深 圳 大 学 实 验 报 告

课程名称：信息检索

实验项目名称：链接分析的实验

学院：计算机与软件学院

专业：计算机科学与技术

指导教师：潘微科

报告人：沈晨珀　学号：2019092121　班级：19 计科 04

实验时间：2022 年 5 月 27 日（周五）-2022 年 6 月 8 日（周三）

实验报告提交时间：2022 年 5 月 31 日星期二

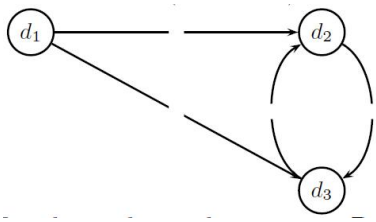
教务部制

实验目的与要求：

实验目的：掌握 PageRank、HITS 等经典的链接分析算法。

实验要求：

(1). 阅读教材《Introduction to Information Retrieval》第 464-470 页 21.2 节中所描述的 PageRank 计算方法（通过 power iteration 方式来实现），用 Java 语言或其他常用语言实现该算法。要求以下图所示的结构为例计算每个 document 的 PageRank 值，其中 teleportation rate=0.05。

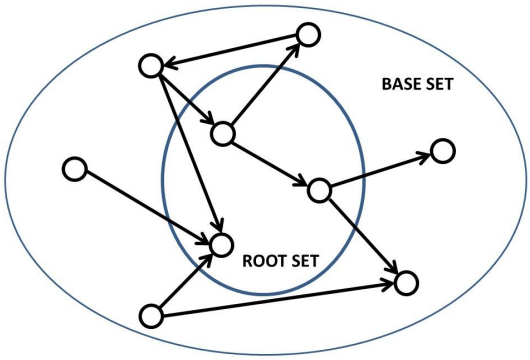


此外，关于 PageRank 算法，谈谈你的理解，并通过类比、关联或演绎的方式，举一个在日常的学习生活中可以应用的例子（要求积极向上且能自圆其说）。

请在报告中附上代码截图（不要复制源代码，请用截图的方式）、每次迭代的中间结果截图、最后运行结果截图和详细的文字说明。程序要有详细的注释。（30 分）

(2). 以另一种方式（不是 power iteration 方式）用笔算（不用程序计算）题(1)中每个 document 的 PageRank 值。要求有详细的说明和计算过程。（10 分）

(3). 阅读教材《Introduction to Information Retrieval》第 474-477 页 21.3 节中所描述的 HITS 计算方法（通过 power iteration 方式来实现），用 Java 语言或其他常用语言实现该算法。要求以下图所示的结构为例计算每个 document 的 authority 值和 hub 值。



此外，关于 HITS 算法，谈谈你的理解，并通过类比、关联或演绎的方式，举一个在日常的学习生活中可以应用的例子（要求积极向上且能自圆其说）。

请在报告中附上代码截图（不要复制源代码，请用截图的方式）、每次迭代的中间结果截图、最后运行结果截图和详细的文字说明。程序要有详细的注释。（40 分）

报告写作。要求：主要思路有明确的说明，重点代码有详细的注释，行文逻辑清晰、可读性强，报告整体写作较为专业。（20 分）

说明：

（1）本次实验课作业满分为 100 分。

（2）本次实验课作业截至时间 2022 年 6 月 8 日（周三）22:00。

（3）报告正文：请在指定位置填写，本次实验需要单独提交源程序文件（源程序单

独打包在 Blackboard 中上传，不要包含外部导入的包）。

（4）个人信息：WORD 文件名中的“姓名”、“学号”，请改为你的姓名和学号；实验报告的首页，请准确填写“学院”、“专业”、“报告人”、“学号”、“班级”、“实验报告提交时间”等信息。

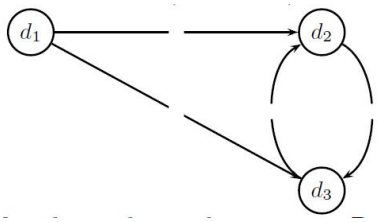
（5）提交方式：截至时间前，请在 Blackboard 平台中提交。

（6）发现抄袭（包括复制&粘贴整句话、整张图），**抄袭者和被抄袭者的成绩记零分**。

（7）延迟提交，不得分；如有特殊情况，请于截至日期之后的 48 小时内发邮件到 panweike@szu.edu.cn，并在邮件中注明课程名称、作业名称、姓名、学号等信息，以及特殊情况的说明，我收到后会及时回复。

