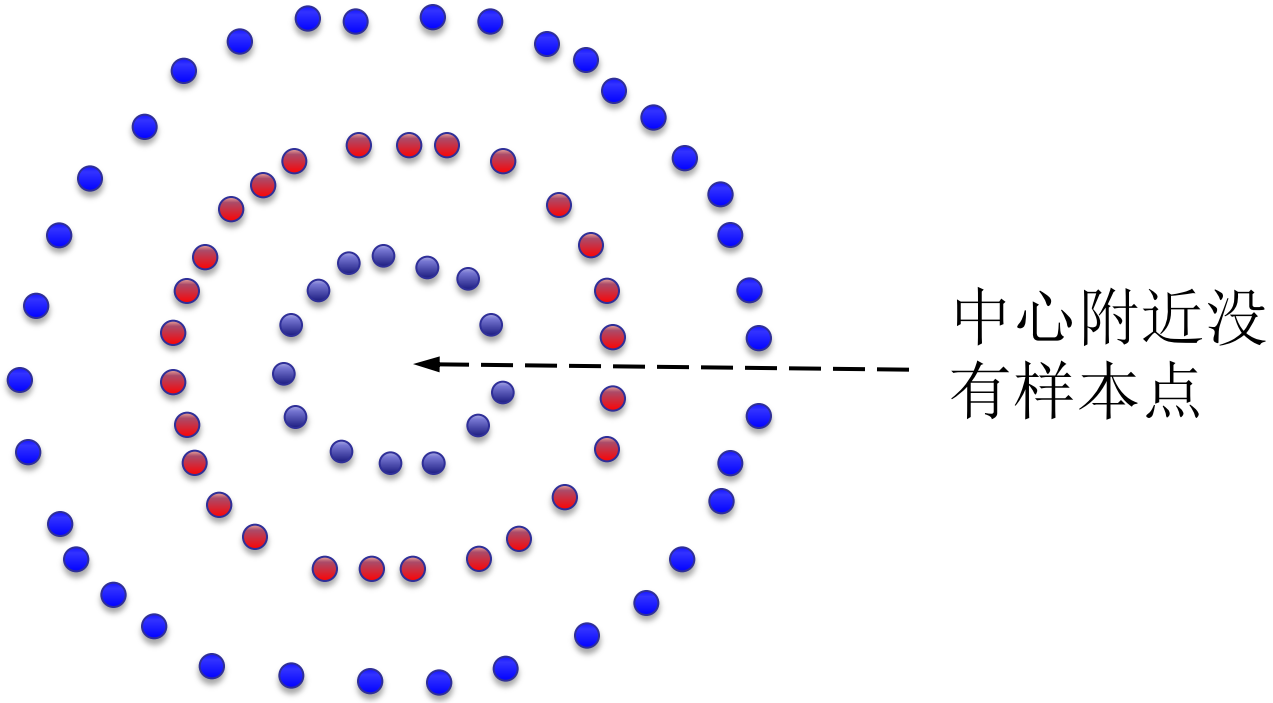


每个簇均可以用中心点来表示

(特别适合于单个簇符合高斯分布的情形)

8.10.1 引言

- 对于其它分布情形



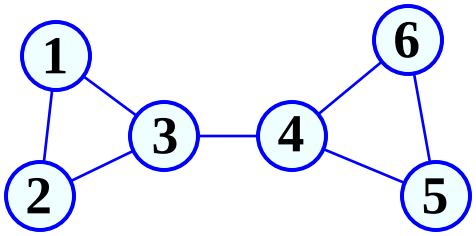
Spectral clustering allows us to address these sorts of clusters!

8.10.1 引言

- **谱学习方法**
  - 广义上讲，任何在学习过程中应用到矩阵特征值分解的方法均叫**谱学习方法**，比如主成分分析（PCA）、线性判别成分分析（LDA）、流形学习中的谱嵌入方法、谱聚类、等等。
- **谱聚类**
  - 谱聚类算法建立在**图论的谱图理论**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。

8.10.2 图论基本概念

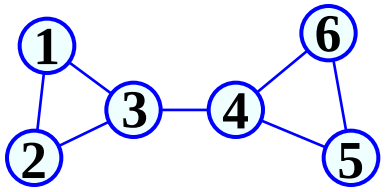
- **图 G：** 由顶点集 V 和边集 E 所构成，记为 G(V,E)。根据边是否有向，可以分为无向图或者有向图。
- **图 G 的邻接矩阵 W：**
  - 行数和列数等于矩阵顶点的个数；
  - 矩阵元素为 0 或 1。1 表示对应的一对顶点有边相连，0表示没有边相连。



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

8.10.2 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**。

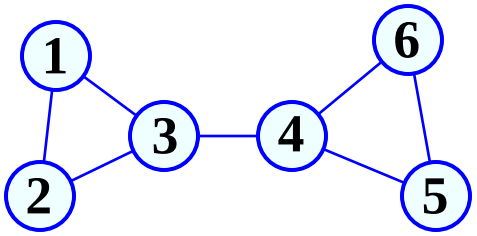


$$\mathbf{W}=\begin{pmatrix}0&1&1&0&0&0\\1&0&1&0&0&0\\1&1&0&1&0&0\\0&0&1&0&1&1\\0&0&0&1&0&1\\0&0&0&1&1&0\end{pmatrix},\quad \mathbf{D}=\begin{pmatrix}2&&&&&\\&2&&&&\\&&3&&&\\&&&3&&\\&&&&2&\\&&&&&2\end{pmatrix}$$

