

第二十讲 聚类分析

2025.5.26

物以类聚，人以群分

聚类分析 Cluster Analysis

在事先不知道类别信息的情况下(无监督学习), 聚类分析将相似度高的或者距离小的个体聚集成一类(cluster), 不相似的个体分属不同的类。



划分 Partition

指标集 $I = \{1, 2, \dots, n\}$ 的子集 C_1, \dots, C_K 称为 I 的一个 K -划分 (partition), 如果它们两两不交, 且 $\bigcup_{i=1}^K C_i = I$.

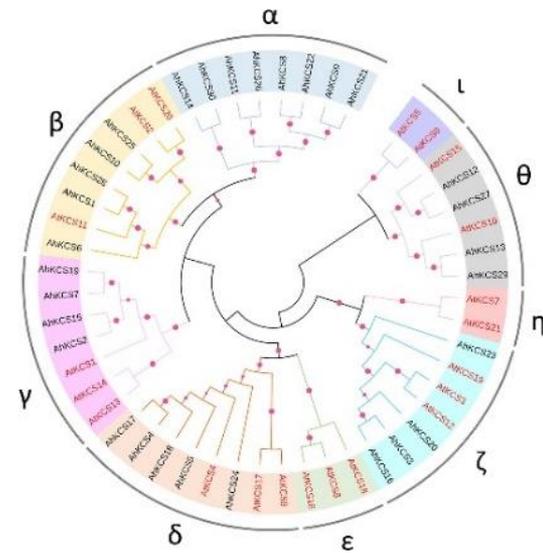
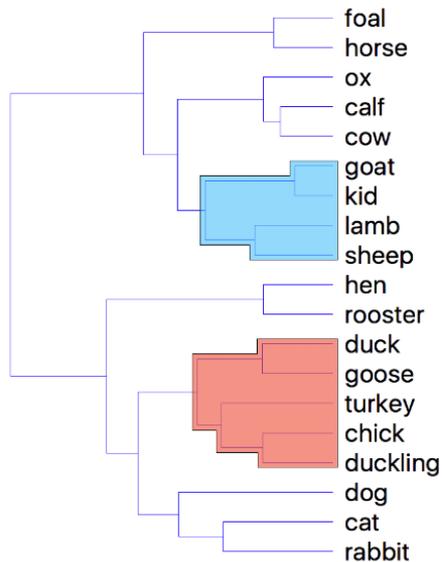
n 个研究对象的 K -划分个数: $\frac{1}{K!} \sum_{j=0}^K (-1)^{K-j} \binom{K}{j} j^n = O(K^n)$

$n = 100, K = 2,$
 $2^{100} = 30$ 位

穷尽所有划分几乎是不可能的, 几乎所有的聚类算法基本都是贪心算法。下面主要介绍几个经典的算法: 层次聚类、 K -medoid 聚类、 K -均值聚类 (混合高斯分布模型)、谱聚类。

层次聚类 Hierarchical Clustering

层次聚类将研究对象逐步合并（或分拆），也称作系统聚类，主要包括**聚合层次聚类法**和**分割层次聚类法**。其中的关键是定义类与类之间的距离。

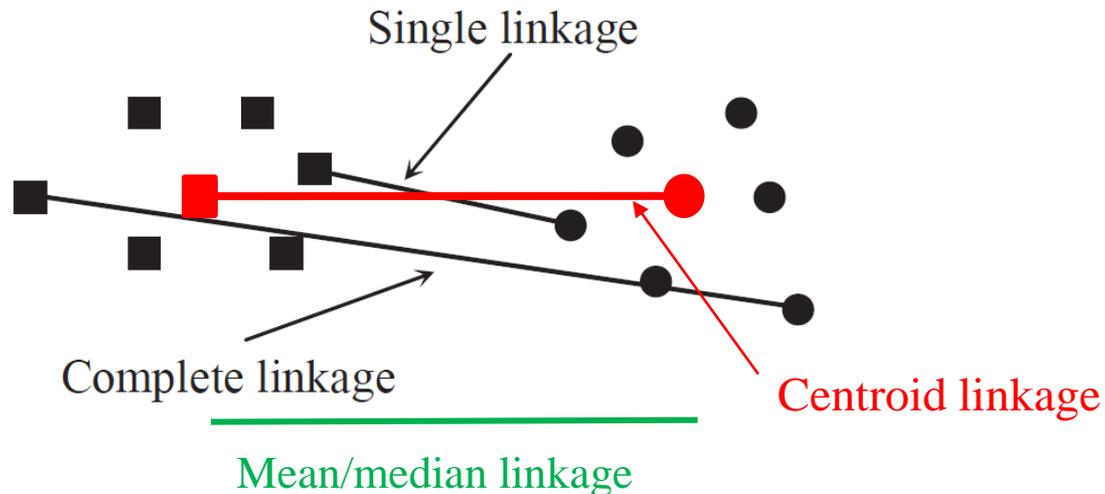


Linkage: 类之间的 距离

层次聚类中需要考虑子集(类)的合并或分拆, 因此需要定义类之间的距离, 我们将距离小的集合进行合并, 称为连结 (linkage: the manner being united)。

假设个体 a, b 之间的距离为 $d(a, b)$, 则集合之间的距离/连结:

- ❑ 单连结(single-linkage): $d_{\min}(A, B) = \min_{a \in A, b \in B} d(a, b)$
- ❑ 完全连结(complete-linkage): $d_{\max}(A, B) = \max_{a \in A, b \in B} d(a, b)$
- ❑ 平均连结(mean-linkage): $d_{\text{mean}}(A, B) = \sum_{a \in A, b \in B} d(a, b) / |A||B|$
- ❑ 其它连结: median, centroid, Ward



Single: 只要两个集合存在一对点距离较小，就认为这两个集合是同一类。



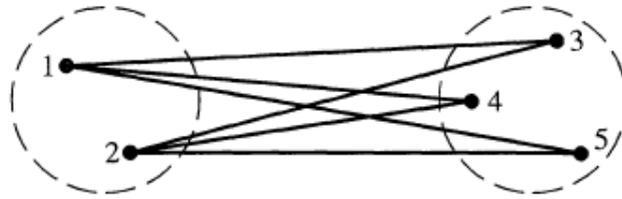
(a)

Complete: 只有当两个集合的所有点都距离较小时才认为两个集合属于同一类



(b)

Average: 当所有点对之间距离的平均/中位数较小时，认为两个集合属于同一类。
median/centroid与此类似。



(c)

Cluster distance

$$d_{24}$$

$$d_{15}$$

$$\frac{d_{13} + d_{14} + d_{15} + d_{23} + d_{24} + d_{25}}{6}$$

聚合层次 聚类算法

聚合（agglomeration）层次聚类法首先把每个研究对象 (object, item, unit) 看作一类，然后逐步合并相异度(距离)最小的或相似度最大的类，相继地逐步聚集

-
1. 初始假定每一个研究对象是一个类(cluster)，计算各个类之间的距离；
 2. 距离最小的两个cluster合并成一个更大的cluster；
 3. 重新计算各个类(cluster)之间的距离；
- 重复2，3直到达达到某个预定标准或者所有对象合并成一类。

```
R: hclust(d) # d: 距离矩阵
```

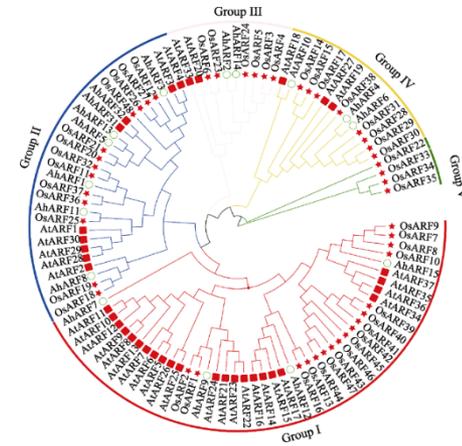
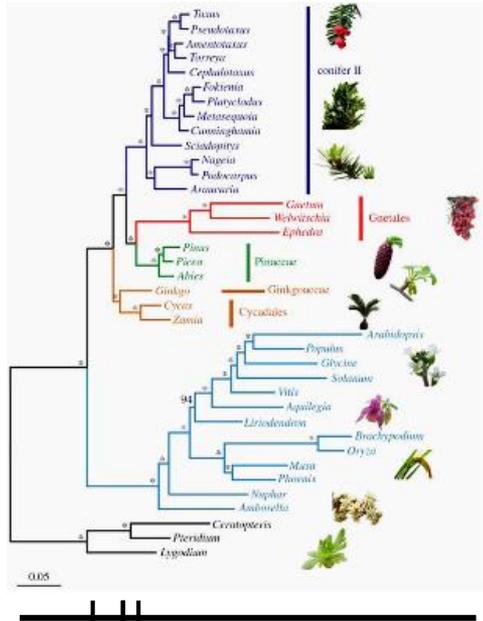
分割层次 聚类算法