（8）期末考试阶段补交无效。

(1). 阅读教材《Introduction to Information Retrieval》第 464-470 页 21.2 节中所描述的 PageRank 计算方法（通过 power iteration 方式来实现），用 Java 语言或其他常用语言实现该算法。要求以下图所示的结构为例计算每个 document 的 PageRank 值，其中 teleportation rate=0.05。



此外，关于 PageRank 算法，谈谈你的理解，并通过类比、关联或演绎的方式，举一个在日常的学习生活中可以应用的例子（要求积极向上且能自圆其说）。

请在报告中附上代码截图（不要复制源代码，请用截图的方式）、每次迭代的中间结果截图、最后运行结果截图和详细的文字说明。程序要有详细的注释。（30 分）

预先设定一些程序参数：

```
# 节点个数，转移概率
n, alpha = 3, 0.05
epsilon, max_iter = 1e-6, 200
```

根据题目中给定的图创建邻接矩阵：

```
def addEdge(graph, u, v):
    graph[u - 1][v - 1] = 1
```

```
# 首先生成一个n*n的矩阵，全部初始化为0
linkMatrix = np.zeros((n, n))

# 根据图添加边
addEdge(linkMatrix, 1, 2)
addEdge(linkMatrix, 1, 3)
addEdge(linkMatrix, 2, 3)
addEdge(linkMatrix, 3, 2)
```

对于此题，邻接矩阵如下所示：
linkMatrix[i][j]=1 说明有一条从节点 i 指向节点 j 的有向边。

0	1	1
0	0	1
0	1	0

然后开始计算转移概率矩阵：
一共三步：
1. 用每行中的 1 的个数取出每个 1
2. 处理后的结果矩阵乘以 1-α
3. 对上面得到的矩阵中的每个元素都加上 alpha/N

```
transitionProbabilityMatrix = np.array(
    [line / np.sum(line) for line in linkMatrix])
transitionMatrixWithTeleporting = transitionProbabilityMatrix * \
    (1 - alpha) + alpha / n
```

最终可以得到本体对应的转移概率矩阵：

0.016	0.491	0.491
0.016	0.016	0.966
0.016	0.966	0.016

进行幂迭代法：
初始化概率分布向量：

$$\vec{x_0} = (1/n, 1/n, 1/n)$$

然后根据如下公式进行迭代，直到概率分布向量收敛：

$$\vec{x_{n+1}} = \vec{x_n} P$$

```
# 初始化起始状态
x = np.array([1/n] * n)
print(f'x = {x}')

for i in range(1, max_iter):
    x_temp = np.dot(x, transitionMatrixWithTeleporting)
    print(f'xP{i} = {np.around(x_temp, decimals=3)}')
    if np.sum(np.abs(x - x_temp)) < epsilon:
        break
    else:
        x = x_temp
```

最终计算结果如下所示：迭代一次后即可收敛

```
x = [0.33333333 0.33333333 0.33333333]
xP1 = [0.017 0.492 0.492]
xP2 = [0.017 0.492 0.492]
```

即 Pagerank(d1)=0.017，Pagerank(d2)=0.492，Pagerank(d3)=0.492。简单分析可知，d2 与 d3 是对称的。同时由于没有 document 指向 d1，只有当遇到随机跳转时会跳转到 document1，所以 Pagerank(d1)会明显小于另外两个值。

PageRank 是一种简单有效且流行的网页排序算法，它通过一个网页的所有入链数目来计算该网页的重要性，其思想类似于一篇论文被引用的次数越大，该论文的影响力越大。也就是说一个网页的影响力不仅仅在于其自身的内容，来自其他网页的跳转链接数，同样可以在很大程度上应该网页的重要性排序。

(2). 以另一种方式（不是 power iteration 方式）用笔算（不用程序计算）题(1)中每个 document 的 PageRank 值。要求有详细的说明和计算过程。（10 分）