8.10.2 图论基本概念

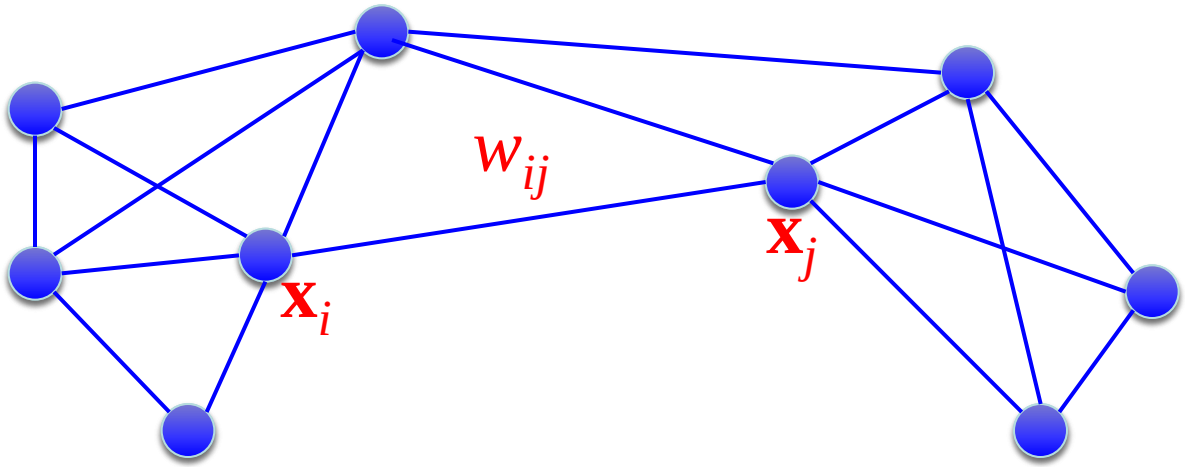
- 拉普拉斯矩阵
  - 度矩阵减去邻接矩阵得到拉普拉斯矩阵 **L**。

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



8.10.2 图论基本概念

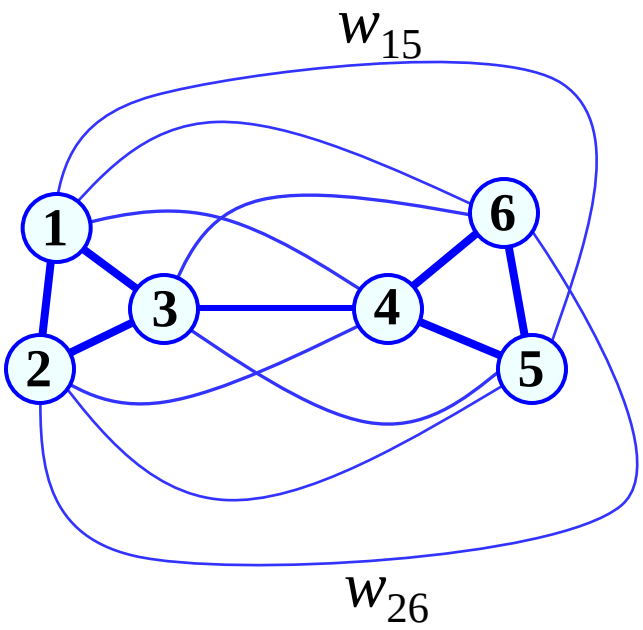
- **基于数据集的图构造**
  - 上述有关图的概念也可以用来描述数据点。
  - 将每一个数据点视为图的一个顶点，顶点之间可以有边相连。每条边上加上一些权重，用来反映点对亲和性（即相似性）。



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

8.10.2 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

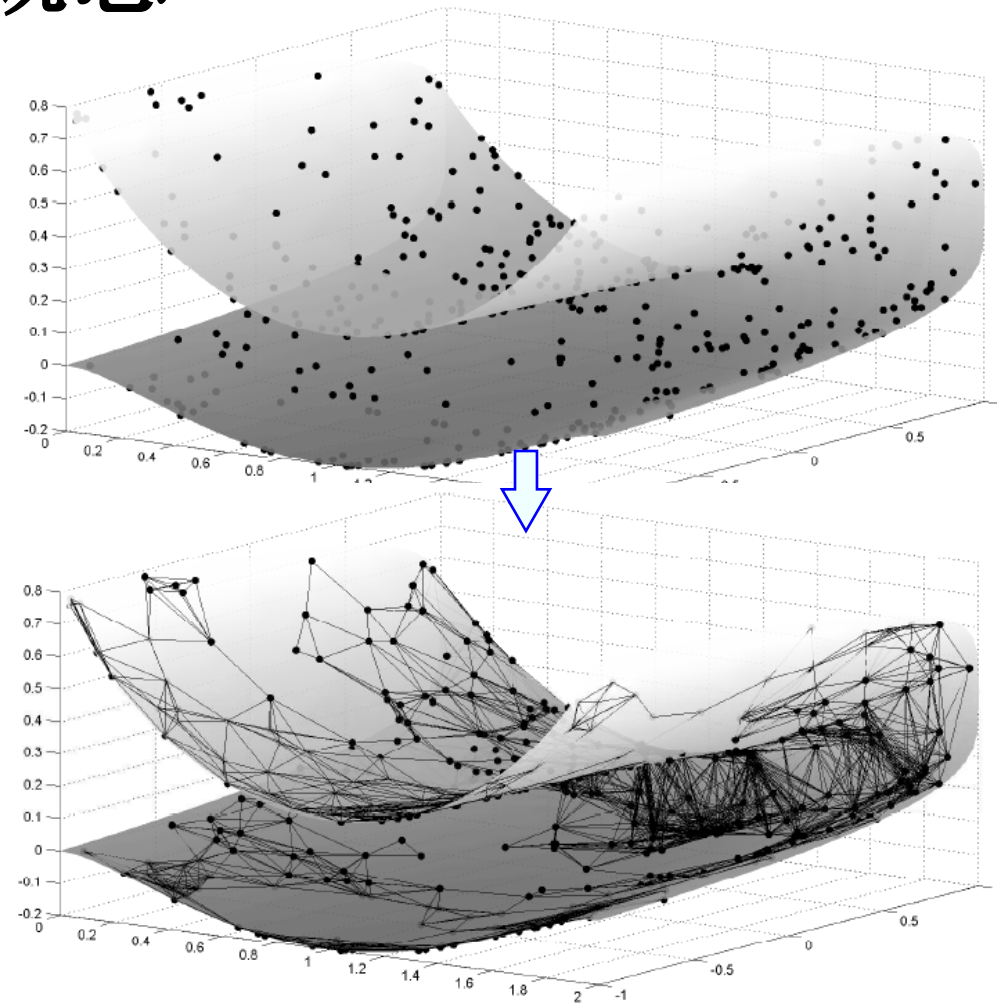


$$\begin{pmatrix} 0 & w_{12} & w_{13} & w_{14} & w_{15} & w_{16} \\ & 0 & w_{23} & w_{24} & w_{25} & w_{26} \\ & & 0 & w_{34} & w_{35} & w_{36} \\ & \text{对} & & 0 & w_{45} & w_{46} \\ & & & \text{称} & 0 & w_{56} \\ & & & & & 0 \end{pmatrix}$$

点对相似度矩阵

8.10.2 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

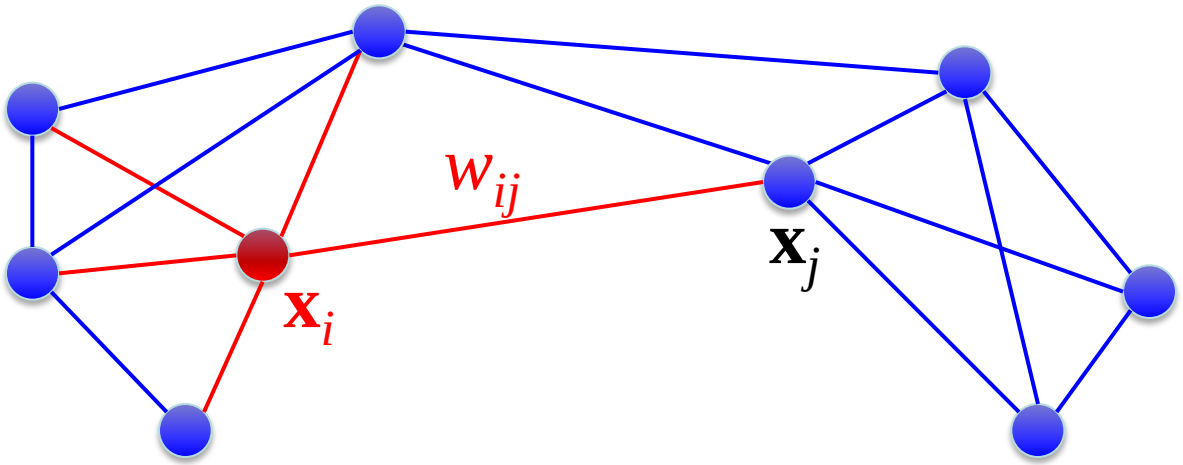


k-近邻：对每个数据点 $\mathbf{x}_i$，首先在所有样本中找出不包含 $\mathbf{x}_i$ 的 k 个最邻近的样本点，然后 $\mathbf{x}_i$ 与每个邻近样本点均有一条边相连，从而完成图构造。

