華中科技大学

课 程 实 验 报 告

课程名称： 大数据分析

专业班级： **CS2011**

学　　号： **U202014774**

姓　　名： 王逸

指导教师： 崔金华

报告日期： **2022.12.28**

计算机科学与技术学院

目录

实验二 PageRank 算法及其实现

2.1 实验目的

1、学习 pagerank 算法并熟悉其推导过程；
2、实现 pagerank 算法[1]；（可选进阶版）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。

2.2 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（aliases.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？（提供预处理代码 preprocess.py 以供参考）。

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 β，用以对概率转移矩阵进行修正，解决 dead ends 和 spider trap 的问题。

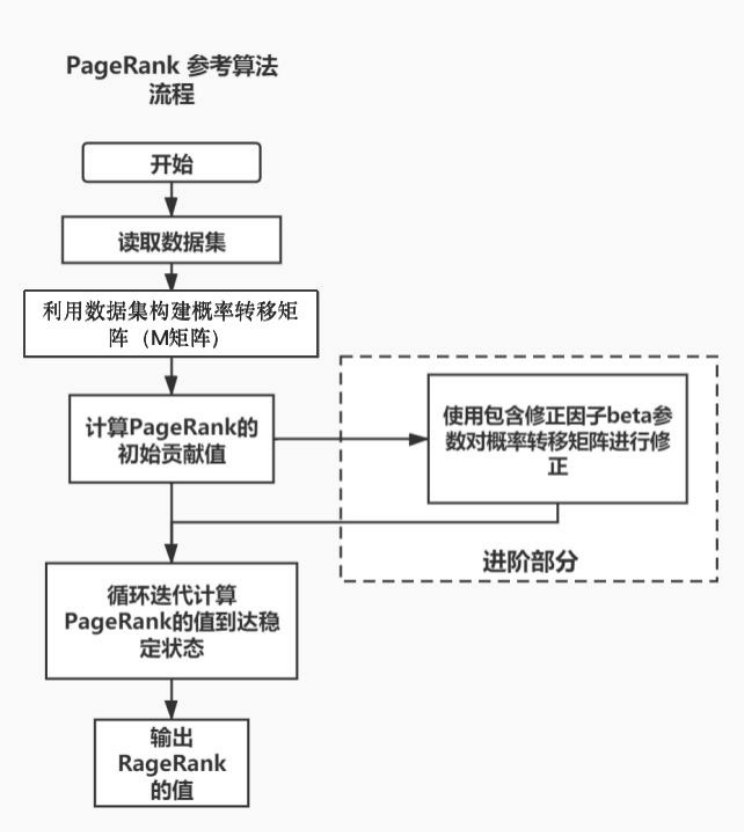
输出人名 id 及其对应的 pagerank 值。

2.3 实验过程

2.3.1 编程思路

本次实验的总体流程如下：

[1] 基本 pagerank 公式 r=Mr



该实验的 py 文件中定义了两个函数：

1) read_data(data_file)：读取文件数据函数，返回边列表 edges 和结点列表 nodes。对于文件中的每一行先后调用 strip 与 split 函数去掉换行符与分隔符，存放到 edges 中，由于读入的文件第一行是列表信息，需要去掉该行。对于 edges 中的每一条边信息，若起点或终点不在 nodes 中则加入。
2) Init_matrix(edges, nodes)：利用边和结点信息初始化概率转移矩阵 m。首先初始化 m 矩阵为 n x n 的全零矩阵（n 为结点总数），然后遍历边列表，假设 i 为边起点，j 为边终点，m[j, i] = 1.最后计算某结点的出度（列和）并将该列除以出度。初始化贡献值矩阵 r，每个值均为 1/n。返回 m，r。
3) 主函数：首先读取 sent_receive.csv 数据，并利用返回的 edges 和 nodes 信息构建概率转移矩阵 m 及初始贡献值矩阵 r。将初始误差值设置为一个较大值，beta 设置为 0.85，进入循环，套用进阶版的 pagerank 公式循环迭代计算，每次计算完之后对 r 矩阵进行归一化处理，误差 e 为处理前 r 与处理后 r 矩阵差值之和，当 $e<10^{-8}$ 时退出循环。最后输出结点 id 及对应的 pagerank 值。

2.3.2 遇到的问题及解决方式

本次实验遇到的问题主要是当时读取数据时并没有考虑到首行列表信息，导致第一次运行时计算输出全为 0.后面在 read_data 中读取完行信息保存到 edges

后去掉 edges[0]，后面即可正确输出。

2.3.3 实验测试与结果分析

输出结果的部分截图如下所示：

```
id: 87, rank score: 0.02714157588752684
id: 80, rank score: 0.31773018004210113
id: 32, rank score: 0.02326263116307286
id: 81, rank score: 0.028107758275780357
id: 185, rank score: 0.00489375730839298
id: 77, rank score: 0.007406321120020525
id: 213, rank score: 0.007195674179671527
id: 194, rank score: 0.00489375730839298
id: 21, rank score: 0.00489375730839298
id: 22, rank score: 0.0069020754994325575
id: 160, rank score: 0.0069020754994325575
id: 216, rank score: 0.00489375730839298
id: 150, rank score: 0.006085687415216483
id: 48, rank score: 0.008744082387677139
id: 170, rank score: 0.010029613552542544
id: 201, rank score: 0.006351973700295756
id: 116, rank score: 0.0092869355298018
id: 143, rank score: 0.00489375730839298
id: 38, rank score: 0.0099977819590863512
```

2.4 实验总结

本次实验主要完成 pagerank 算法的实现，由于之前因为个人兴趣就已经了解并实现过该算法，所以在算法编写部分没有太大问题，数据读取部分因为最开始忽略了第一行信息导致后面输出有问题，改正后也能正常输出。

[i] 进阶版 pagerank 公式：$r = \beta Mr + (1-\beta)[\frac{1}{N}]_{N\times N}$，其中$\beta$为阻尼系数，常见值为 0.85