可以根据代数算法进行计算：

PageRank 的定义式：

$$\vec{x} = (1-\alpha)M\vec{x} + \frac{\alpha}{n}1$$

于是：

$$\vec{x}[I-(1-\alpha)M] = \frac{\alpha}{N}1$$

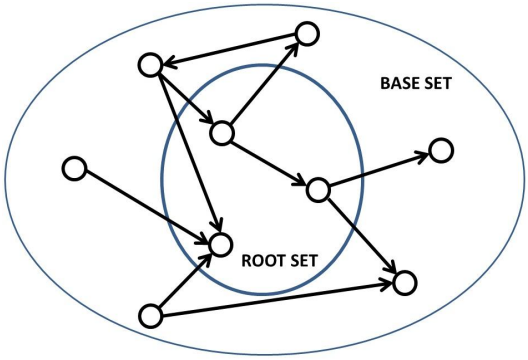
$$\vec{x} = \frac{\alpha}{N}1[I-(1-\alpha)M]^{-1}$$

其中 M 为转移概率矩阵（无随机跳转），1 为[1*N]的全一矩阵，I 为单位矩阵

$$\vec{x} = \frac{0.05}{3} * [1,1,1] * [\begin{bmatrix}1,0,0\\0,1,0\\0,0,1\end{bmatrix} - (1-0.05) * \begin{bmatrix}0,0.5,0.5\\0,\ 0,\ \ 1\\0,\ 1,\ \ 0\end{bmatrix}]^{-1}$$

由此可以得到$\vec{x} = [0.017, 0.492, 0.492]$，Pagerank(d1)=0.017，Pagerank(d2)=0.492，Pagerank(d3)=0.492，结果同幂迭代法得到结果。

(3). 阅读教材《Introduction to Information Retrieval》第 474-477 页 21.3 节中所描述的 HITS 计算方法（通过 power iteration 方式来实现），用 Java 语言或其他常用语言实现该算法。要求以下图所示的结构为例计算每个 document 的 authority 值和 hub 值。



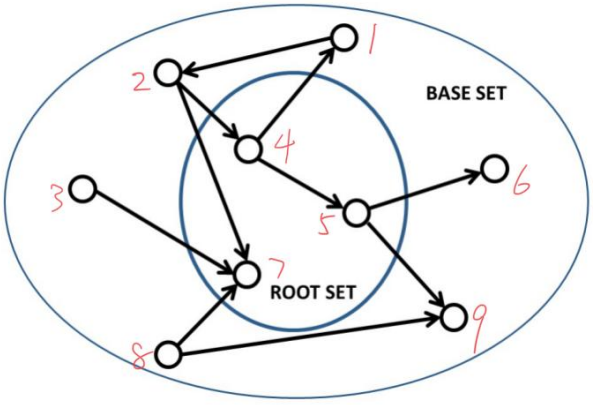
此外，关于 HITS 算法，谈谈你的理解，并通过类比、关联或演绎的方式，举一个在日常的学习生活中可以应用的例子（要求积极向上且能自圆其说）。

请在报告中附上代码截图（不要复制源代码，请用截图的方式）、每次迭代的中间结果截图、最后运行结果截图和详细的文字说明。程序要有详细的注释。（40 分）

预先设定一些程序参数：

```
# 节点个数，转移概率
n = 9
epsilon, max_iter = 1e-6, 200
```

然后对图中的节点进行标号，并生成对应的邻接矩阵，如下所示：



```
# 首先生成一个n*n的矩阵，全部初始化为0
linkMatrix = np.zeros((n,n))

# 根据图添加边
addEdge(linkMatrix, 1, 2)
addEdge(linkMatrix, 2, 4)
addEdge(linkMatrix, 2, 7)
addEdge(linkMatrix, 3, 7)
addEdge(linkMatrix, 4, 1)
addEdge(linkMatrix, 4, 5)
addEdge(linkMatrix, 5, 6)
addEdge(linkMatrix, 5, 9)
addEdge(linkMatrix, 8, 7)
addEdge(linkMatrix, 8, 9)
```

邻接矩阵：

```
[[0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 1. 0. 0. 1. 0. 0.]
 [0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [1. 0. 0. 0. 1. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 1. 0. 0. 1.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 1. 0. 1.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0.]]
```