分割层次聚类法

与聚合聚类相反，首先将所有个体当作一类，逐步分割成距离最大或相似度最小的两类。效果与聚合方法类似，不常用。

树图

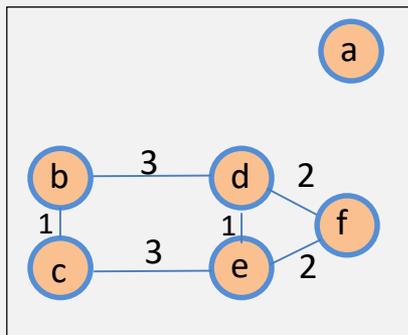
树状图(Dendrogram) 表示层次聚类关系，在根部所有物体为一类，朝枝叶方向依次细分（左图）树旁的刻度尺表示聚集或分拆时的距离。更紧凑的树图如右图（不再是树状！）



```
myhc= hclust(d,method=) #层次聚类  
plot(myhc) #画树图
```

```
library(ape)  
plot(as.phylo(myhc), type = "fan")
```

例1. 平面上的点a-f, 单连结聚合层次聚类



	a	b	c	d	e
b	9				
c	9	1			
d	7	3	4		
e	8	4	3	1	
f	7	5	5	2	2

步骤细节:

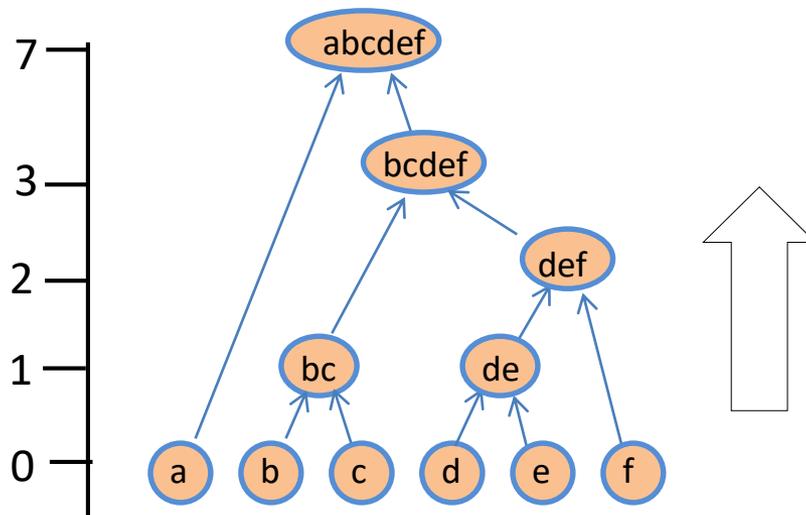
b,c之间、d,e 之间的距离都是1, 最小, 首先将它们合并集为(bc), (de)两个类。至此有4个类:

(bc), (de), a,f

计算这它们的两两距离, f与(de)距离2最小, 合并为(def), 至此有3个类

(bc), (def), (a)

3类间(bc), (def)距离3最小, 合并(bcdef), (a)

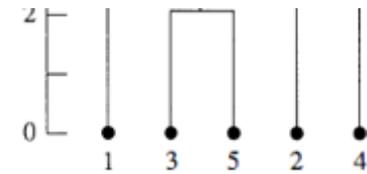
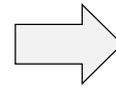


交汇/合并处的数字(Height)表示聚合时的距离, 比如f与(d,e)聚合时的距离为2。

例2 (课本例12.3), 5个物体的距离矩阵如右, 考虑Single-linkage 聚集层次聚类分析。

$$D = \{d_{ik}\} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0 & & & & \\ 9 & 0 & & & \\ 3 & 7 & 0 & & \\ 6 & 5 & 9 & 0 & \\ 11 & 10 & \textcircled{2} & 8 & 0 \end{bmatrix} \end{matrix}$$

1. 把每个对象当作一个类, $d_{35} = \textcircled{2} = \min(d_{ij})$, 将3,5合并为一类(35), 得到4个类: (35), 1, 2, 4



计算4个类:(35),1,2,4的两两距离:

$$d_{(35)1} = \min(d_{31}, d_{51}) = \min(3, 11) = 3,$$

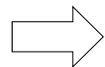
$$d_{(35)2} = \min(d_{32}, d_{52}) = 7,$$

$$d_{(35)4} = \min(d_{34}, d_{54}) = 8$$

1,2,4之间的距离9,6,5

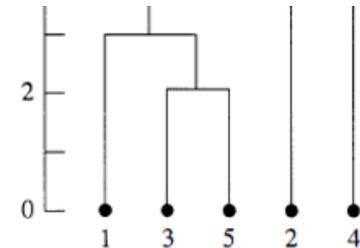
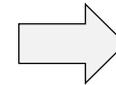


$$\begin{matrix} & \begin{matrix} (35) & 1 & 2 & 4 \end{matrix} \\ \begin{matrix} (35) \\ 1 \\ 2 \\ 4 \end{matrix} & \begin{bmatrix} 0 & & & \\ \textcircled{3} & 0 & & \\ 7 & 9 & 0 & \\ 8 & 6 & 5 & 0 \end{bmatrix} \end{matrix}$$



最小距离 $\textcircled{3} = d_{(35)1}$

合并(35), (1)

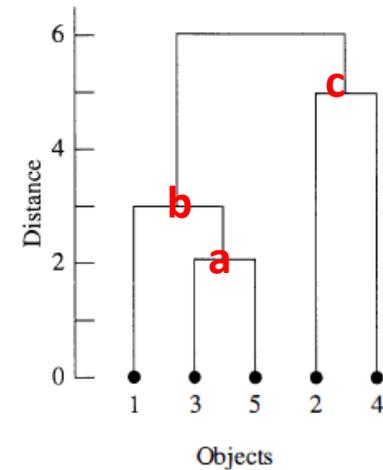
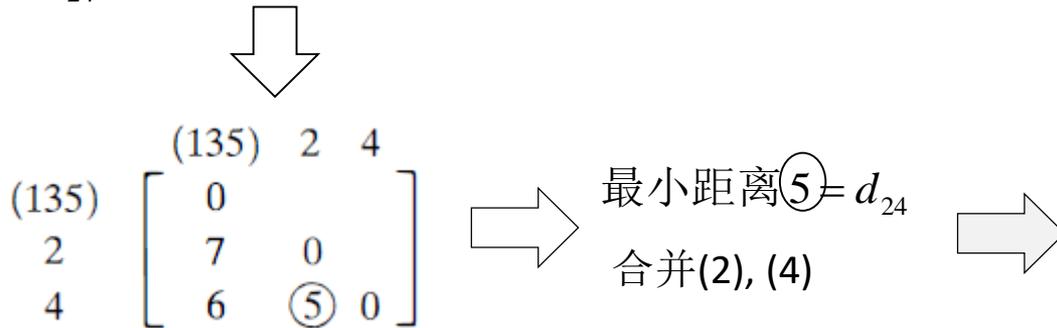


2. 合并(35)和1之后，计算(351),2,4三个类的两两距离：

$$d_{(351)2} = \min(d_{(35)2}, d_{12}) = \min(7, 9) = 7$$

$$d_{(351)4} = \min(d_{(35)4}, d_{14}) = \min(8, 6) = 6$$

$$d_{24} = 5$$



3. 合并2,4为之后，共有两个类(135),(24)，其距离 = 6，合并，完成。

读树图：

- a:** 3, 5首先合并，距离=2，
- b:** (35)与1合并，距离=3，
- c:** 2,4合并，距离=5。

例3. 欧洲语言例子

```
similarity=read.table("http://staff.ustc.edu.cn/~ynyang/vector/data/Euro-language1.txt")
```

```
similarity=as.matrix(similarity)
```

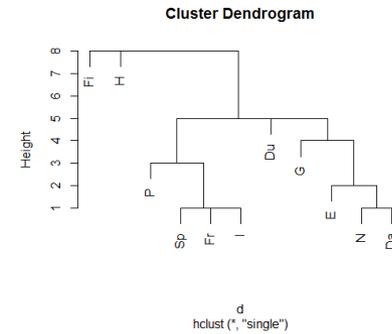
```
d=10-similarity # 相似性转为距离
```

```
#或  $d_{ij} = \sqrt{s_{ii} + s_{jj} - 2s_{ij}}$ 
```

```
d=as.dist(d) #
```

```
myclust= hclust(d, method="complete")
```

```
plot(myclust) #画树图 (右图)
```



