

**课 程 实 验 报 告**

**课程名称： 大数据分析**

**专业班级：**

**学 号：**

**姓 名：**

**指导教师：**

**报告日期：**

**计算机科学与技术学院**

**目录**

[实验二 PageRank算法及其实现 1](#_Toc57042479)

[**1.1实验目的** 1](#_Toc57042480)

[**1.2 实验内容** 1](#_Toc57042481)

[**1.3 实验过程** 1](#_Toc57042482)

[1.3.1 编程思路 1](#_Toc57042483)

[1.3.2 遇到的问题及解决方式 1](#_Toc57042484)

[1.3.3 实验测试与结果分析 1](#_Toc57042485)

[**1.4 实验总结** 2](#_Toc57042486)

# 实验二 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（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。

## **1.3 实验过程**

### 1.3.1 编程思路

由于之前的相关经历，对networkx库较为熟悉，所以选择采用networkx库来完成实验。

1. 建图

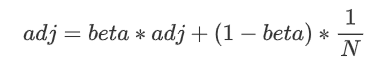
首先利用networkx建一个有向图，并从数据集读入数据。

2. 重命名边

由于原本的数据集person id并不是连续，所以需要简历一个映射关系，使得其映射成连续的id。

3. 获得转移矩阵

首先根据networkx自带的方法获得邻接矩阵，再根据beta值构造转移矩阵，公式为：



4. 利用公式进行迭代



利用该公式即可进行迭代，并利用两次迭代的误差，判断收敛，获得结果

### 1.3.2 遇到的问题及解决方式

1. 矩阵的转置问题

由包获得的邻接矩阵为Mij，是行对应的，需要进行转置获得列对应。

2. 转移矩阵的构造。

由于对公式的错误理解，以为每次迭代都需要进行beta的修正，但其实只需要对转移矩阵修饰一次即可。

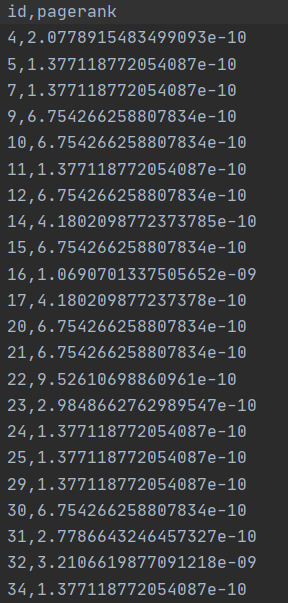
3. 误差的衡量

最开始以为是对误差进行L2范数的统计，结果是对无穷范数进行统计，即求每一维误差的最大值作为最终误差。

4. 归一化问题

自己写的初始版本每次迭代都会对rank值做一个归一化处理，为了方便助教检查，需要去掉归一化操作。

### 1.3.3 实验测试与结果分析



可以看到rank值是非常小的，一方面是因为没有做归一化操作，另一方面是因为边的权重默认为1，所以1在进行转移，自然会比较小。

## **1.4 实验总结**

这次实验比较简单，主要是对pagerank算法的理解，包括转移矩阵的构建的迭代的方法，我个人觉得最重要的有以下几点：

1. 理解了矩阵乘法是如何实现信息（rank值）的传播，还好线代没有忘记，通过这次实验，让我明白了“转移”一词的真正含义。

2. 通过实验进一步加深了对利用teleport解决dead end 和 spider trap问题的认识。