# 《计算机应用数学第二次作业》

# 截止时间：2024.05.08

## 计算题（80 分）

1. （30 分）给定图 G（如图 1，图中数值为边权值），图切割将其分割成多个互不连通的子图。请使用谱聚类算法将图 G 聚类成 k = 2 类，使得：
   (a) RatioCut 最小；
   (b) NormalizedCut 最小。
   并给出计算过程，以及对应的 RatioCut 值和 NormalizedCut 值

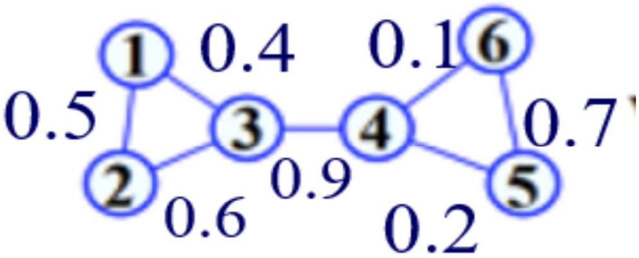



图 1

2. （10 分）参数未知的正态分布生成了如下 4 个数据：**-12.2119174, 7.62328757, 9.60884573, -8.36968016。**请使用极大似然估计方法估计出正态分布的参数，写出计算过程。（注：若数值计算过程较为复杂，可以使用 python 相关库进行计算，但仍需给出计算推导过程。）

3. （20 分）HITS 全称是 Hyperlink-Induced Topic Search，即超链接诱导主题搜索，是一个链接分析方法，由康奈尔大学的 Kleinberg 教授提出。HITS 算法中有 2 个比较重要的概念，一个是“Authority”页面，另一个是“Hub”页面。“Authority”，即权威，所有权威页面，是指页面本身的质量比较高，比如，百度搜索或者谷歌搜索首页。“Hub”，即枢纽，表示本页面指向的其他很多高质量的“Authority”页面。一个好的“Authority”会被很多好的“Hub”页面指向。一个好的“Hub”页面会指向很多好的“Authority”页面。权威值和枢纽值是互相依存、互相影响的。网页的权威值是所有指向它的网页的枢纽值之和：

$$auth(p) = \sum_{q\in\{指向p的网页\}} hub(q)$$

网页的枢纽值是所有它指向网页的权威值之和：

$$hub(p) = \sum_{q\in\{p指向的网页\}} auth(q)$$

HITS 算法流程如下:

1. 初始化：将各节点的权威值和枢纽值均设为 1。
2. 更新节点的权威值;
3. 更新节点的枢纽值;
4. 将权威值和枢纽值归一化，即权威值和枢纽值分别和为 1;
5. 重复 2-4 步骤，直至最终收敛。

(a) 根据如图 2 所示的网络关系，每个节点的 PageRank 初始值都是 1
写出 PageRank 算法前两轮迭代的过程。

(b) 根据如图 2 所示的网络关系，每个节点的初始权威值和枢纽值均设为 1，写出 HITS 算法前两轮迭代的过程。

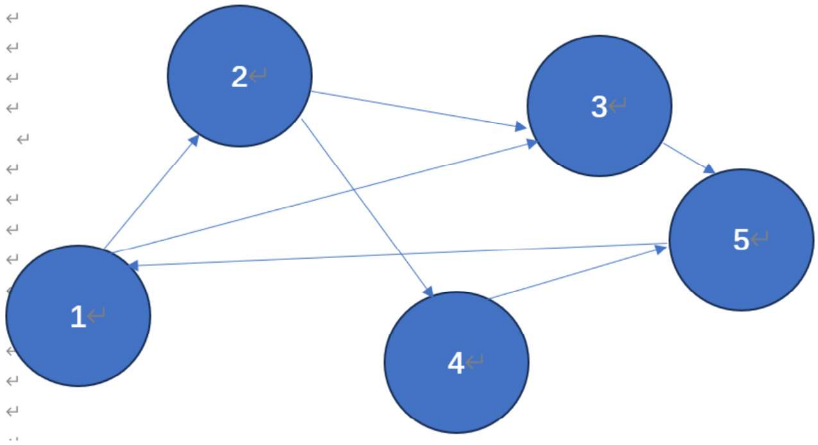



图 2

4. (10 分) 假设机器人必须越过迷宫并到达终点。有地雷，机器人一次只能移动一个地砖。如果机器人踏上地雷，机器人就死了。机器人必须在尽可能短的时间内到达终点。请给出第 1 轮之后和第 2 轮之后 Q-table 中每个位置的值。并对第一列的所有值写出计算过程。学习率设为 0.2，折扣因子为 0.1。