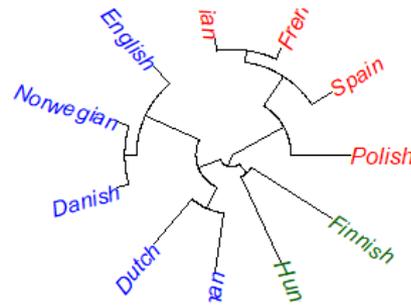
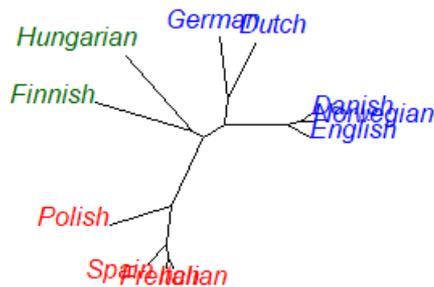
#树图的其它画法:

```
library(ape)
```

```
co=c(rep("blue",5), ep("red",4),rep("darkgreen",2))
```

```
plot(as.phylo(myclust), type = "unrooted")
```

```
plot(as.phylo(myclust), type = "fan",tip.color=co)
```



```

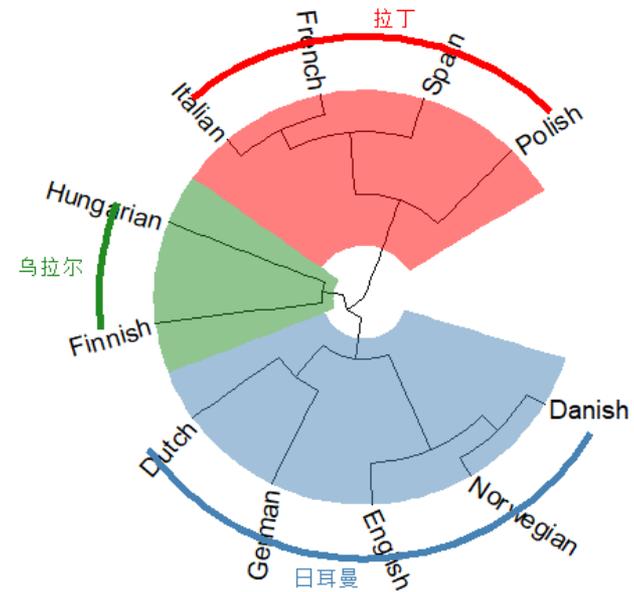
install.packages("BiocManager")
BiocManager::install("ggtree")
# Update all/some/none? [a/s/n]: n
library(ggtree)
hc=hclust(d)

ggtree(hc,layout="circular")+
  xlim(0,7)+
  geom_tiplab2(offset=0.1,
              size=5)+
  # geom_text(aes(label=node))+

  geom_highlight(node = 13,fill="red")+
  geom_highlight(node=15,fill="steelblue")+
  geom_highlight(node=16,fill="forestgreen")+

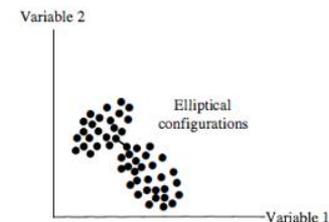
  geom_cladelabel(node=13,label="拉丁",
                 offset=1.2,barwidth = 2,
                 vjust=-0.5,color="red")+
  geom_cladelabel(node=15,label="日耳曼",
                 offset=1.2,barwidth = 2,
                 hjust=1.2,vjust=1.5,color="steelblue")+
  geom_cladelabel(node=16,label="乌拉尔",
                 offset=1.2,barwidth = 2,
                 hjust=1.2,color="forestgreen")

```

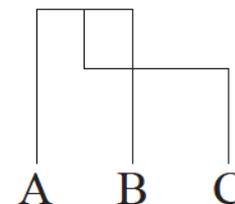


各种连结的比较

- **Complete**方法以两个类的最远的点之间的距离来决定两个类是否应该聚合或拆分，因而适合类内较为**紧密**，类之间分离的较好的情形。
- **Single**方法容易形成**长链**式的聚集效果，因而对非线性的聚簇数据聚类效果可能较好（右图）。
- **Average**方法介于single、complete之间。
- 层次聚类计算复杂度 $O(n\log(n))$
- **Centroid, median** 较为稳健，但可能会产生逆序(*inversion*)现象（右图）。
- **Ward**合并使得组间平方和变化最小的两类。



(a) Single linkage confused by near overlap



K-medoid 聚类

K-medoid 聚类算法使用距离矩阵进行聚类，各类的中心限定为某个个体，称为 medoid。尤其适用于只有主观距离而没有原始向量数据的情形。

中心点 Medoid

定义(medoid): 假设 n 个个体的距离矩阵为 $D = (d_{ij})_{n \times n}$, 与其它点的距离之和最小的点定义为 n 个个体的中心(medoid):

$$\text{medoid} = \arg \min_{j \in \{1, \dots, n\}} \sum_{i=1}^n d_{ij}.$$

K-medoid 聚类问题

假设 n 个个体的距离为 $D = (d_{ij})_{n \times n}$, $d_{ij} = \text{dist}(i, j)$, K -medoid 方法求解 $\{1, 2, \dots, n\}$ 的 K -划分 C_1, \dots, C_K 及其中心 medoids $\{m_1, \dots, m_K\} \subset \{1, 2, \dots, n\}$, 使得组内距离之和极小:

$$\sum_{k=1}^K \sum_{i \in C_k} \text{dist}(i, m_k) = \min!$$

- 当划分已知时，根据定义，容易确定各类的中心；
- 当各类中心已知时，我们只需把每个个体划分到离其最近的中心所在的类；给定划分或中心的初始值，

上述两步递归迭代。

K - medoid clustering算法：

1. 给定 $\{1,2,\dots,n\}$ 的一个划分 C_1,\dots,C_K ,对每一个类 $1\leq k\leq K$,类内与其它个体总距离最小的点 $m_k \in C_k$ 作为中心：

$$m_k = \arg \min_{i \in C_k} \sum_{j \in C_k} d_{ij}$$

2. 给定当前的中心 $\{m_k, k = 1, \dots, K\}$,将每个个体划分到离它最近的中心所属的类中，即对所有 $1 \leq i \leq n$,求

$$k_i = \arg \min_{1 \leq k \leq K} d_{im_k}, \quad 1 \leq k_i \leq K,$$

并将个体 i 划归到第 k_i 类。由此得到更新的划分 C_1,\dots,C_K 。

3. 迭代1-2
-

```
R: pam 函数 (package: cluster)
> Library(cluster)
> pam(d, k=2)
```

例5 (欧洲语言) 欧洲11种语言k-medoid聚类, 应用k-medoids

```
#距离矩阵 d=10-s, s: similarity matrix
> pam(d,4) # k=4 (4类)
```

Medoids:

ID

```
[1,] "2" "Norwegian" 日耳曼语中心
[2,] "8" "Italian" 拉丁语中心
[3,] "10" "Hungarian"
[4,] "11" "Finnish"
```

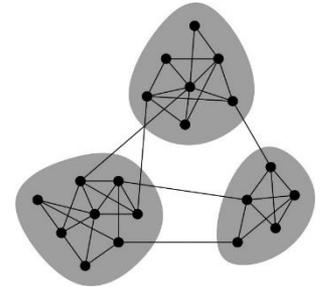
Norwegian和Italian分别是日耳曼和拉丁的中心。拉丁源于罗马，日耳曼源于挪威？

Clustering vector:

English	Norwegian	Danish	Dutch	German	
1	1	1	1	1	
French	Spain	Italian	Polish	Hungarian	Finnish
2	2	2	2	3	4

K-均值聚类 (K-means clustering)

K-means 聚类方法将对象划分为K类，使得类内距离极小(\Leftrightarrow 类间距离最大)，各类的中心定义为类内样本平均值 (Lloyd,1957; MacQueen, 1967)。
K-means 方法应用非常广泛。



K-means方法与K-medoid算法类似，前者主要应用于有原始数据的情形，而后者方法适用于没有原始数据而只有距离矩阵的情形。K-means方法中各类中心为类内样本平均，而K-medoid的中心是一个研究对象。

平方和分解

引理：假设 $\mathbf{x}_1, \dots, \mathbf{x}_n \in R^p, K \geq 2$, 假设 C_1, \dots, C_K 是指标集 $I = \{1, 2, \dots, n\}$ 的一个划分(即 C_1, \dots, C_K 互斥, 且 $\bigcup C_i = I$). 则总平方和可分解为(向量模平方式) 组内平方和 W 和组间平方和 B :

$$SS_T = \sum_{i=1}^n \|\mathbf{x}_i - \bar{\mathbf{x}}\|^2 = \sum_{k=1}^K \sum_{i \in C_k} \|\mathbf{x}_i - \bar{\mathbf{x}}_k\|^2 + \sum_{k=1}^K \sum_{i \in C_k} \|\bar{\mathbf{x}}_k - \bar{\mathbf{x}}\|^2 \stackrel{\Delta}{=} W + B$$

K均值聚类问题

给定 n 个点 $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in R^p$, 以及类的个数 K .