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

8.10.2 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 $w_{ij}= 0$。）

8.10.2 图论基本概念

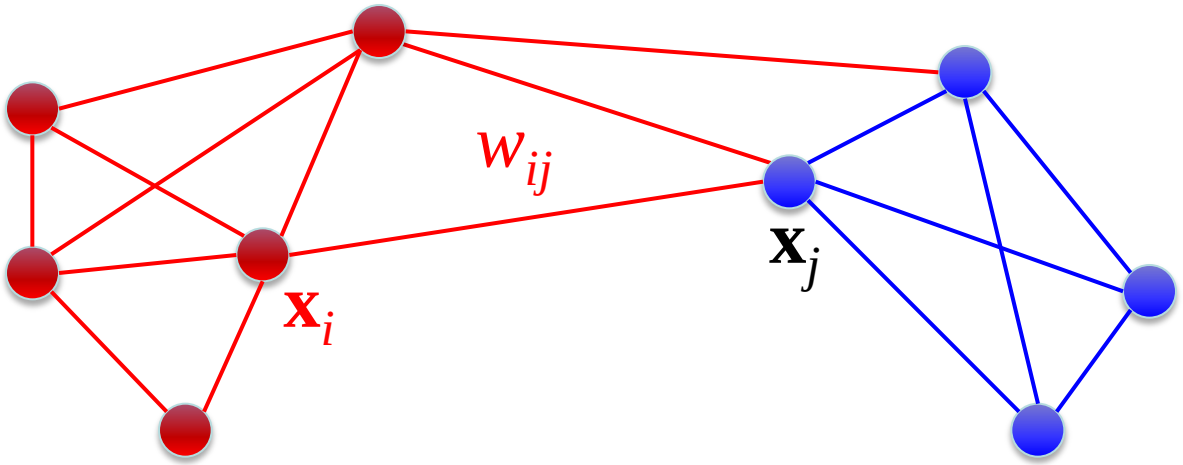
- **拉普拉斯矩阵(Laplacian matrix)**
  - 拉普拉斯矩阵是描述图的一种矩阵。给定一个具有 n 个顶点的图，其拉普拉斯矩阵描述为：

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

  - 其中，**D** 为一个对角矩阵，主对角元素表示顶点的度。**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）。

8.10.2 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。



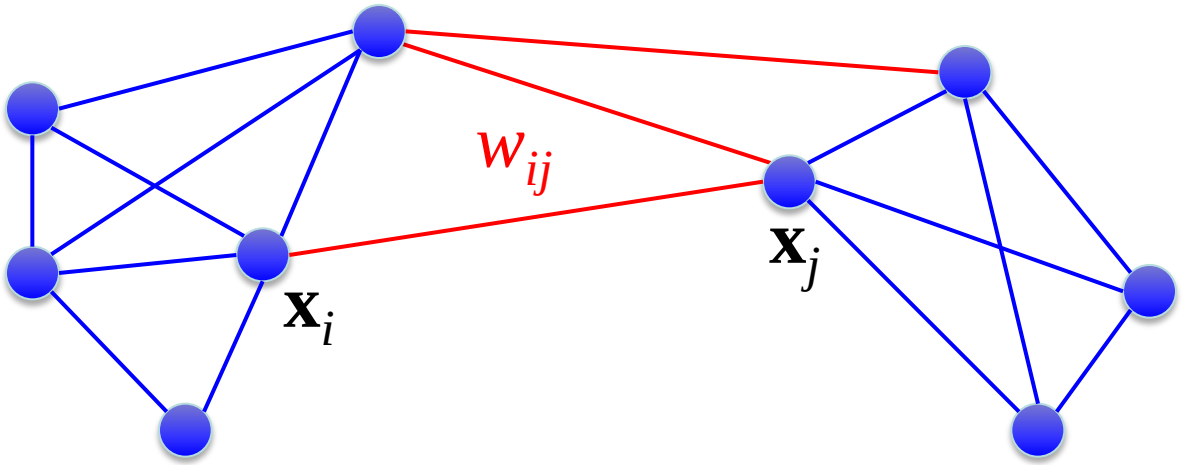
$$vol(A) = \sum_{i \in A} d_i$$

8.10.2 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。



8.10.2 图论基本概念

- **图切割：**
  - 设 $A_1, A_2, \ldots, A_k$ 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 $A_1, A_2, \ldots, A_k$ 为图 G 的一个分割。
  - **聚类指示向量：**
    - 设 $A_1, A_2, \ldots, A_k$ 为图 G 的 k 个连图子图，或为其 k 个划分，由每个 A_i 均可以定义一个 n 维指示向量，该向量对于 A_i 各顶点的元素值为 1，其余元素全为0：