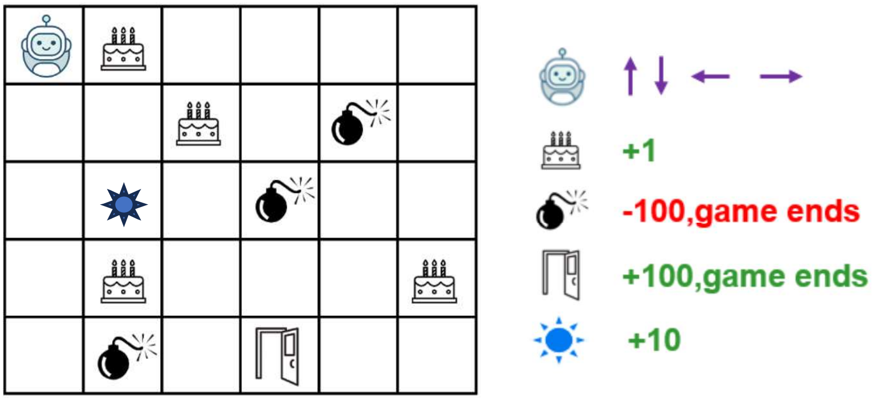



图 3

5. (10 分) 假设有四个观测点,分别为(-4, -0.55), (1, 0.45) , (-0.5, 0.22)和(5, 1.25)，用最小二乘法求解直线$y = ax + b$使其最佳拟合这些散点。 给出求解过程。

## 编程题 （20 分）

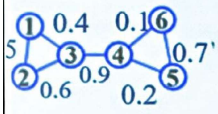
说明：建议使用开源工具包，例如 scikit-learn 中有朴素贝叶斯、高斯混合模型等函数实现，sknetwork 中有 PageRank 函数实现。

**Node2vec**

数据集：PageRank_Dataset.csv。
任务描述：利用 Node2vec 计算每个节点的 embedding 值。
要求输出：1）每个节点的 embedding 值列表（csv 文件）；2）随机挑选 10 个 node pair，对比他们在 embedding 上的相似度和在 betweenness centrality 上的相似度（使用 Jaccard similarity）。

(如图 1、图中数值为边权值)，图切割将其分割成多个互不连通的
算法将图 G 聚类成 k = 2 类，使得：

最小。
及对应的 RatioCut 值和 NormalizedCut 值




、度矩阵、拉普拉斯矩阵

权重矩阵 $W$:

$$W=\begin{bmatrix}0&0.5&0.4&0&0&0\\0.5&0&0.6&0&0&0\\0.4&0.6&0&0.9&0&0\\0&0&0.9&0&0.2&0.1\\0&0&0&0.2&0&0.7\\0&0&0&0.1&0.7&0\end{bmatrix}$$

度矩阵 $D$:

$$D=\begin{bmatrix}0.9&0&0&0&0&0\\0&1.1&0&0&0&0\\0&0&1.9&0&0&0\\0&0&0&1.2&0&0\\0&0&0&0&0.9&0\\0&0&0&0&0&0.8\end{bmatrix}$$

拉普拉斯矩阵 $L$:

$$L=D-W=\begin{bmatrix}0.9&-0.5&-0.4&0&0&0\\-0.5&1.1&-0.6&0&0&0\\-0.4&-0.6&1.9&-0.9&0&0\\0&0&-0.9&1.2&-0.2&-0.1\\0&0&0&-0.2&0.9&-0.7\\0&0&0&-0.1&-0.7&0.8\end{bmatrix}$$