目标: 寻找最优的 $I = \{1, 2, \dots, n\}$ 的划分 $C = \{C_1, \dots, C_K\}$, 以及各组的中心 $\{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K\}$ 使得组内平方和最小

$$\min_{C_k, \boldsymbol{\mu}_k, k=1, \dots, K} \sum_{k=1}^K \sum_{i \in C_k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2$$

划分: C_k 's 互斥且 $\bigcup_{k=1}^K C_k = I$

K-means 算法

K-means 算法 (Lloyd, 贪心算法)

- 当划分给定时, 第 k 个类的中心 $\boldsymbol{\mu}_k$ 的最优估计为平均值

$$\hat{\boldsymbol{\mu}}_k = \bar{\mathbf{x}}_k = \frac{1}{|C_k|} \sum_{i \in C_k} \mathbf{x}_i, \quad |C_k| \text{ 为 } C_k \text{ 中点的个数}$$

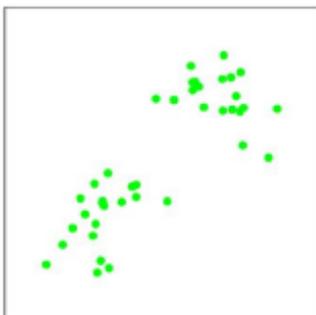
- 而当中心 $\boldsymbol{\mu}_k, k = 1, \dots, K$ 给定时, C_k 容易决定:

$$C_k = \{1 \leq i \leq n: \|\mathbf{x}_i - \boldsymbol{\mu}_k\| \leq \|\mathbf{x}_i - \boldsymbol{\mu}_s\|, \text{ 对所有 } s \neq k\}$$

反复迭代上述两步, 直至收敛。

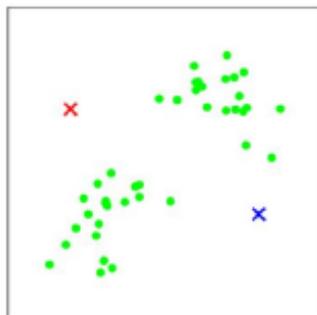
K-means算法示意图

(a)数据



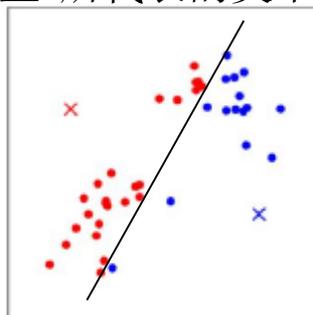
(a)

(b)任取两个初始中心
(以红x、蓝色x表示)

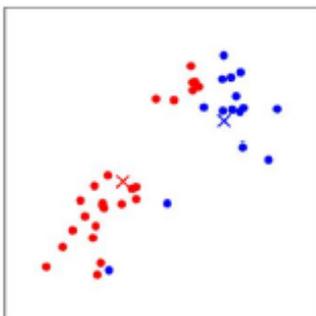


(b)

(c)红x、蓝色x中垂线
将数据分别划分到红、
蓝x所代表的类中

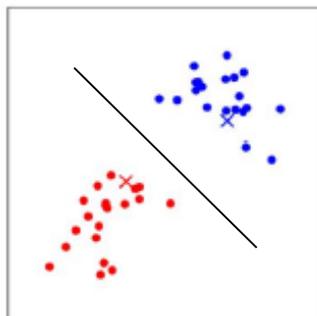


(c)



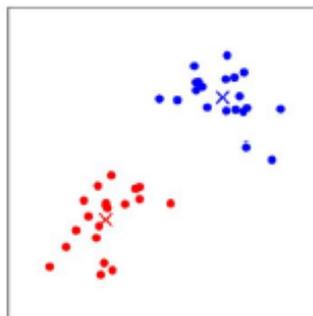
(d)

(d) 计算红点和蓝点的均值，作为两类新的中心(红、蓝x)。



(e)

(e) 红、蓝x)的中垂线将数据分别重新划分到新的红、蓝x所代表的类中



(f)

(f) 重新计算中心，再次分类，与上次相同。停止迭代。

注:

- 初始步可以指定初始中心，也可以指定初始的划分。
- K - means, K - medoid都是原型 $prototype$ 方法，中心代表类，称为原型（不同的是 K - medoid的各类中心必须是某些研究对象）。
两者的算法思想几乎一样，都是中心、划分两个步骤的反复迭代。
- K - means得到的解未必一定是全局最优解，但非常高效，其计算复杂度为 $O(npKi)$,其中 i = 迭代次数， i 通常较小($i < 10$).
- 为了避免得到局部最优解，通常取多组不同起始点，多次进行聚类，从中选取目标函数最小者。 R 函数 $kmeans$ 中指定重复次数 $nstart$:
> $kmeans(x, centers = k, nstart = 25)$
#25次,每次随机选取 x 的 k 行作为 k 类的中心初值。

高斯混合分布模型

高斯混合分布假设数据点来自于若干个正态分布（混合），允许不同的类有不同的方差。K-means聚类方法可看作是高斯混合分布模型的一种特殊情况（方差都是球对称情况）。

Gaussian
mixture

记 $f_k(\mathbf{x})$ 为正态分布 $N_p(\boldsymbol{\mu}_k, \Sigma_k)$ 的密度函数，若随机向量 \mathbf{x} 的概率密度

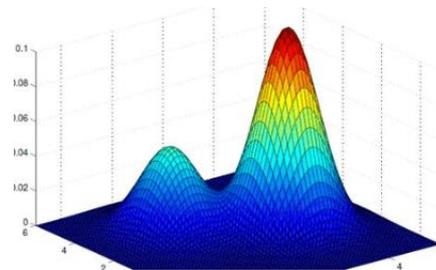
$$f(\mathbf{x}) = \sum_{k=1}^K p_k f_k(\mathbf{x}), p_k \geq 0, p_1 + \dots + p_K = 1,$$

称为是一个高斯混合分布（Gaussian Mixture Model）。

记 $\Theta_k = \{\boldsymbol{\mu}_k, \Sigma_k, p_k\}$ 为第 k 个类的所有参数。

为什么假设多个正态混合？

总体不一致，不是单个正态，而是具有多个峰。
具体到每个样本，我们并不知道它来自于哪个总体。



假设 \mathbf{x} 有一个对应的类别标号 G (潜变量, 不可观测), 其分布为

$$P(G = k) = p_k, \quad p_k \geq 0, \quad 1 \leq k \leq K, \quad \sum_{k=1}^K p_k = 1.$$

假设给定 $G = k$ 时, $\mathbf{x} \sim N(\boldsymbol{\mu}_k, \Sigma_k)$, 即 $\mathbf{x} | G = k \sim f_k(\mathbf{x})$, 则 \mathbf{x} 的边际分布:

$$f(\mathbf{x}) = \sum_{k=1}^K P(G = k)P(\mathbf{x} | G = k) = \sum_{k=1}^K p_k f_k(\mathbf{x}).$$

这是一个潜变量生成模型。

EM算法试图预测每个样本所属的类别 (潜变量) $G_i, i = 1, \dots, n$, 求解步骤为E - step和M - step反复迭代, 本质上与 k - means算法相同:

E - step: 在各类中心已知条件下, E - step将每个样本划分到其所属的类, 这需要计算每个样本点属于每个类的条件概率 / 条件期望, 然后将每个样本点划分到概率最大的一类中。

M - step: 当所有样本所属类别已知的条件下, 用极大似然方法估计每个类的中心 (以及其它参数)。

EM算法细节:

条件期望E用于预测潜变量，极大似然M用于估计参数

- M-step (Maximization - 极大似然) : 估计类的中心

当知道每个样本所属类别时(即已知 $G_i, i = 1, \dots, n$), 求各类中心:

$$\hat{\boldsymbol{\mu}}_k = \sum_{G_i=k} \mathbf{x}_i / n_k = \sum_{i \in \text{class } k} \mathbf{x}_i / n_k \quad (\text{第}k\text{类内样本的平均值})$$

$$\text{以及 } \hat{\boldsymbol{\Sigma}}_k = \sum_{G_i=k} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^T / n_k, \hat{p}_k = \sum_{i=1}^n 1_{(G_i=k)} / n, k = 1, \dots, K.$$

- E-step (Expectation - 条件期望) : 预测 G_i

给定 $\Theta_k, k = 1, \dots, K$, 将每个数据点划分到概率最大的类中:

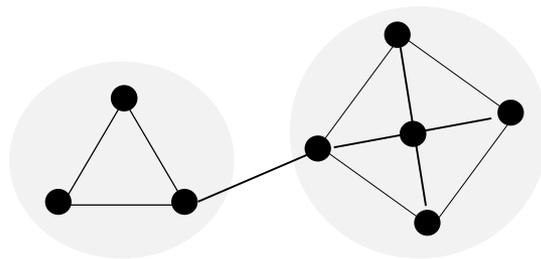
若 $f_k(\mathbf{x}_i) > f_l(\mathbf{x}_i), \forall l \neq k$, 则 G_i 预测为 $\hat{G}_i = k$,

即 $\hat{G}_i = \arg \max_k P(G_i = k | \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k)$.