如果顶点排好序

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

piecewise constant

8.10.2 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：

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

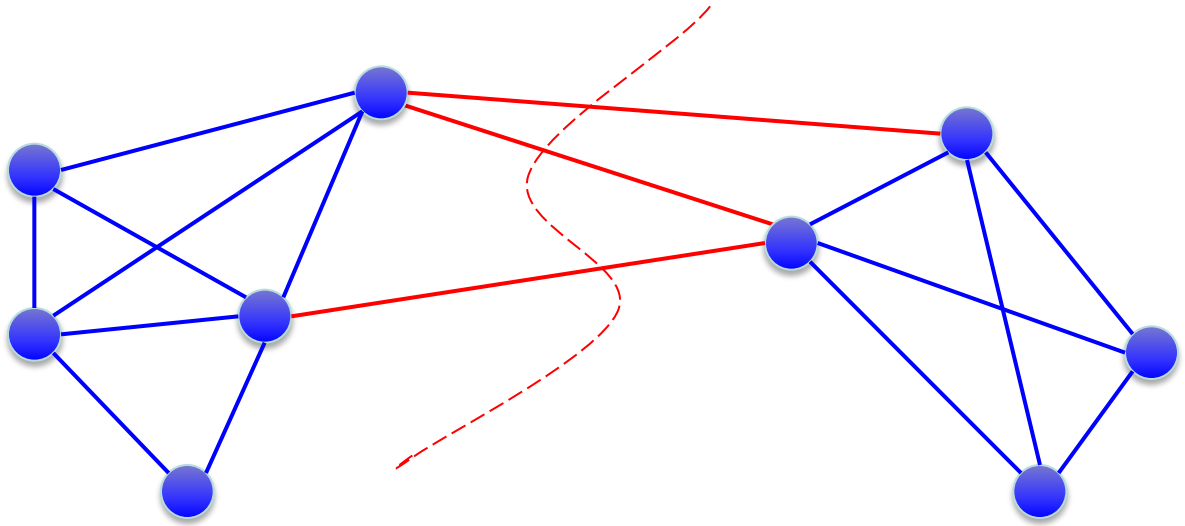
- **子图之间的切割：** **子图 A 与子图 B 的切割定义：**

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

注：如果两个顶点不相连，则权重为零。

8.10.3 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。



8.10.3 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：**

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

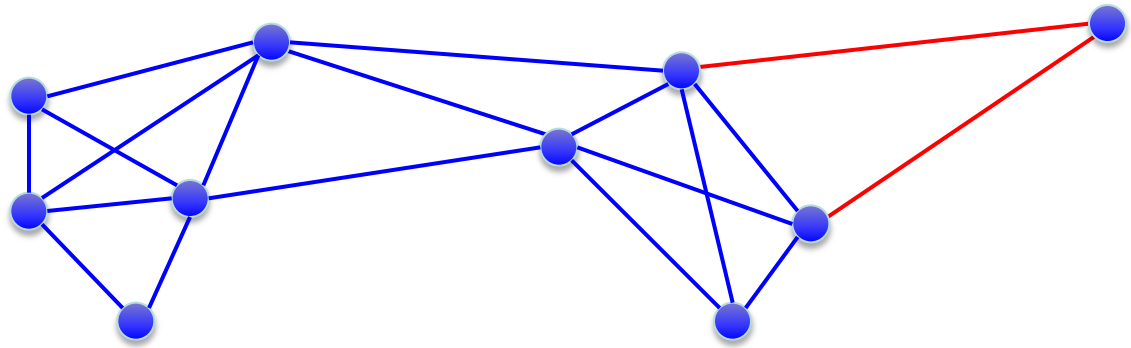
$$s.t. \quad A \neq \varnothing,$$

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

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

8.10.3 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。



8.10.3 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，即采用如下目标函数：
    - 采用子图的势：

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

    - 采用子图的体积：

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

8.10.3 图切割

- **K-切割 (k > 2)：**
  - 考虑将图分成 k 个子图：$A_1, A_2, \ldots, A_k$。一种直观的方法是将图切割问题理解为多个最小二分割问题的综合。
- **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：**

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

8.10.4 拉普拉斯矩阵的性质

- 拉普拉斯矩阵： $L = D - W$
- 性质：
  - L 的行和为零：
    - 因为$L=D-W$，D 的主对角元素为 W 各行元素之和。
  - 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：
    - 因为 $L*1=(D-W)*1=0=0*1$。
  - L 有 n 个非负的特征值， n 为图的顶点个数。

8.10.4 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：

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

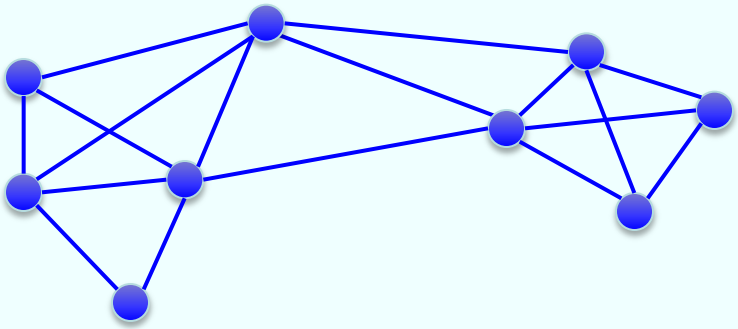
$$= \frac{1}{2}\left(\sum_{i=1}^{n} d_i f_i^2 - 2\sum_{i,j=1}^{n} f_i f_j w_{ij} + \sum_{j=1}^{n} d_j f_j^2 \right)$$

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

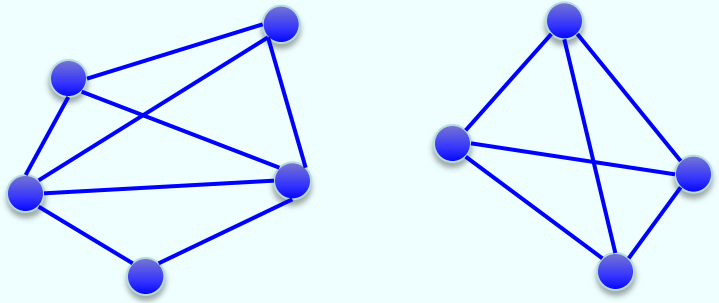
$$\geq 0$$

8.10.4 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*。**



零特征值的重数为1



零特征值的重数为2

8.10.4 拉普拉斯矩阵的性质

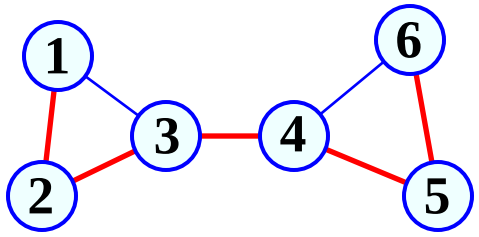
- 证明：
  - 首先考虑图 G 是连通的，即 $k=1$。对此情形，需要证明 **L** 矩阵只有一个特征值为 0，且对应的特征向量由元素全为 1 的向量所构成。
  - 假定 **f** 为特征值 0 对应的特征向量，于是有：

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

  - 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 **f** 向量的所有分量均相等。这就意味着 **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：

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

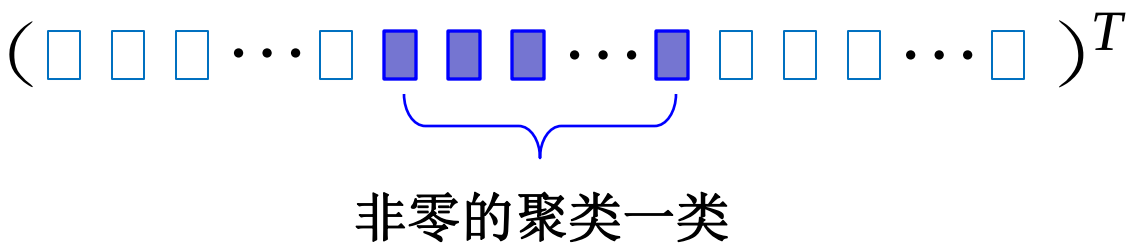
- 且每一个 $\mathbf{L}_i$ 均为一个拉普拉斯矩阵，对应一个连通的子图。所以我们一共可以构造 k 个非零特征向量 (均为特征值0所对应的），而且不可能构造多于 k 个非零特征向量使特征向量空间的基大于 k：