初始化 hub 以及 authority 向量：

```
# 设置hub和authority初始化
h = np.array([1 / n] * n)
a = np.array([0] * n)
```

根据如下公式开始迭代，同时每一次迭代过后需要对于向量进行归一化处理，直到 hub 与 authority 向量收敛：

$$\vec{a} = A^T \vec{h}$$

$$\vec{h} = A\vec{a}$$

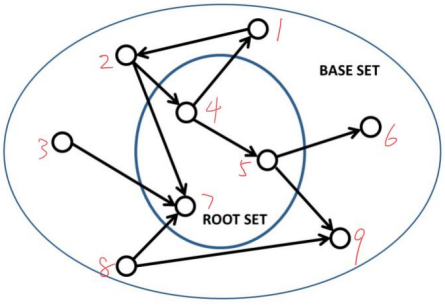
```python
for i in range(1, max_iter):
    # 迭代
    a_temp = np.dot(linkMatrix.T, h)
    h_temp = np.dot(linkMatrix, a_temp)

    # 归一化
    a_temp = a_temp / np.sum(a_temp)
    h_temp = h_temp / np.sum(h_temp)

    if np.sum(np.abs(h - h_temp)) + np.sum(np.abs(a - a_temp)) < epsilon:
        break
    else:
        h, a = h_temp, a_temp
        print(f'h{i} = {np.around(h, decimals = 3)}\na{i} = {np.around(a, decimals = 3)}\n')
```

最终运行结果如下所示：

hub	节点 1	节点 2	节点 3	节点 4	节点 5	节点 6	节点 7	节点 8	节点 9
h0	0.1111	0.1111	0.1111	0.1111	0.1111	0.1111	0.1111	0.1111	0.1111
h1	0.0556	0.2222	0.1667	0.1111	0.1667	0	0	0.2778	0
h2	0.0156	0.25	0.1875	0.0625	0.1719	0	0	0.3125	0
h3	0.0041	0.2645	0.1983	0.0331	0.1736	0	0	0.3264	0
...									
h21	0	0.2798	0.2091	0	0.1729	0	0	0.3383	0

authority	节点 1	节点 2	节点 3	节点 4	节点 5	节点 6	节点 7	节点 8	节点 9
a0	0	0	0	0	0	0	0	0	0
a1	0.1	0.1	0	0.1	0.1	0.1	0.3	0	0.2
a2	0.0625	0.0312	0	0.0125	0.0625	0.0938	0.375	0	0.25
a3	0.0348	0.0087	0	0.1391	0.0384	0.0957	0.4174	0	0.2696
...									
a21	0	0	0	0.1562	0	0.0965	0.4618	0	0.2854

最终 hub 值最大的是节点 8（处于 base set），节点 8 指向了节点 7 和 9，同时 7 和 9 又被多个节点指向（authority 值高），因此节点 8 的 hub 值最高十分合理。
authority 值最大的是节点 7 （处于 root set），节点 7 被节点 2，3，8 指向，同时节点 2，3，8 的 hub 值高，因此节点 7 的 authority 值最高十分合理。



hub/authority 值可以反应一个网页的导航度与权威度。不同的网站目的应该侧重于不同的指标，例如导航网站应该侧重 hub 值，这样可以指向更精准；而门户网站则应该侧重

authority 值，让更多导航网站指向它，提高权威度。

++

其他（例如感想、建议等等）。

本次实验中完成了对 PageRank，hub/authority 值的计算。这两个指标都可以在一定程度上对于网页进行评估与排序。而对于不同的网站，应该侧重于不同的评价指标，具体应该根据业务场景进行设计。

在做第二道题的时候，我第一次得到答案的时候有一些疑惑，我发现虽然许多节点既有出链又有入链，但是仍然 h，a 值均为 0，比如节点 1。后来进行了深入的研究，发现对于节点 1，虽然他指向了节点 2，但是节点 2 并没有被任何其他节点所指向，所以节点 2 的 authority 值很低，从而导致节点 1 虽然有出链，但是 hub 值仍然为 0（就类似与指向了一个垃圾网站并不会提升网站导航值）。其他节点原理类似。

指导教师批阅意见：

成绩评定：

指导教师签字：

2022 年　　月　　日

备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。