当各类都是球对称正态时, G_i 预测这一步等价于将样本划分到欧氏距离最近的类中。当不是球形分布时, 这一步计算马氏距离。

谱聚类

基于相似度矩阵（邻接矩阵或权重矩阵）的特征向量进行聚类。



图表示

聚类问题在图(graph)或网络的框架中讨论或演示比较方便。给定 n 个对象的 $n \times n$ 相似度矩阵 $A = (a_{ij})$, $a_{ij} \geq 0$, 我们将对象当作图的节点, 若 $a_{ij} > 0$, 则称节点 i, j 是相邻的, 记作 $i \sim j$, 即它们之间有一条边, 而相似度 a_{ij} 称为该条边的权重。

$A = (a_{ij})$ 通常称为权重矩阵(视作相似度矩阵)。

邻接矩阵

最重要的是相似度/权重为0-1的情形,

$$a_{ij} = \begin{cases} 1, & i \sim j \\ 0, & \text{无边} \end{cases}$$

此时 $A = (a_{ij})$ 称为邻接矩阵(adjacency matrix)。

度数

第 i 个节点的度数: $d_i = \sum_{j=1}^n a_{ij}$

度数矩阵: $D = \text{diag}(d_1, \dots, d_n)$

拉普拉斯矩阵(Laplacian):

$$L = D - A$$

作为线性变换, L 与拉普拉斯算子有类似的作用, 即拉普拉斯矩阵比较每个节点 i 与邻点的平均。说明如下:

仅考虑0-1邻接情形(其它类似)。假设节点 i 的属性/特征标签 x_i , $\mathbf{x} = (x_1, \dots, x_n)^\top$, 令 $\mathbf{y} = L\mathbf{x}$, 令 $\bar{x}_i = \sum_{j:j \sim i} x_j / d_i$ 为与节点 i 邻接/相似的节点的平均, 则

$$y_i = d_i x_i - \sum_{j:j \sim i} x_j = d_i(x_i - \bar{x}_i),$$

表示节点 i 与其相邻点的平均值的差异。 $L\mathbf{x} = 0$ 意味着每个节点与其相邻的点无差异。

拉普拉斯算子: $\Delta f = \nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}$,

$$\begin{aligned} \frac{\partial^2 f}{\partial x^2} &\approx \frac{f'(x+h) - f'(x)}{h} \approx \frac{[f(x+h) - f(x)]/h - [f(x) - f(x-h)]/h}{h} \\ &= \frac{f(x+h) + f(x-h) - 2f(x)}{h^2} \end{aligned}$$

热传导方程: $\frac{\partial T}{\partial t} = \kappa \Delta T$, 某点的温度 $T(x, y, z; t)$ 的时间变化率与该点空间位置附近的温差成正比。

如果加权图的节点由 K 个互不连通的子集构成，则我们认为每个子集构成一个类(*cluster*)。如果图不是全连通的，我们希望划分节点为 K 个*cluster*，不同*cluster*之间的相似性/连通性最小。

mincut

假设节点分属若干不同的类，*cut*定义为相邻但不同类的边的相似系数/权重之和

$$\text{cut} = \sum_{i, j} a_{ij} 1(i, j \text{ 不同类})$$

最优划分使得*cut*数最小。

为什么称为*cut*? 对于0-1图，即 $a_{ij} = 0/1$ 的情形：

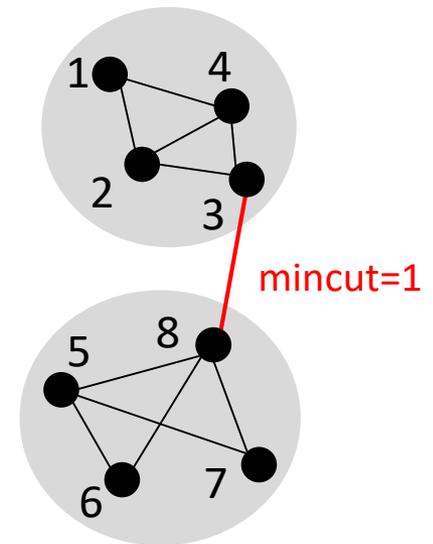
cut = 相邻但不同类的边的个数，

如果移除这些边，则图可以划分为若干互不连通的子集。

右图：

cut = 1: 移除边3-8后可到不连通的两个子集

cut = 2: 移除边2-3和3-4两条边也可得到不连通的两个子集；



2-cluster 组合优化

下面针对两类问题给出mincut问题的数学表示，为此引入每个节点 i 的类别标签 $x_i = 1$ 或 0 ，则

$$\text{mincut} = \min \sum a_{ij} 1_{(x_i \neq x_j)} = \min \sum a_{ij} (x_i - x_j)^2$$

这是组合优化问题。与谱配列中一样，我们有

若 a_{ij} 较大，则
应该有 $x_i = x_j$

$$\text{命题1. } \sum a_{ij} (x_i - x_j)^2 = 2\mathbf{x}^\top L\mathbf{x}, \quad L = D - A$$

$$\begin{aligned} \text{验证: } \sum_{i,j} a_{ij} (x_i - x_j)^2 &= 2 \sum_j d_j x_j^2 - 2 \sum_{i,j} a_{ij} x_i x_j \\ &= 2\mathbf{x}^\top (D - A) \mathbf{x} = 2\mathbf{x}^\top L\mathbf{x} \end{aligned}$$

放松 $x_i = 1$ 或 0 的限制，假设 $\mathbf{x} = (x_1, \dots, x_n)^\top \in S^{n-1}$

2-cluster 二次优化

定义 $L = D - A$ 为拉普拉斯矩阵，mincut问题转化为

$$\min_{\mathbf{x}} \mathbf{x}^\top L\mathbf{x} \quad \text{s.t. } \mathbf{x} \in S^{n-1}, \quad \mathbf{x} \perp \mathbf{1}$$

约束：为了解唯一
一及避免平凡解

得到的解 \mathbf{x} 是 L 的特征向量，含有cluster信息（参见命题2，3），将它们划分为两类即可。

K类 mincut

考虑将 n 个节点划分为 K 个类: C_1, \dots, C_K , 以 $V_{n \times K} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^\top$ 代表节点的类别属性, 其中 $\mathbf{x}_i^\top = (1_{(i \in C_1)}, \dots, 1_{(i \in C_K)})$ 为 C_1, \dots, C_K 的示性变量

$$\text{cut} = \sum a_{ij} 1_{(\mathbf{x}_i \neq \mathbf{x}_j)} = \frac{1}{2} \sum_{i,j} a_{ij} \|\mathbf{x}_i - \mathbf{x}_j\|^2 = \text{tr}(V^\top L V)$$

其中 V 是one-hot矩阵, 每一行仅有一个1, 其余全为0.

现在放松 V 分量为0-1的要求, 但为了唯一性, 增加 V 各列模长为1, 相互正交、且与 $\mathbf{1}$ 正交的限制, 即 $V^\top V = I_K$

二次优化 mincut

定义 $L = D - A$ 为拉普拉斯矩阵, mincut问题转化为:

$$\text{mincut} = \min \sum a_{ij} \|\mathbf{x}_i - \mathbf{x}_j\|^2 \Leftrightarrow \min_{V \in \mathbb{R}^{n \times K}} \text{tr} V^\top L V$$

$$\text{s. t. } V^\top V = I_K.$$

谱聚类对 L 的前 K 个最-小特征根(包括0)的特征向量进行K-means聚类, 原因参见下页命题2-3。

注: 组合优化问题: $\min_{\mathbf{x}: \text{one-hot}} \sum a_{ij} \|\mathbf{x}_i - \mathbf{x}_j\|^2$ 是一个模板, 我们已经多次利用它, 通过放松 \mathbf{x} 的限制诱导出二次优化问题。

全连通加权图：对任何两个节点 i, j , 存在节点 x, y, \dots, z , 使得 $i \sim x \sim y \sim \dots \sim z \sim j$, 其中 $a_{ix} > 0, a_{xy} > 0, \dots, a_{zj} > 0$.

L 的特征向量与连通性

命题2. (1) $L \geq 0$, 最小特征根为0, $\mathbf{1}$ 是特征根0对应的特征向量;
(2) 图是全连通的 (即对任何两个节点, 都存在一个若干边组成的路径连结它们) 当且仅当 L 的0特征根重数为1 (即除了常数因子之外, $\mathbf{1}$ 是唯一特征向量)。

拉普拉斯矩阵 L 的最小非0特征根称为Fiedler数

证明: (1) 由命题1, 对任何 \mathbf{x} , $\mathbf{x}^\top L \mathbf{x} = \frac{1}{2} \sum a_{ij} (x_i - x_j)^2 \geq 0$, 所以 $L \geq 0$.

因为 $L\mathbf{1} = (D - A)\mathbf{1} = D\mathbf{1} - A\mathbf{1} = \mathbf{d} - \mathbf{d} = \mathbf{0}$, 所以0是特征根, $\mathbf{1}$ 是特征向量。

(2) 若 \mathbf{x} 是0特征根的特征向量, 则 $L\mathbf{x} = \mathbf{0x} = \mathbf{0} \Rightarrow$

$$\mathbf{x}^\top L \mathbf{x} = \frac{1}{2} \sum_{i \sim j} a_{ij} (x_i - x_j)^2 = 0 \Rightarrow \text{当 } a_{ij} > 0 \text{ 时, } x_i = x_j,$$

若图是全连通的, 则 $x_1 = \dots = x_n$, 即 $\mathbf{x} = \mathbf{1}x_1$, 唯一, 所以0重数为1.