## RatioCut 切图

***k*个子图：** $RatioCut(A_1,A_2,\ldots A_k)=\frac{1}{2}\sum_{i=1}^{k}\frac{W(A_i,\bar{A}_i)}{|A_i|}$

$$\begin{aligned}RatioCut(A_1,A_2,\ldots A_k)&=\sum_{i=1}^{k}h_i^TLh_i\\&=\sum_{i=1}^{k}(H^TLH)_{ii}\\&=\operatorname{tr}(H^TLH)\end{aligned}$$

$$\underset{H}{\operatorname{argmin}}\quad \operatorname{tr}(H^TLH)$$
$$\text{s.t. } H^TH=I$$

$$RatioCut(A_1,A_2,\ldots,A_k)=\frac{1}{2}\sum_{i=1}^{k}\frac{W(A_i,\bar{A}_i)}{|A_i|}=tr(H^TLH)$$

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

$$A_1=\{1,2,3,4\},\ A_2=\{5,6\}$$

$$RatioCut(A_1,A_2)=0.1125$$

## NCut 切图

$$NCut(A_1,A_2,\ldots A_k)=\frac{1}{2}\sum_{i=1}^{k}\frac{W(A_i,\bar{A}_i)}{vol(A_i)}$$

$$\begin{aligned}\mathrm{NCut}(A_1,A_2,\cdots,A_k)&=\sum_{i=1}^{k}h_i^TLh_i\\&=\sum_{i=1}^{k}(H^TLH)_{ii}\\&=\operatorname{tr}(H^TLH)\end{aligned}$$

$$H^TH\neq I\text{，而是 }H^TDH=I$$

**优化目标：**

$$\arg\min_H tr(H^TLH)$$
$$s.t.\quad H^TDH=I$$

令 $H=D^{-1/2}F$，$H^TLH=F^TD^{-1/2}LD^{-1/2}F$，$H^TDH=F^TF=I$，

$$\arg\min_F tr(F^TD^{-1/2}LD^{-1/2}F)$$
$$s.t.\quad F^TF=I$$

问题变成了求矩阵 $D^{-1/2}LD^{-1/2}$的$k$个最小的特征值

得到特征向量 $F$ 将 $F$当成样本送入 Kmeans 聚类

$$NormalizedCut(A_1,A_2,\ldots,A_k)=\frac{1}{2}\sum_{i=1}^{k}\frac{W(A_i,\bar{A}_i)}{vol(A_i)}$$

$$A_1=\{1,2,3,4\},\ A_2=\{5,6\}$$

$$NormalizedCut(A_1,A_2)\approx 0.235$$

2. （10 分）参数未知的正态分布生成了如下 4 个数据：-12.2119174, 7.62328757, 9.60884573, -8.36968016。请使用极大似然估计方法估计出正态分布的参数，写出计算过程。（注：若数值计算过程较为复杂，可以使用 python 相关库进行计算，但仍需给出计算推导过程。）

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$L(\mu,\sigma^2|x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\sum_{i=1}^{n}(x_i-\mu)^2}$$

$$\ln L(\mu,\sigma^2|x) = -n \ln \sqrt{2\pi\sigma^2} - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2$$

$$\frac{\partial}{\partial\mu} \ln L(\mu,\sigma^2|x) = \frac{1}{2\sigma^2} 2 \sum_{i=1}^{n}(x_i-\mu) = 0$$

$$\frac{\partial}{\partial\sigma} \ln L(\mu,\sigma^2|x) = -\frac{n}{\sigma} + \frac{1}{\sigma^3} 2 \sum_{i=1}^{n}(x_i-\mu)^2 = 0$$

$$\hat{\mu} = \frac{x_1 + x_2 + \ldots + x_n}{n} = \bar{x} \approx -0.837$$

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2 \approx 91.706$$

(a) 根据如图 2 所示的网络关系，每个节点的 PageRank 初始值都是 1
写出 PageRank 算法前两轮迭代的过程。

$$P = \begin{bmatrix} 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$PR_0 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