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

piecewise constant

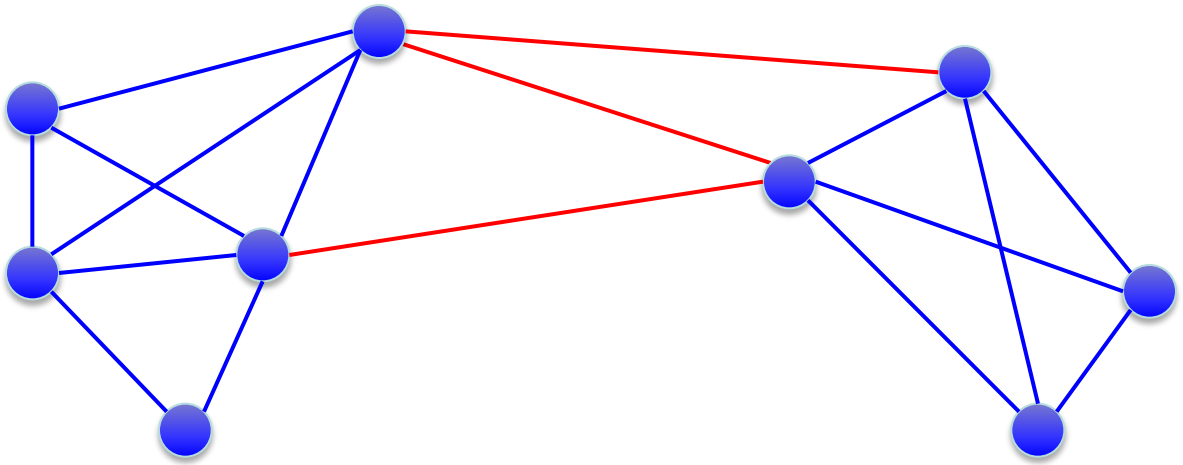
8.10.4 拉普拉斯矩阵的性质

- 该定理告诉我们一个重要的结论：
  - 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，这正是我们所期待的。



8.10.4 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。**



8.10.5 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。

8.10.5 谱聚类

- **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。

8.10.5 谱聚类

- **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**。

8.10.5 谱聚类

- 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)：

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

8.10.5 谱聚类

- 关于 $\mathbf{L}_{sym}$及 $\mathbf{L}_{rw}$，有如下性质：
  - 对任意的 $\mathbf{f} = [f_1, f_2, \ldots, f_n]^T \in \mathrm{R}^n$，有

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

  - 若 λ 为 $\mathbf{L}_{rw}$ 的特征值，且与其对应的特征向量为 $\mathbf{u}$，当且仅当 λ 为 $\mathbf{L}_{sym}$ 的一个特征值，$\mathbf{D}^{1/2}\mathbf{u}$为其对应的特征向量。
  - 若 λ 为 $\mathbf{L}_{rw}$ 的特征值，且与其对应的特征向量为 $\mathbf{u}$，当且仅当 $\mathbf{Lu} = \lambda \mathbf{Du}$。

8.10.5 谱聚类

- 关于$\mathbf{L}_{sym}$及$\mathbf{L}_{rw}$，有如下性质（续）：
  - $\mathbf{L}_{rw}$ 有一个零特征值且对应的特征向量分量全为 1，$\mathbf{L}_{sym}$ 也有一个零特征值且对应的特征向量为 $\mathbf{D}^{1/2}\mathbf{e}$，这里 $\mathbf{e}$ 为分量全为 1 的 n 维向量。
  - $\mathbf{L}_{sym}$ 及 $\mathbf{L}_{rw}$ 为半正定矩阵，特征值为正实数，且至少有一个为零。
  - 设 G 为一个具有非负连接权重的无向图，由图 G 导出的 $\mathbf{L}_{sym}$ 及 $\mathbf{L}_{rw}$ 的零特征值的重数等于图 G 的连通子图个数。
  - 设$A_1, A_2, \ldots, A_k$ 为 G 的 k 个连通成分，设 $\mathbf{e}_{A_i}$ 为对应于子图 A_i 为的指示向量，则 $\mathbf{e}_{A_i}$ 为 $\mathbf{L}_{rw}$零特征值对应的特征向量，$\mathbf{D}^{1/2}\mathbf{e}_{A_i}$ 为 $\mathbf{L}_{sym}$ 的一个特征向量。

8.10.5 谱聚类

- 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 A_1 , A_2 , …, A_k

7 output A_1 , A_2 , …, 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 $\mathbf{U} \in \mathrm{R}^{n\times k}$ 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 form the matrix $\mathbf{T} \in \mathrm{R}^{n\times k}$ 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 $\mathbf{y}_i \in \mathrm{R}^k$ 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 $\mathrm{A}_1, \mathrm{A}_2, \ldots, \mathrm{A}_k$

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

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

8.10.5 谱聚类

- **解释**
  - 算法2中，其广义特征值分解与 $\mathbf{L}_{rw}$ 的特征值分解所得特征向量空间相同。
  - 上述三个算法的核心是将原始的数据点 $\mathbf{x}_i$ 转换为在特征空间的数据点 $\mathbf{y}_i$，在新的空间对原始数据进行描述。
  - 通常来讲，normalized spectral clustering algorithms are better than the unnormalized one.

8.10.5 谱聚类

- 解释 **(细节见：8.10.6节)**
  - 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)

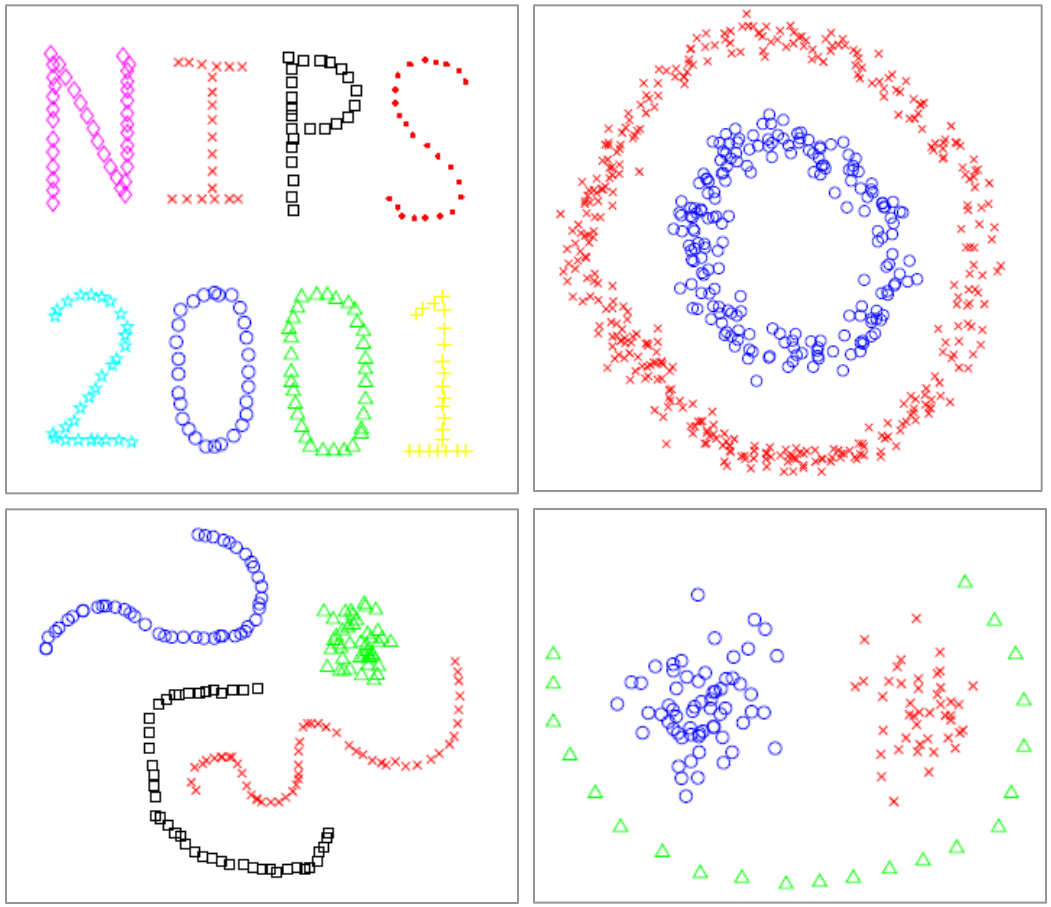
$$\min_{\mathbf{H}\in R^{n\times k}} tr\left(\mathbf{H}^T\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}\mathbf{H}\right), \quad s.t. \ \mathbf{H}^T\mathbf{H}=\mathbf{I}$$