命题3. 拉普拉斯矩阵 L 的0特征根的重数 K (L 的零空间维数)等于图(包括加权图)的互不连通的全连通子图的个数, 且零空间由 $\mathbf{1}_{G_1}, \dots, \mathbf{1}_{G_K}$ 张成, 其中 G_1, \dots, G_K 为连通子图的节点集合。

证明: 不同的子图之间没有连结, 所以可以重排 L 成为分块对角阵:

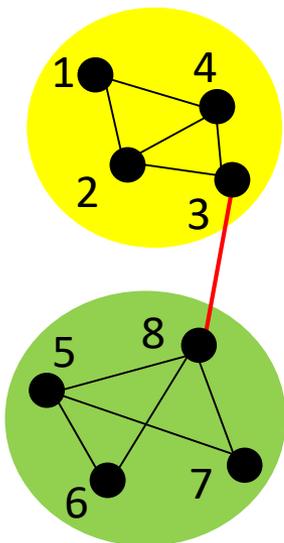
$$L = \begin{pmatrix} L_{G_1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & L_{G_K} \end{pmatrix},$$

由命题2, 在连通子图上特征向量 \mathbf{x} 的分量取值全部相同,

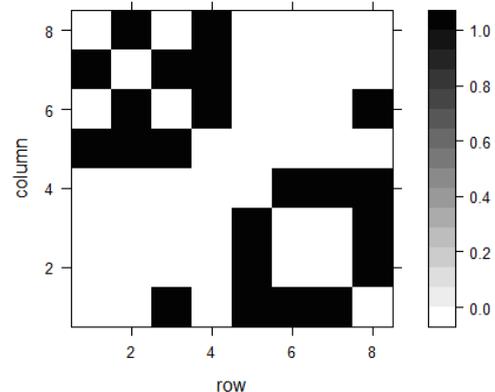
不同的全连通子集上取值可能不同, 即 $\mathbf{x} = c_1 \mathbf{1}_{G_1} + \dots + c_K \mathbf{1}_{G_K}$. 证毕。

如果一个图由不连通的 K 个子图构成, 那么0特征根的所有 K 个特征向量包含所有子图信息。实际问题中的图一般是全连通的(特征根0的重数为1), 但我们还是以 L 的 $K - 1$ 个最小非0特征根对应的特征向量进行聚类。

例1. 下图的邻接矩阵如下



$$A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{matrix}$$



拉普拉斯矩阵L的最小2个特征根为: 0, 0.3, 对应的特征向量为

特征根	0	0.3
1	-0.35	-0.45
2	-0.35	-0.37
3	-0.35	-0.18
4	-0.35	-0.37
5	-0.35	0.37
6	-0.35	0.37
7	-0.35	0.37
8	-0.35	0.26

第二列特征向量说明

节点1-4为一类,

节点5-8为一类。

目标：给定 n 个物体的相似度矩阵 A ，将 n 个物体聚为 K 类， K 已知。

- 假设 $A_{n \times n}$ 是 n 个个体/物体的相似度矩阵（如果是0-1矩阵，可认为是普通图的邻接矩阵，否则认为是加权图的权重矩阵）。
- 计算度数矩阵 $D = \text{diag}(\mathbf{d})$ ，其中 $\mathbf{d} = (d_1, \dots, d_n)^\top = A\mathbf{1}$ ，拉普拉斯 $L = D - A$
- 计算 L 的谱分解，取其前 K 个最小特征根对应的特征向量

$$V = (\mathbf{v}_1, \dots, \mathbf{v}_K)$$

包括0特征根(可能重复)及其它最小非0特征根的特征向量。

- 记 V 各行为 $\mathbf{y}_1, \dots, \mathbf{y}_n \in R^K$ ，即 $V = (\mathbf{y}_1, \dots, \mathbf{y}_n)^\top$
对 $\mathbf{y}_1, \dots, \mathbf{y}_n \in R^K$ 进行 K -means聚类，将 n 个节点聚成 K 类。
-

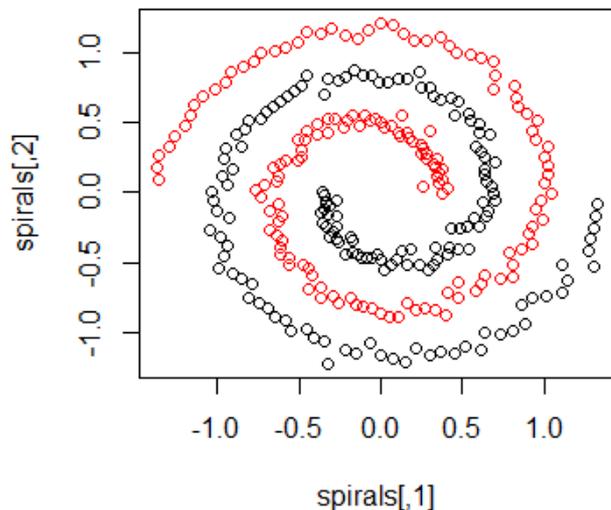
如果问题给定的是距离矩阵 (d_{ij}) ，则可以通过某种变换转化为相似度矩阵，通常用Gaussian kernel: $a_{ij} = \exp(-d_{ij}^2 / 2\sigma^2), i \neq j$.

谱聚类R函数：

```
specc (package:kernlab); cluster_fast_greedy  
(package:igraph) ; fastClustering (package:sClust) ...
```

例1. kernlab中的数据集spirals是二维平面上的两类螺旋状数据（ 300×2 ），我们使用谱聚类方法进行聚类，点之间的距离用欧氏距离，点之间的相似度（权重）用高斯核计算。基于拉普拉斯矩阵的两个最小特征根对应的特征向量进行k-means聚类，聚类结果如下图两种颜色。

```
library(kernlab)  
data(spirals)  
sc <- specc(spirals, centers=2, kernel= "rbfdot" ) # 高斯核  
plot(spirals, col=sc)
```



因为有300个样本的原始数据，我们不需要图或加权图进行可视化。图的概念在本例没有任何作用，

#igraph 中的数据集 karate 具有igraph特殊格式，需要用as_adj函数转化为（提取）一般的邻接矩阵

```
library(igraph)
```

```
library(igraphdata)
```

```
data(karate)
```

```
A=as_adj(karate) #提取karate中的邻接矩阵
```

```
A=as.matrix(A)
```

```
faction = vertex_attr(karate)$Faction #提取faction: 真实的派别标号
```

```
library(kernlab)
```

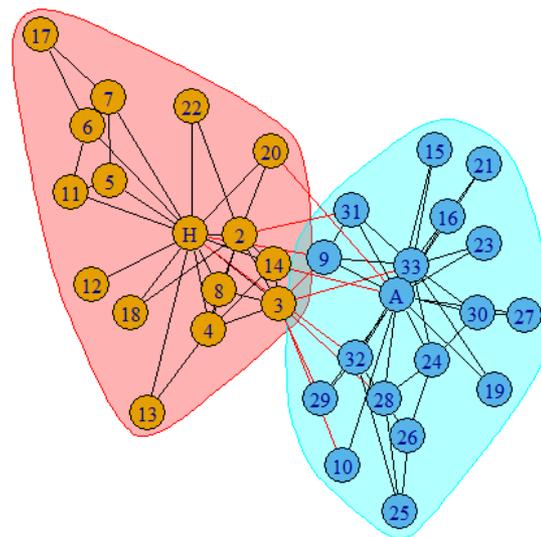
```
A=as.kernelMatrix(A)
```

```
clust =specc(A,centers=2) # 聚类
```

```
table(faction,clust) # 比较聚类结果与真实类别
```

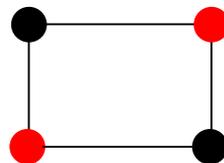
```
      clust
faction 1 2
      1 18 0
      2  1 15
```

只有一个人（Actor 3）分错了类。



0-1图的谱分析

偶数长度的环（可二染色）
会使得聚类算法出现困难。



矩阵的特征根称为谱（spectrum），基于特征根、特征向量的分析方法称为谱方法。这里我们针对0-1图进一步探讨标准邻接矩阵（或拉普拉斯矩阵）的特征根和特征向量，多数结论对加权图（一般的邻接矩阵）也成立。。

给定 $n \times n$ 邻接矩阵 $A = (a_{ij})$, $a_{ij} = 0, 1$, 定义度数 $d_i = \sum_{j=1}^n a_{ij}$, 度数矩阵 $D = \text{diag}(d_1, \dots, d_n)$, 定义Laplacian 矩阵

$$L = D - A。$$

某种意义上，Laplacian $L = D - A$ 与 $D^{-1}A$ 等价，比如 L 的最小特征根0的特征向量 \mathbf{x} 是 $B = D^{-1}A$ 的最大特征根1的特征向量

$$0 = L\mathbf{x} = (D - A)\mathbf{x} \Leftrightarrow D\mathbf{x} = A\mathbf{x} \Leftrightarrow D^{-1}A\mathbf{x} = \mathbf{x}$$

