

华中科技大学  
Huazhong University of Science and Technology

# 课程实验报告

课程名称： 大数据分析

专业班级： 校交 1901 班

学 号： U201910681

姓 名： 骆瑞霖

指导教师： 崔金华

报告日期： 2021-12-12

计算机科学与技术学院

# 目录

<b>1</b>	<b>PageRank 算法及其实现</b>	<b>3</b>
1.1	实验目的 . . . . .	3
1.2	实验内容 . . . . .	3
1.3	实验过程 . . . . .	3
1.3.1	编程思路 . . . . .	3
1.3.2	遇到的问题及解决方式 . . . . .	6
1.3.3	实验测试与结果分析 . . . . .	6
1.4	实验总结 . . . . .	6

# 1 PageRank 算法及其实现

## 1.1 实验目的

1. 学习 pagerank 算法并熟悉其推导过程；
2. 实现 pagerank 算法, 理解阻尼系数的作用；
3. 将 pagerank 算法运用于实际, 并对结果进行分析。

## 1.2 实验内容

提供的数据集包含邮件内容(emails.csv), 人名与 id 映射(persons.csv), 别名信息(aliasess.csv), emails 文件中只考虑 MetadataTo 和 MetadataFrom 两列, 分别表示收件人和寄件人姓名, 但这些姓名包含许多别名, 思考如何对邮件中人名进行统一并映射到唯一 id.

完成这些后, 即可由寄件人和收件人为节点构造有向图, 不考虑重复边, 编写 pagerank 算法的代码, 根据每个节点的入度计算其 pagerank 值, 迭代直到误差小于  $10^{-8}$ .

实验进阶版考虑加入 teleport , 用以对概率转移矩阵进行修正, 解决 dead ends 和 spider trap 的问题。

输出人名 id 及其对应的 pagerank 值。

## 1.3 实验过程

### 1.3.1 编程思路

本次实验最重要的莫过于对 pagerank 算法的理解, 首先 pagerank 是由 Google 提出的计算互联网网页重要度的算法, 在 pagerank 高时搜索相关内容时会出现在较前位置, 基础的 pagerank 公式可以用如下表示:

$$PR(a)_{i+1} = \sum_{i=0}^n \frac{PR(T_i)_i}{L(T_i)}$$

$PR(T_i)_i$  表示的是其他节点的 (指向 a 结点) 的 PR 值,  $L(T_i)$  表示其他节点的出链数。将求 pagerank 的过程进行矩阵化表达就是使用转移概率矩阵, 又称为马尔可夫矩阵, 通过转移矩阵可以快速计算下一轮的 pagerank, 例如对于下图: 那么会有转移矩阵

$$\begin{bmatrix} 0 & 0 & 1/2 & 1 \\ 1/2 & 0 & 0 & 0 \\ 1/2 & 1 & 0 & 0 \\ 0 & 0 & 1/2 & 0 \end{bmatrix}$$

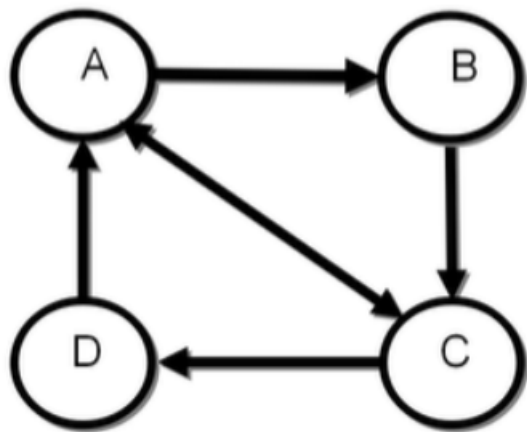


图 1: 示例图

使用均分的初始化方法，得到初始化的 PR 值序列为：

$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}$$

那么此时直接由：

$$M \cdot PR = \begin{bmatrix} 3/8 \\ 1/8 \\ 3/8 \\ 1/8 \end{bmatrix}$$

此时就得到了新的 PR；再经过多次迭代使得结果与上一次结果的距离小于误差要求时，就作为最终结果的 rankpage 值。

在了解了基本的算法思路后，我们进一步地去解决 Dead Ends 问题，Dead Ends 问题可以抽象化为这样一种叙述，如果某结点没有任何出链，它会导致网站权重在迭代的过程中变为 0；对于 Dead Ends 问题，使用 Teleport 来解决，我们对转移矩阵进行修正：

$$M + a^T \left( \frac{e}{n} \right)$$

- $a = [a_0, a_1, \dots, a_n]$ , 对于  $M$  中全为 0 的列  $i$ , 有  $a_i = 1$
- $e$  为由 1 填满的列矩阵
- $n$  为矩阵  $M$  行列数

由此修正， $M$  中全为 0 的列可以由  $\frac{1}{n}$  来填充，从而解决了没有出链的结点的问题。

然后解决 Spider Traps 问题，Spider Traps 是某结点与其他结点之间没有 out links，导致网站的权重会朝向一个结点进行偏移。

为了解决该问题，进一步对转移矩阵进行修正，Spider Traps 的结果就是某网站不会向打开其他网站这个行为进行跳转，那么就分配一个随机跳转到其他网页的概率即可，修正结果如下：

$$M = \beta M + (1 - \beta) \frac{ee^T}{n}$$

- $\beta$  表示跟随出链打开网页的概率；
- $1-\beta$  就表示随机跳到其他网页的概率；
- $ee^T$  表示大小为  $n * n$  的单位矩阵；