8.10.5 谱聚类

- 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。

8.10.5 谱聚类

- 一些例子



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

8.10.5 谱聚类

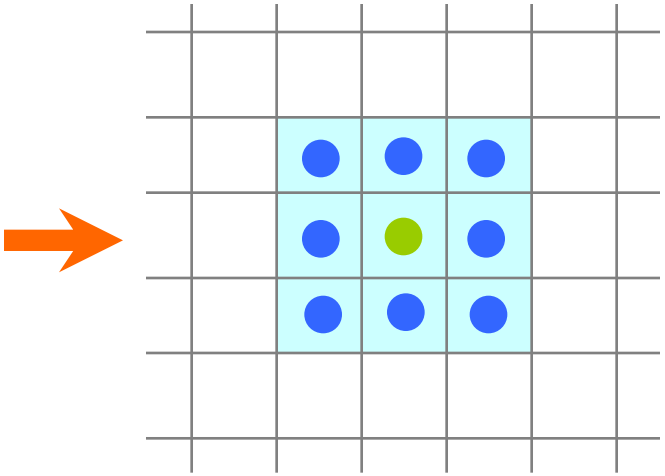
- 采用谱聚类将图像的前景目标分割出来

如何构造近邻图？

8.10.5 谱聚类

- 采用谱聚类将图像的前景目标分割出来
  - 以每个像素为一个顶点，以3X3为一个基本邻域，连接各像素，构建一个图。

图像



格子图

8.10.6 对谱聚类算法的解释

本节不讲，感兴趣的同学自己看

- 进一步解释
  - 可以从如下几个方面进行解释：
    - Graph cut point of view
    - Random walks point of view
    - Perturbation theory point of view
  - 我们主要从前两个角度进行解释。

Ulrike von Luxburg, A Tutorial on Spectral Clustering, Statistics and Computing, 17 (4), pp. 395-416, 2007.

8.10.6 对谱聚类算法的解释

- 图切割—基本定义
  - 子图A的大小：
    - 方法一：采用子图 A 所包含的顶点个数：|A|，**即集合 A 的势。**
    - 方法二：采用子图A所包含的顶点的度数之和，**即体积：**

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

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

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

8.10.6 对谱聚类算法的解释

- 图切割—基本定义
  - 图分割：
    - 设 $A_1, A_2, \ldots, A_k$ 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 $A_1, A_2, \ldots, A_k$ 为图G的一个分割。
  - 聚类指示向量：
    - 设 $A_1, A_2, \ldots, A_k$ 为图 G 的 k 个连图子图，或为其 k 个划分，由每个 A_i 均可以定义一个 n 维指示向量，该向量对于 A_i 各顶点的元素值为 1，其余元素全为0：

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

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

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

piecewise constant

8.10.6 对谱聚类算法的解释

- 从图切割角度分析谱聚类
  - 聚类任务就是根据数据相似性将其分为不同的组。
  - 当数据相似矩阵给定之后，我们希望找到一个图分割，使不同组之间具有较低的相似度，组内部数据点之间具有较高的相似度。
  - 我们将看到谱聚类近似地实现了这一目标。
- 目标函数
  - 考虑将图分成 k 个子图：$A_1, A_2, \ldots, A_k$。一种直观的方法是将图切割问题理解为最小割问题，其目标函数为：

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

8.10.6 对谱聚类算法的解释

- 目标函数
  - 考虑将图分成 k 个子图：$A_1, A_2, \ldots, A_k$。一种直观的方法是将图切割问题理解为经典的最小割问题：

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

  - 在实践中，上述目标函数通常将一个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。
  - 一种消除此缺点的方法是希望分割后的簇有合理的大小，即簇与簇之间的大小不能相差太大。
  - 因此采用集合的势或子图的体积来进行归一化。

8.10.6 对谱聚类算法的解释

- 归一化目标函数
  - 比例切割：

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

  - 归一化切割：

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

  - So what both objective functions try to achieve is that the clusters are “balanced”, as measured by the number of vertices or edge weights, respectively.
  - 但问题为NP难。对Ncut的松驰将导致归一化谱聚类算法

8.10.6 对谱聚类算法的解释

- **Approximating RatioCut for $k = 2$**
  - 此时的目标函数为：

$$\min_{A \subset V} \quad \text{RatioCut}(A, \bar{A})$$

  - 给定一个子集 A⊂V，通过 A 我们来定义一个浮点化的指示向量 $\mathbf{f} = [f_1, f_2, \ldots, f_n]^T \in \mathrm{R}^n$，其分量具有如下形式：

$$f_i = \begin{cases} \sqrt{|\bar{A}|/|A|} & , \text{ if } v_i \in A \\ -\sqrt{|A|/|\bar{A}|} & , \text{ if } v_i \in \bar{A} \end{cases}$$

8.10.6 对谱聚类算法的解释

- **Approximating RatioCut for $k = 2$**
  - 于是，我们有：

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

$$= \frac{1}{2}\sum_{i\in A, j\in\bar{A}} w_{ij}\left(\sqrt{\frac{|\bar{A}|}{|A|}} + \sqrt{\frac{|A|}{|\bar{A}|}}\right)^2 + \frac{1}{2}\sum_{i\in\bar{A}, j\in A} w_{ij}\left(-\sqrt{\frac{|A|}{|\bar{A}|}} - \sqrt{\frac{|\bar{A}|}{|A|}}\right)^2$$

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

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

$$= |V|\,\mathrm{RatioCut}\left(A, \bar{A}\right)$$

8.10.6 对谱聚类算法的解释

- **Approximating RatioCut for $k = 2$**
  - 进一步，我们有：

$$\sum_{i=1}^{n} f_i = \sum_{i\in A}\sqrt{\frac{|\bar{A}|}{|A|}} - \sum_{i\in \bar{A}}\sqrt{\frac{|A|}{|\bar{A}|}} = |A|\sqrt{\frac{|\bar{A}|}{|A|}} - |\bar{A}|\sqrt{\frac{|A|}{|\bar{A}|}} = 0$$