我们下面考虑 $B = D^{-1}A$ 的谱分析。

标准邻接矩阵

$B = D^{-1}A$ ：标准化的邻接矩阵，其每一行总和为1（是概率分布），因此也称为随机游走邻接矩阵。

特征根性质：

$$\lambda(AB) = \lambda(BA)$$

$$\lambda(I \pm A) = 1 \pm \lambda(A)$$

$$0 \leq A \leq I \Leftrightarrow 0 \leq \lambda(A) \leq 1$$

命题4. $B = D^{-1}A$ 的任意特征根都是实数，且 $|\lambda(B)| \leq 1$ ， $\lambda_{\max}(B) = 1$

证1：因为对 $\forall \mathbf{x} \in R^n$ ， $\mathbf{x}^T L \mathbf{x} = \sum a_{ij}(x_i - x_j)^2/2 \geq 0$ ，
 $\Rightarrow L = D - A \geq 0$ (半正定)， $\Rightarrow D^{-1/2} L D^{-1/2} = I_n - D^{-1/2} A D^{-1/2} \geq 0$
 $\Rightarrow 0 \leq \lambda(I_n - D^{-1/2} A D^{-1/2}) = 1 - \lambda(D^{-1/2} A D^{-1/2})$
 $\Rightarrow \lambda(D^{-1} A) = \lambda(D^{-1/2} L D^{-1/2}) \leq 1$ 。

类似地对任何 $\mathbf{x} \in R^n$ ， $\mathbf{x}^T (D + A) \mathbf{x} = \sum a_{ij}(x_i + x_j)^2/2 \geq 0$
 $\Rightarrow D + A \geq 0 \Rightarrow D^{-1/2} (D + A) D^{-1/2} = I_n + D^{-1/2} A D^{-1/2} \geq 0$
 $\Rightarrow 1 + \lambda(D^{-1/2} A D^{-1/2}) \geq 0 \Rightarrow \lambda(D^{-1} A) \geq -1$

证2：因为 $B = D^{-1}A$ 与 $D^{-1/2} A D^{-1/2}$ 有相同的特征根，后者是对称矩阵，特征根都是实数。假设特征根 $\lambda \triangleq \lambda(B)$ 的特征向量为 \mathbf{x} ， $\mathbf{x} \neq \mathbf{0}$ ，即 $B\mathbf{x} = \mathbf{x}\lambda$ 。记 $|x_k| = \max |x_i| > 0$ ，
由 $|\lambda x_k| = |(B\mathbf{x})_k| = |\sum_{j \sim k} x_j|/d_k \leq \sum_{j \sim k} |x_j|/d_k \leq |x_k|$ ，所以 $|\lambda| \leq 1$ 。因为 $B\mathbf{1} = D^{-1}A\mathbf{1} = \mathbf{1}$ ，故1是特征根， $\mathbf{1}$ 是对应的特征向量。

注：也可直接应用圆盘定理,证2的方法实际上就是圆盘定理的证明方法。

B-变换

假设 $\mathbf{x} = (x_1, \dots, x_n)^\top \in R^n$, x_i 代表节点的标号或某种属性, 比如类别、次序、信息、能量等等。线性变换 $\mathbf{y} = B\mathbf{x}$ 将属性 \mathbf{x} 在互相连结的节点之间重新分配 (流动) :

$$\mathbf{y} = B\mathbf{x} = D^{-1}A\mathbf{x} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)^\top, \quad \bar{x}_i = \frac{1}{d_i} \sum_{j: j \sim i} x_j$$

即 $y_i = \bar{x}_i =$ 与 i 邻接的节点属性的平均

幂次迭代: 反复B变换

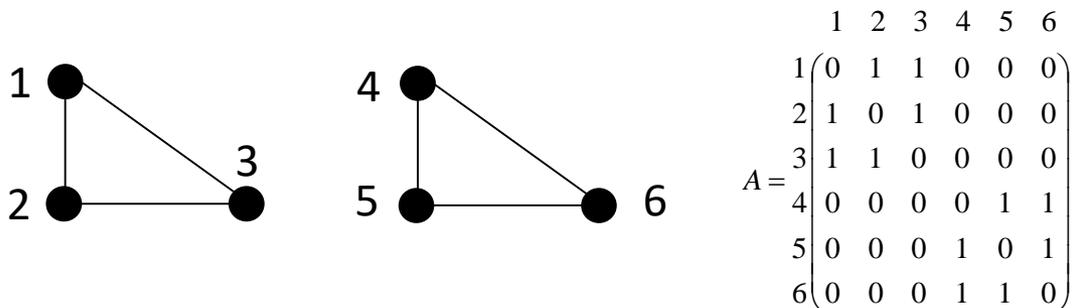
给定任何节点标号初值 \mathbf{x} , 反复做 B -变换 (幂次迭代) :

$$\mathbf{x}^{(0)} = \mathbf{x}, \quad \mathbf{x}^{(k)} = B\mathbf{x}^{(k-1)}, \quad k = 1, 2, \dots$$

将会使得连结紧密的节点子集的标号趋同, 而不连通的节点子集之间互不影响。

因此在理想情况下, 幂次迭代收敛到的最大特征根的特征向量具有 $(a, a, a, b, b, c, c, c, \dots)$ 的形式, 相同的字母/数值代表相应的节点属于同一个连通子集。

例如，下图由两个互不联通的子集 {1,2,3}, {4,5,6} 组成，



给定任何非0初值： $\mathbf{x} = (x_1, \dots, x_6)^\top$

$$B\mathbf{x} = D^{-1}A\mathbf{x} = \left(\frac{x_2 + x_3}{2}, \frac{x_1 + x_3}{2}, \frac{x_1 + x_2}{2}, \frac{x_5 + x_6}{2}, \frac{x_4 + x_6}{2}, \frac{x_4 + x_5}{2} \right)^\top$$

$$B^2\mathbf{x} = \left(\frac{2x_1 + x_2 + x_3}{4}, \frac{x_1 + 2x_2 + x_3}{4}, \frac{x_1 + x_2 + 2x_3}{4}, \frac{2x_4 + x_5 + x_6}{4}, \frac{x_4 + 2x_5 + x_6}{4}, \frac{x_4 + x_5 + 2x_6}{4} \right)^\top$$

$$B^3\mathbf{x} = \left(\frac{2x_1 + 3x_2 + 3x_3}{8}, \frac{3x_1 + 2x_2 + 3x_3}{8}, \frac{3x_1 + 3x_2 + 2x_3}{8}, \frac{2x_4 + 3x_5 + 3x_6}{8}, \frac{3x_4 + 2x_5 + 3x_6}{8}, \frac{3x_4 + 3x_5 + 2x_6}{8} \right)^\top$$

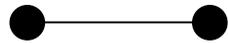
$$\rightarrow \mathbf{v}_1 = (a, a, a, b, b, b)^\top, \quad a = \frac{x_1 + x_2 + x_3}{3}, \quad b = \frac{x_4 + x_5 + x_6}{3}$$

特征向量 \mathbf{v}_1 表明存在两个互不连通的子集/类。

但幂次迭代也会出现不收敛的情况，例如只有两个邻接节点时，

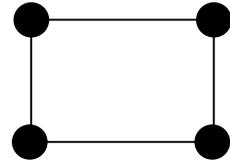
$$B = A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, B\mathbf{x} = \begin{pmatrix} x_2 \\ x_1 \end{pmatrix}, B^2\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, B^3\mathbf{x} = B\mathbf{x} = \begin{pmatrix} x_2 \\ x_1 \end{pmatrix}, \dots$$

即幂次变换在 $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} x_2 \\ x_1 \end{pmatrix}$ 之间震荡，不收敛。右下图也不收敛，



$$A = B = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

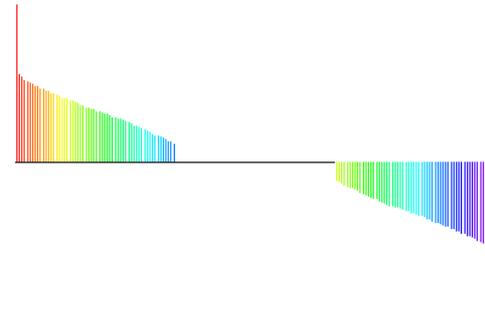
$$\lambda(B) = 1, -1$$



$$A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \end{matrix}$$

$$\lambda(B) = 1, 0, 0, -1$$

我们将会看到，当 B 特征根关于 0 对称（此时最大特征根为 1 ，最小特征根为 -1 ）时会出现不收敛情况。



幂次迭代的收敛性

对于一般方阵 C (未必对称),幂次迭代在一定条件下收敛到绝对值最大的特征根对应的特征向量。

命题5. 假设 $n \times n$ 矩阵 C 是可相似对角化的 $C = P\Lambda P^{-1}$, 其中 $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ 为特征根对角阵, $P = (\mathbf{v}_1, \dots, \mathbf{v}_n)$ 的列为特征向量。对任意 $\mathbf{x} \in R^n$, 记 $\mathbf{c} = P^{-1}\mathbf{x} = (c_1, \dots, c_n)^T$ 。

- (1) 假设 $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n| \geq 0$, $C^k \mathbf{x} / \lambda_1^k \rightarrow \mathbf{v}_1 c_1, k \rightarrow \infty$,
(即极限是特征向量 $\mathbf{v}_1 c_1$, 除非 $c_1 = 0$, 即 \mathbf{x} 恰好与 \mathbf{v}_1 无关).
- (2) 假设 $|\lambda_1| = \dots = |\lambda_m| > |\lambda_{m+1}| \geq \dots \geq |\lambda_n| \geq 0$, 且 $\lambda_1, \dots, \lambda_m$ 同号 (重根), 则 $C^k \mathbf{x} / \lambda_1^k \rightarrow \mathbf{v}_1 c_1 + \dots + \mathbf{v}_m c_m, k \rightarrow \infty$.
- (3) 若(2)中的 $\lambda_1, \dots, \lambda_m$ 不全同号, 则不收敛。