综上，可以得到最终的修正公式：

$$PR(a) = [\beta(M + a^T(\frac{e}{n})) + (1 - \beta)\frac{ee^T}{n}] * r$$

修正的关键过程代码如下：

```
1 //Python code
2 def update_mat(mat):
3     col_sum = np.sum(mat, axis=0)
4     n, beta = len(mat), 0.85
5     ind = [i for i in range(len(col_sum)) if col_sum[i] == 0]
6     for i in range(len(mat)):
7         for j in ind:
8             mat[i][j] = 1 / n
9     mat = np.multiply(mat, beta)
10    mat += np.multiply(np.full(mat.shape, 1 / n), 1 - beta)
11    return mat
```

在使用修正公式之前，首先将发送-接收邮件信息转化为转移矩阵，注意到 excel 中的序号并不是连续且不空缺，首先将发送接收方的所有序号向 0,1,... 进行映射转移，因为矩阵的行列  $i,j$  表示的是结点  $i$  和  $j$  之间的连接关系，构建转移矩阵后进行修正，然后对初始 rankpage 的  $n * 1$  矩阵进行迭代更新，主要代码如下：

```
1 # 计算两个n*1列向量的距离
2 def cal_distance(r1, r2):
3     return np.sqrt(np.sum(np.square(r1 - r2)))
4 new_R = np.dot(M, R)
5 epsilon = math.pow(10, -8) # 误差规定为1e-8
6 while cal_distance(new_R, R) >= epsilon:
7     R = new_R
8     new_R = np.dot(M, R)
```

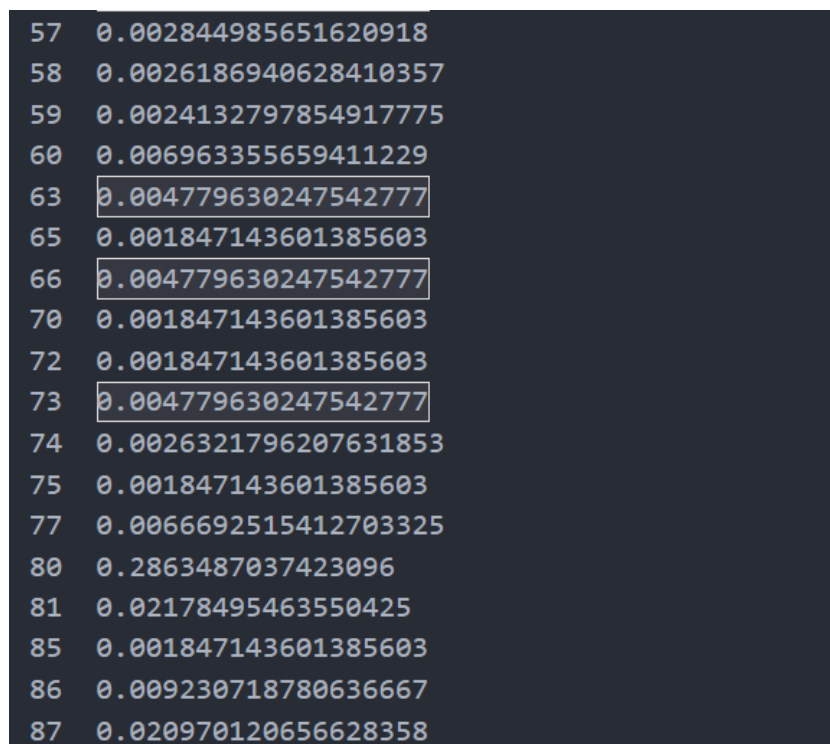
将最终结果写入 ans 文件中。

### 1.3.2 遇到的问题及解决方式

- 首先遇到的问题就是对修正公式已经遗忘并且对于原理也不够熟悉，这些通过知乎和csdn上的教程进行了查缺补漏；
- 一开始没有正确地使用转移矩阵，应该是使用初始连接矩阵进行标准化后再进行转置后的结果来进行运算；
- 对 teleport  $\beta$  的合理性和有效性有一定的怀疑，因为使用  $\beta$  进行修正之后不一定满足转移矩阵列之和为 1 的性质，所以我认为仅仅使用  $\beta$  并不能解决 Dead End 的问题，事实上这应该正是阻尼系数的缺点之一。

### 1.3.3 实验测试与结果分析

以序号-pagerank 值的形式输出结果到文件 ans 中，部分结果如下图所示：



```
57 0.002844985651620918
58 0.0026186940628410357
59 0.0024132797854917775
60 0.006963355659411229
63 0.004779630247542777
65 0.001847143601385603
66 0.004779630247542777
70 0.001847143601385603
72 0.001847143601385603
73 0.004779630247542777
74 0.0026321796207631853
75 0.001847143601385603
77 0.0066692515412703325
80 0.2863487037423096
81 0.02178495463550425
85 0.001847143601385603
86 0.009230718780636667
87 0.020970120656628358
```

图 2: 部分运行结果

## 1.4 实验总结

在本次实验中我复习了 pagerank 算法的大致流程，了解了 pagerank 算法在网页访问这样一个我们日常生活中十分频繁的行为其背后运行的机理，同时也在实验手册的指示下，在博客网站的查阅和理解下成功完成了实验。