这说明，该解与元素全为1的向量是正交的。

  - 同时有：

$$\sum_{i=1}^{n} f_i^2 = |A|\frac{|\bar{A}|}{|A|} + |\bar{A}|\frac{|A|}{|\bar{A}|} = |\bar{A}| + |A| = n$$

这说明，该解是具有有限元素的。

8.10.6 对谱聚类算法的解释

- **Approximating RatioCut for $k = 2$**
  - 因此有如下松驰问题：

$$\min_{A \subset V} \quad \text{RatioCut}(A, \bar{A})$$

离散划分，仍然NP难

$$\min_{A \subset V} \mathbf{f}^T \mathbf{L} \mathbf{f}, \ \ s.t., \ \ \mathbf{f}^T \mathbf{e} = 1, \ \ \| \mathbf{f} \| = \sqrt{n}, \ \mathbf{f} \text{ as defined}$$

$$\min_{\mathbf{f} \in R^n} \mathbf{f}^T \mathbf{L} \mathbf{f}, \quad s.t. \quad \mathbf{f}^T \mathbf{e} = 1, \quad \| \mathbf{f} \| = \sqrt{n}$$

划分：

$$\begin{cases} f_i \geq 0, & v_i \in A \\ f_i < 0, & v_i \in \bar{A} \end{cases}$$

8.10.6 对谱聚类算法的解释

- **Approximating RatioCut for *k* >2**
  - Given a partition V into A_1 , A_2 , …, A_k , construct indicator vector $\mathbf{h}_i = (h_{i,1}, h_{i,2}, \ldots, h_{i,n})$ for A_i as follows:

$$h_{i,j} = \begin{cases} 1/\sqrt{|A_j|}, & \text{if } v_i \in A_j \\ 0, & \text{otherwise} \end{cases} \quad (i=1,2,\ldots n;\ j=1,2,\ldots,k)$$

  - Further let $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_k] \in R^{n\times k}$, then the vectors $\mathbf{h}_1$, $\mathbf{h}_2$, …, $\mathbf{h}_k$ are orthogonal to each other and $\mathbf{H}^T\mathbf{H}=\mathbf{I}$. Here **I** is an $k \times k$ identity matrix.

8.10.6 对谱聚类算法的解释

- **Approximating RatioCut for *k* >2**
  - 类似地，我们有：

$$\mathbf{h}_j^T \mathbf{L} \mathbf{h}_j = \frac{\mathrm{cut}\left(A_j, \bar{A}_j\right)}{\left|A_j\right|}, \quad \text{且} \ \mathbf{h}_j^T \mathbf{L} \mathbf{h}_j = \left(\mathbf{H}^T \mathbf{L} \mathbf{H}\right)_{jj} \quad \text{（主对角元素）}$$

  - 对所有子图进行累加，于是可得

$$\mathrm{Ratiocut}(A_1, A_2, \cdots, A_k)$$

$$= \sum_{j=1}^{k} \frac{\mathrm{cut}(A_j, \bar{A}_j)}{|A_j|} = \sum_{j=1}^{k} \mathbf{h}_j^T \mathbf{L} \mathbf{h}_j = \sum_{j=1}^{k} \left(\mathbf{H}^T \mathbf{L} \mathbf{H}\right)_{jj} = tr\left(\mathbf{H}^T \mathbf{L} \mathbf{H}\right)$$

8.10.6 对谱聚类算法的解释

- **Approximating RatioCut for k >2**
  - 因此有如下松驰问题：

$$\min_{A_1, A_2, \cdots, A_k} tr\left(\mathbf{H}^T \mathbf{L} \mathbf{H}\right), \quad s.t. \quad \mathbf{H}^T \mathbf{H} = \mathbf{I}, \ \mathbf{H} \text{ as defined}$$

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

划分： $m = \arg\max \{h_{i,j}\}_{j=1,2,\cdots,k} \Rightarrow v_i \in A_m$

$(i = 1,2,\ldots n)$

8.10.6 对谱聚类算法的解释

- **Approximating Ncut for $k = 2$**
  - 此时的目标函数为：

$$\min_{A \subset V} \quad \mathrm{Ncut}(A, \bar{A})$$

  - 给定一个子集 A⊂V，通过 A 我们来定义一个浮点化的指示向量 $\mathbf{f} = [f_1, f_2, \ldots, f_n]^T \in \mathrm{R}^n$，其分量具有如下形式：

$$f_i = \begin{cases} \sqrt{\dfrac{\mathrm{vol}(\bar{A})}{\mathrm{vol}(A)}} & , \text{ if } v_i \in A \\ -\sqrt{\dfrac{\mathrm{vol}(A)}{\mathrm{vol}(\bar{A})}} & , \text{ if } v_i \in \bar{A} \end{cases}$$

8.10.6 对谱聚类算法的解释

- Approximating Ncut for $k = 2$
  - 进一步，不难得到以下结论：

$$(\mathbf{Df})^T\mathbf{e} = 0,\ \mathbf{f}^T\mathbf{Df} = \mathrm{vol}(V),\quad \mathbf{f}^T\mathbf{Lf} = \mathrm{vol}(V)\mathrm{Ncut}(A, \bar{A})$$

  - 于是有如下松驰问题：

$$\min_{A} \mathbf{f}^T\mathbf{Lf},\ s.t.,\ (\mathbf{Df})^T\mathbf{e} = 0,\ \mathbf{f}^T\mathbf{Df} = \mathrm{vol}(V),\ \mathbf{f}\ \text{as defined}$$

$$\min_{\mathbf{f}\in R^n} \mathbf{f}^T\mathbf{Lf},\quad s.t.,\ (\mathbf{Df})^T\mathbf{e} = 0,\ \mathbf{f}^T\mathbf{Df} = \mathrm{vol}(V)$$

8.10.6 对谱聚类算法的解释

- Approximating Ncut for $k = 2$
  - 进一步，我们令 $\mathbf{g} = \mathbf{D}^{1/2}\mathbf{f}$, 于是有：

$$\min_{\mathbf{g}\in R^n} \mathbf{g}^T\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}\mathbf{g}, \quad s.t., \ (\mathbf{D}^{1/2}\mathbf{e})^T\mathbf{g} = 0, \ \|\mathbf{g}\|^2 = \mathrm{vol}(V)$$

注意到: $\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{L}_{sym}$， $\mathbf{D}^{1/2}\mathbf{e}$ 为 $\mathbf{L}_{sym}$ 的最小的特征值（0）对应的特征向量，且vol(V)为常数。因此上述问题就是一个标准的Rayleigh-Ritz 理论所讨论的问题，即该问题的最优解为 $\mathbf{L}_{sym}$ 第二小的特征向量。

如果将 $\mathbf{f} = \mathbf{D}^{-1/2}\,\mathbf{g}$ 重新带入上述问题，不难看到 $\mathbf{f}$ 将是 $\mathbf{L}_{rw}$第二小的特征向量，或者说等价于求如下广义特征值问题 $\mathbf{Lu}=\lambda\mathbf{Du}$ 的第二小的特征向量。

8.10.6 对谱聚类算法的解释

- Approximating Ncut for $k > 2$
  - Given a partition V into $A_1, A_2, \ldots, A_k$, comstruct indicator vector $\mathbf{h}_i = (h_{i,1}, h_{i,2}, \ldots, h_{i,n})$ for A_i as follows:

$$h_{i,j} = \begin{cases} 1/\sqrt{\mathrm{vol}(A_j)}, & \text{if } v_i \in A_j \\ 0, & \text{otherwise} \end{cases} \qquad (i=1,2,\ldots n;\ j=1,2,\ldots,k)$$

  - Further let $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_k] \in \mathrm{R}^{n\times k}$, then the vectors $\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_k$ are orthogonal to each other and **$\mathbf{H}^T\mathbf{D}\mathbf{H}=\mathbf{I}$.** Here **I** is an $k \times$ k identity matrix.