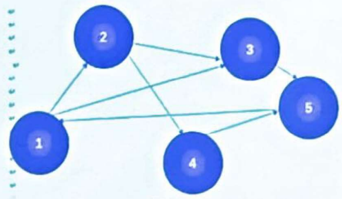


图 2

$$PR_1 = PR_0 * P = \begin{bmatrix} 1 & 0.5 & 1 & 0.5 & 2 \end{bmatrix}$$

$$PR_2 = PR_1 * P = \begin{bmatrix} 2 & 0.5 & 1.5 & 0.25 & 1.5 \end{bmatrix}$$

和枢纽值是互相依存、互相影响的。网页的权威值是所有指向它的网页的枢纽值之和：

$$auth(p) = \sum_{q\in(\text{指向}p\text{的网页})} hub(q)$$

网页的枢纽值是所有它指向网页的权威值之和：

$$hub(p) = \sum_{q\in(p\text{指向的网页})} auth(q)$$

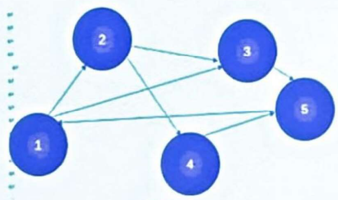


图 2

HITS 算法流程如下：

1. 初始化：将各节点的权威值和枢纽值均设为 1。
2. 更新节点的权威值；
3. 更新节点的枢纽值；
4. 将权威值和枢纽值归一化，即权威值和枢纽值分别和为 1；
5. 重复 2-4 步骤，直至最终收敛。

(a) 根据如图 2 所示的网络关系，每个节点的 PageRank 初始值都是 1
写出 PageRank 算法前两轮迭代的过程。

(b) 根据如图 2 所示的网络关系，每个节点的初始权威值和枢纽值均设为 1，写出 HITS 算法前两轮迭代的过程。

矩阵 $Q$：

$$Q = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}, Q^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

$$Auth_0 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$Hub_0 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$Auth_1 = Hub_0 * Q = \begin{bmatrix} 1 & 1 & 2 & 1 & 2 \end{bmatrix}$$

$$Hub_1 = Auth_1 * Q^T = \begin{bmatrix} 3 & 3 & 2 & 2 & 1 \end{bmatrix}$$

矩阵 $Q$：

$$Q = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}, Q^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

$$Auth_0 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$Hub_0 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$Auth_1 = Hub_0 * Q = \begin{bmatrix} 1 & 1 & 2 & 1 & 2 \end{bmatrix}$$

$$Hub_1 = Auth_1 * Q^T = \begin{bmatrix} 3 & 3 & 2 & 2 & 1 \end{bmatrix}$$

归一化：

$$Auth_1 = \frac{1}{7}\begin{bmatrix} 1 & 1 & 2 & 1 & 2 \end{bmatrix}$$

$$Hub_1 = \frac{1}{11}\begin{bmatrix} 3 & 3 & 2 & 2 & 1 \end{bmatrix}$$

$$Auth_2 = Hub_1 * Q = \frac{1}{11}\begin{bmatrix} 1 & 3 & 6 & 3 & 4 \end{bmatrix}$$

$$Hub_2 = Auth_2 * Q^T = \frac{1}{11}\begin{bmatrix} 9 & 9 & 4 & 4 & 1 \end{bmatrix}$$

归一化：

$$Auth_2 = \frac{1}{17}\begin{bmatrix} 1 & 3 & 6 & 3 & 4 \end{bmatrix}$$

$$Hub_2 = \frac{1}{27}\begin{bmatrix} 9 & 9 & 4 & 4 & 1 \end{bmatrix}$$

假设机器人必须越过迷宫并到达终点。有地雷，机器人一次只能移动一个地砖。
器人踏上地雷，机器人就死了。机器人必须在尽可能短的时间内到达终点。请给
轮之后和第 2 轮之后 Q-table 中每个位置的值。并对第一列的所有值写出计算过
习率设为 0.2，折扣因子为 0.1。