注: (3)说明, 如果矩阵 C 的最大特征根和最小特征根符号相反, 绝对值相同, 则幂次迭代不收敛。

证明:(1) 因为 $\Lambda^k / \lambda_1^k = \text{diag}(1, \lambda_2^k / \lambda_1^k, \lambda_3^k / \lambda_1^k, \dots) \rightarrow \text{diag}(1, 0, \dots, 0), k \rightarrow \infty,$

$$\text{所以 } C^k \mathbf{x} / \lambda_1^k = P \Lambda^k \mathbf{c} / \lambda_1^k \rightarrow (\mathbf{v}_1, \dots, \mathbf{v}_n) \begin{pmatrix} 1 & & \\ & 0 & \\ & & \ddots \\ & & & 0 \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} \rightarrow \mathbf{v}_1 c_1.$$

(2) $\Lambda^k / \lambda_1^k = \text{diag}(1^k, \dots, 1^k, \lambda_{m+1}^k / \lambda_1^k, \dots) \rightarrow \text{diag}(1, \dots, 1, 0, \dots, 0),$

$$C^k \mathbf{x} / \lambda_1^k = P \Lambda^k \mathbf{c} / \lambda_1^k \rightarrow (\mathbf{v}_1, \dots, \mathbf{v}_n) \begin{pmatrix} 1 & & \\ & 1 & \\ & & \ddots \\ & & & 0 \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} \rightarrow \mathbf{v}_1 c_1 + \dots + \mathbf{v}_m c_m$$

(3) 若 $\lambda_1, \dots, \lambda_m$ 绝对值相同, 但不全同号, 比如 λ_1 与其它符号相反, 则 Λ^k / λ_1^k 的第 $2, \dots, m$ 对角元为 $(-1)^k$, 不收敛。

当 $k \rightarrow \infty, C^k \mathbf{x} / \lambda_1^k = \mathbf{v}_1 c_1 + (-1)^k \mathbf{v}_1 c_1 + \dots + (-1)^k \mathbf{v}_m c_m + o(1)$

命题6.

- (1) $B = D^{-1}A$ 的所有特征根 λ 为实数, 且 $|\lambda| \leq 1$; 最大特征根为 $\lambda_1 = 1$;
- (2) $\lambda_1 = 1$ 的重数为1当且仅当图全连通的
- (3) $\lambda_1 = 1$ 的重数为 k 当且仅当图由 k 个互不连通的连通子图 G_1, \dots, G_k 构成, 则 $\lambda_1 = 1$ 的特征空间的基为 $\mathbf{v}_1, \dots, \mathbf{v}_k, v_{ij} = 1_{(i \in G_j)}, j = 1, \dots, k; i = 1, \dots, n$.

证明: 与命题2,3(laplacian情形)的证明几乎完全相同:

(2) 假设图是全连通的, 我们只需证明除了一个常数倍数之外, $\mathbf{1}_n$ 是 $\lambda_1 = 1$ 的唯一特征向量。设 \mathbf{x} 是 B 的对应于 $\lambda_1 = 1$ 的特征向量,

$$\begin{aligned}
 B\mathbf{x} = \mathbf{x} &\Leftrightarrow D^{-1}A\mathbf{x} = \mathbf{x} \Leftrightarrow A\mathbf{x} = D\mathbf{x} \Leftrightarrow (D - A)\mathbf{x} = \mathbf{0} \Rightarrow 0 = \mathbf{x}^T(D - A)\mathbf{x} \\
 &= \mathbf{x}^T L\mathbf{x} = \frac{1}{2} \sum_{i \sim j} (x_i - x_j)^2 \Rightarrow x_1 = \dots = x_n \Rightarrow \mathbf{x} \propto \mathbf{1}_n.
 \end{aligned}$$

(3) 假设图由 k 个互斥的连通子图 G_1, \dots, G_k 构成, 重排邻接矩阵

$$A = \begin{pmatrix} A_1 & & \\ & \ddots & \\ & & A_k \end{pmatrix}, \text{ 记 } \mathbf{v}_1 = \begin{pmatrix} \mathbf{1}_{G_1} \\ \vdots \\ \mathbf{0} \end{pmatrix}, \dots, \mathbf{v}_k = \begin{pmatrix} \mathbf{0} \\ \vdots \\ \mathbf{1}_{G_k} \end{pmatrix}.$$

设 \mathbf{x} 是 B 的对应于 $\lambda_1 = 1$ 的特征向量, $B\mathbf{x} = \mathbf{x} \Leftrightarrow (D - A)\mathbf{x} = \sum_{i \sim j} (x_i - x_j)^2 / 2 = 0$

\Rightarrow 对每个子图 $\sum_{i \sim j, i, j \in G_s} (x_i - x_j)^2 = 0$, 即每个子图上的节点对应的 x 's全部相等。

$\Rightarrow \mathbf{x}$ 是 $\mathbf{v}_1, \dots, \mathbf{v}_k$ 的线性组合.

由(2), 除了一个常数倍数之外, $\mathbf{1}_{G_s}$ 是 A_s 对应于特征根1的唯一特征向量, 从而 $\mathbf{v}_1, \dots, \mathbf{v}_k$ 都是 A 的对应于特征根1的特征向量, 故 $\lambda_1 = 1$ 的重数为 k .

命题7 (幂次迭代不收敛情形) 假设图是全连通的, 则下述陈述等价:

- (a) B 有特征根 -1 (幂次迭代不收敛);
- (b) 二分图: 节点可由两种颜色染色, 使得邻接节点颜色不同;
- (c) 没有奇数长度的环;
- (d) B 的所有特征根(谱)关于0对称。

证明: (a) \Leftrightarrow (b)

设 $\mathbf{x} \neq \mathbf{0}$ 是 B 的对应于 $\lambda = -1$ 的特征向量, $B\mathbf{x} = -\mathbf{x} \Leftrightarrow D^{-1}A\mathbf{x} = -\mathbf{x}$

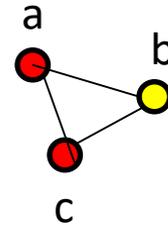
$$\Leftrightarrow (D + A)\mathbf{x} = \mathbf{0} \Rightarrow 0 = \mathbf{x}^T (D + A)\mathbf{x} = \frac{1}{2} \sum_{i \sim j} (x_i + x_j)^2$$

\Rightarrow 若 $i \sim j$, 则 $x_i = -x_j$, 所以相邻的节点或者符号相反, 或者都为0。

因为图是全连通的, 任何两个节点之间都有一条路径连结它们, 所以所有 x_i 绝对值相同且非0, 相邻的节点符号相反。

(c) \Leftrightarrow (b)

如果存在奇数环(右图), 那么不可能用两种颜色染色使得相邻节点的颜色不同。



(a) \Leftrightarrow (d)

当 -1 是 B 的特征根时, 我们可以将节点重新排列, 使得前面 m 个都是同一种颜色 (互不连结), 后面 $n - m$ 个是另外一种颜色 (互不连结), 但两种颜色间有连结。

邻接矩阵如下

$$A = \begin{pmatrix} 0 & C \\ C^\top & 0 \end{pmatrix} \Rightarrow B = D^{-1}A = \begin{pmatrix} 0 & B_1 \\ B_2 & 0 \end{pmatrix}$$

假设 λ 是 B 的一个特征根，对应的特征向量为 $\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix}$ ，记 $\mathbf{y} = \begin{pmatrix} \mathbf{x}_1 \\ -\mathbf{x}_2 \end{pmatrix}$

由 $B\mathbf{x} = \mathbf{x}\lambda$ ，即 $\begin{pmatrix} 0 & B_1 \\ B_2 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} B_1\mathbf{x}_2 \\ B_2\mathbf{x}_1 \end{pmatrix} = \begin{pmatrix} \lambda\mathbf{x}_1 \\ \lambda\mathbf{x}_2 \end{pmatrix}$ 。令 $\mathbf{y} = \begin{pmatrix} \mathbf{x}_1 \\ -\mathbf{x}_2 \end{pmatrix}$ ，则

$$B\mathbf{y} = \begin{pmatrix} 0 & B_1 \\ B_2 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ -\mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} -B_1\mathbf{x}_2 \\ B_2\mathbf{x}_1 \end{pmatrix} = \begin{pmatrix} -\lambda\mathbf{x}_1 \\ \lambda\mathbf{x}_2 \end{pmatrix} = -\lambda\mathbf{y}$$

所以 $-\lambda$ 也是 B 的特征根，对应的特征向量为 $\mathbf{y} = \begin{pmatrix} \mathbf{x}_1 \\ -\mathbf{x}_2 \end{pmatrix}$ 。

所以 B 的所有特征根关于0对称。证毕。

二分图(bipartite)

- \Leftrightarrow 两种着色，相邻的点不同色（左图）
- $\Leftrightarrow \lambda_{\min}(B) = -1$
- $\Leftrightarrow B$ 的特征根关于0对称。