8.10.6 对谱聚类算法的解释

- **Approximating Ncut for k >2**
  - 类似地，我们有：

$$\mathbf{h}_j^T\mathbf{D}\mathbf{h}_j = 1,\ \mathbf{h}_j^T\mathbf{L}\mathbf{h}_j = \frac{\mathrm{cut}\left(A_j, \bar{A}_j\right)}{\mathrm{vol}(A_j)},\ \text{且}\ \mathbf{h}_j^T\mathbf{L}\mathbf{h}_j = \left(\mathbf{H}^T\mathbf{L}\mathbf{H}\right)_{jj}$$

  - 对所有子图进行累加，于是可得

$$\mathrm{Ncut}(A_1, A_2, \cdots, A_k)$$

$$= \sum_{j=1}^{k} \frac{\mathrm{cut}(A_j, \bar{A}_j)}{\mathbf{vol}(A_j)} = \sum_{j=1}^{k} \mathbf{h}_j^T\mathbf{L}\mathbf{h}_j = \sum_{i=1}^{k} \left(\mathbf{H}^T\mathbf{L}\mathbf{H}\right)_{jj} = tr\left(\mathbf{H}^T\mathbf{L}\mathbf{H}\right)$$

- Approximating Ncut for $k>2$
  - 因此有如下松驰问题：

$$\min_{A_1,A_2,\cdots,A_k} tr\left(\mathbf{H}^T\mathbf{L}\mathbf{H}\right), \quad s.t. \quad \mathbf{H}^T\mathbf{D}\mathbf{H}=\mathbf{I},\ \mathbf{H}\text{ as defined}$$

$$\Downarrow$$

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

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

上述问题是一个标准的Rayleigh-Ritz 理论所讨论的问题，即该问题的最优解为 $\mathbf{L}_{sym}$前 k 个最小的特征向量。

如果将 $\mathbf{H}=\mathbf{D}^{-1/2}\mathbf{T}$ **重新带入上述问题，不难看到 H 将由** $\mathbf{L}_{rw}$ 前 k 个最小的特征向量所构成，或者说等价于求如下广义特征值问题 $\mathbf{Lu}=\lambda\mathbf{Du}$ 的前 k 个最小的特征向量。

8.10.6 对谱聚类算法的解释

- Random walks point of view
  - 随机游走：其概念接近于布朗运动，是布朗运动的理想数学状态。
  - 想象一个粒子在一个二维格子上游走。每个格子点可以理解为一个状态。粒子从一个格子点移动至相邻格子点（即从一个状态移动到相邻状态）由一个转移概率来控制。
  - 由于粒子可以移动到所有状态，因此需要一个状态转移概率矩阵来描述。一旦定义了转移状态概率矩阵，粒子的游走从统计上就可得到描述。
  - 粒子可以一直游走下去，达到一个稳态。所谓稳态，是指状态的概率分布不再进行变化。

8.10.6 对谱聚类算法的解释

- **Random walks point of view**
  - 图上的随机游走
    - 从顶点 v_i 至 v_j 一步转移概率由边权值来定义：$p_{ij} = w_{ij} / d_i$
    - 转移概率矩阵：$\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$
    - 如果图是连通的非二部图，一定存在一个稳态分布：$\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_n)^T$, 且$\pi_i = d_i / \mathrm{vol}(V)$。
    - 我们可以看到：$\mathbf{L}_{rw} = \mathbf{I} - \mathbf{P}$

二部图，二分图，偶图，是图论中一种特殊模型。指顶点可以分成两个不相交的集使得在同一个集内的顶点不相邻（没有共同边）的图

8.10.6 对谱聚类算法的解释

- Random walks point of view
  - Random walks VS Ncut
  - 设G为一个连通的非二部图，其转移概率矩阵和稳定分布分别由 **P** 和 π 描述。随机游走记为X_0，$X_1, \ldots, X_t$，…。假设A和B是图G的两个相邻子集，定义粒子从A中出发，最初到达B的概率为$P(B|A) = P(X_1 \in B \mid X_0 \in A)$，则有如下结论：

$$\mathrm{Ncut}\left(A, \bar{A}\right) = P(\bar{A} \mid A) + P(A \mid \bar{A})$$

- 证明如下：

$$P\left(X_0 \in A, X_1 \in B\right) = \sum_{i \in A, j \in B} P\left(X_0 = i, X_1 = j\right) = \sum_{i \in A, j \in B} \pi_i p_{ij}$$

$$= \sum_{i \in A, j \in B} \frac{d_i}{\mathrm{vol}(V)} \frac{w_{ij}}{d_i} = \frac{1}{\mathrm{vol}(V)} \sum_{i \in A, j \in B} w_{ij}$$

进一步，有：

$$P\left(X_1 \in B \mid X_0 \in A\right) = \frac{P\left(X_0 \in A, X_1 \in B\right)}{P\left(X_0 \in A\right)}$$

$$= \left(\frac{1}{\mathrm{vol}(V)} \sum_{i \in A, j \in B} w_{ij}\right)\left(\frac{\mathrm{vol}(A)}{\mathrm{vol}(V)}\right)^{-1} = \frac{\sum_{i \in A, j \in B} w_{ij}}{\mathrm{vol}(A)}$$

最后，根据Ncut的相关定义，直接可以得到结论。

8.11 一些挑战性问题

- **有待进一步解决的问题**
  - 聚类仍然是机器学习、数据挖掘、模式识别中基本研究问题，也是一个热点研究问题。
  - **主要的挑战：**
    - 聚类算法应具有可伸缩性
    - 聚类算法应具有处理不同类型属性的能力
    - 聚类算法应具有发现具有任意形状的类的能力
    - 聚类算法应具有处理高维数据的能力

8.11 一些挑战性问题

- 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，都应该是有效的；无论是对于小类别数据还是具有大别类数目的数据，都应该是有效的。
  - 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。

8.11 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。
    - 数据对象在高维空间的聚类更具有挑战性，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。
  - 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。

8.11 一些挑战性问题

- **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（这一点一直没有解决好）。
    - 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）。

Thank All of You!