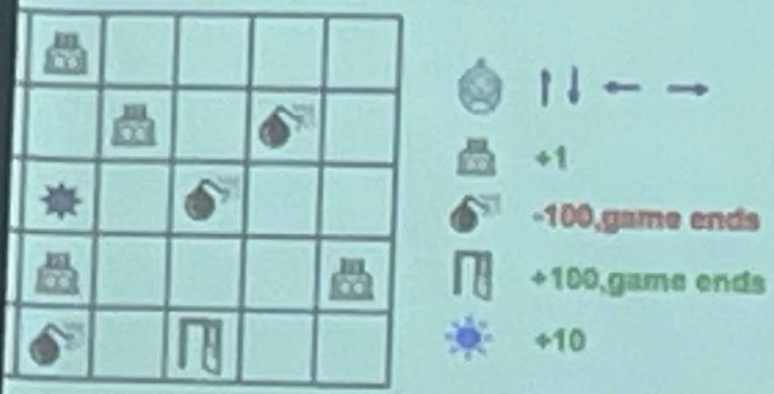



图 3

| State | ↑ | ↓ | ← | → |
|---|---|---|---|---|
| Start | 0 | 0 | 0 | 0 |
| Blank | 0 | 0 | 0 | 0 |
| Cake | 0 | 0 | 0 | 0 |
| Boom | 0 | 0 | 0 | 0 |
| Exit | 0 | 0 | 0 | 0 |
| Sun | 0 | 0 | 0 | 0 |

表 1: Q-table-0

$$Q_{new}(S_t, a_t) \leftarrow Q(S_t, a_t) + \alpha \left[ r + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, a_t) \right]$$

$$\alpha = 0.2, \ \gamma = 0.1$$

| State | ↑ | ↓ | ← | → |
|---|---|---|---|---|
| Start | 0 | 0 | 0 | 0.2 |
| Blank | 0 | 0 | 0 | 0 |
| Cake | 0 | 0 | 0 | 0 |
| Boom | 0 | 0 | 0 | 0 |
| Exit | 0 | 0 | 0 | 0 |
| Sun | 0 | 0 | 0 | 0 |

表 2: Q-table-1

$$Q_{new}(Cake, \downarrow) \leftarrow 0 + 0.2[-1 + 0.1 \times 0 - 0]$$

| State | ↑ | ↓ | ← | → |
|---|---|---|---|---|
| Start | 0 | 0 | 0 | 0.2 |
| Blank | 0 | 0 | 0 | 0 |
| Cake | 0 | -0.2 | 0 | 0 |
| Boom | 0 | 0 | 0 | 0 |
| Exit | 0 | 0 | 0 | 0 |
| Sun | 0 | 0 | 0 | 0 |

表 3: Q-table-2

$$Q_{new}(Cake, \downarrow) \leftarrow 0 + 0.2[-1 + 0.1 \times 0 - 0]$$

- 最小二乘法求解，相当于寻找$a, b$。使得如下公式尽量成立

$$\begin{cases} y_1 = ax_1 + b \\ y_2 = ax_2 + b \\ \vdots \\ y_n = ax_n + b \end{cases}$$

- 可建立优化模型如下：

$$\min_{a,b} f(a,b) = \sum_{i=1}^{n} (ax_i + b - y_i)^2$$

- 对自变量a, b求导，并令其为0：

$$\begin{cases} \dfrac{\partial f}{\partial a} = 2\sum_{i=1}^{n} (ax_i + b - y_i)x_i = 0 \\ \dfrac{\partial f}{\partial b} = 2\sum_{i=1}^{n} (ax_i + b - y_i) = 0 \end{cases}$$

- 化简可得解为：

$$\begin{cases} a = \dfrac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} = 0 \\ b = \dfrac{1}{n}\sum_{i=1}^{n} y_i - \dfrac{a}{n}\sum_{i=1}^{n} x_i \end{cases}$$

$$a = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \approx 0.20$$

$$b = \bar{y} - a\bar{x} \approx 0.27$$