

**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](#_Toc134304083)

[**2.1实验目的** 1](#_Toc134304084)

[**2.2 实验内容** 1](#_Toc134304085)

[**2.3 实验过程** 1](#_Toc134304086)

[2.3.1 编程思路 1](#_Toc134304087)

[2.3.2 遇到的问题及解决方式 2](#_Toc134304088)

[2.3.3 实验测试与结果分析 3](#_Toc134304089)

[**2.4 实验总结** 3](#_Toc134304090)

# 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，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以及处理后的sent\_receive.csv数据以供参考）。

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

根据相关的定义，相关的迭代式如图所示：

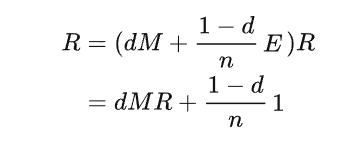


图2.1迭代公式图

其中d为阻尼因子，也即实验中的alpha，其值为0.85，M为原有向图构建的邻接矩阵经过归一化处理后得到的矩阵。对于本实验，我们无需担心迭代次数问题，我们只需要关注迭代前和迭代后两次的R矩阵的范数之差是否在我们所给范围内，如果是则可以输出对应的id以及迭代算出的pagerank值。

首先我们需要根据sent\_receive.csv文件，构建对应的有向图，python中提供了networkx库方便处理复杂的图网络结构。我们只需要将csv文件中第二列以及第三列id，当作有向边的两个点添加到提前定义好的有向图graph之中即可。

相关程序部分如下所示：

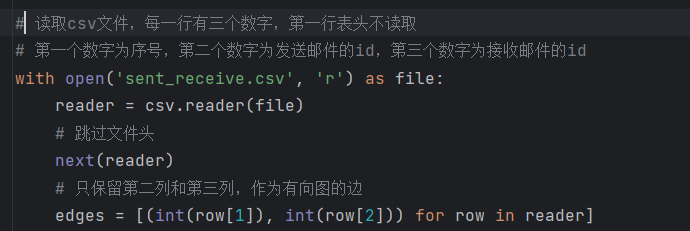


图2.2读取文件转化为有向图

然后我们需要根据这个有向图，构建对应的邻接矩阵，然后进行归一化处理，R的初始矩阵处理也和下面相似，相关代码如下所示：

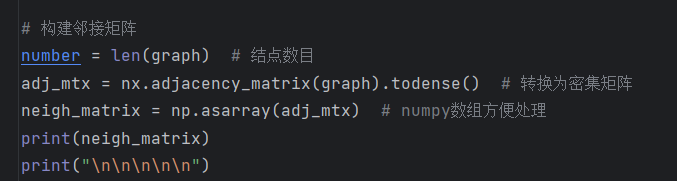


图2.3有向图转换为邻接矩阵

最后我们只需要进行迭代处理即可，迭代部分如下所示：

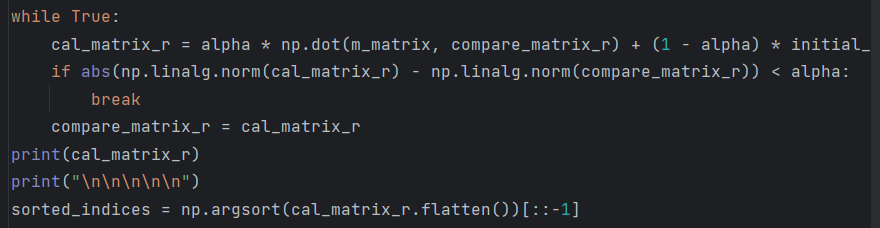


图2.4进行迭代处理

### 2.3.2 遇到的问题及解决方式

问题一，如何将得到的有向图转换为邻接矩阵。

在networkx库中给我们提供了networkx.adjacency\_matrix方法，可以十分快捷的将有向图直接转换为邻接矩阵。由于邻接矩阵大部分时候都常为稀疏矩阵，而稀疏矩阵会导致后续计算中的一些过程会降低处理速度，所以使用todense()将稀疏矩阵转换为密集矩阵。

问题二，如何衡量矩阵大小，如何将其做差比较与我们所给误差的大小。

在numpy库中给我们提供了numpy.linalg.norm方法，帮助我们直接计算矩阵的范数。关于范数，是矩阵元素的某个函数，可以帮助我们衡量矩阵的大小。

### 2.3.3 实验测试与结果分析

在执行PageRank.py程序后，我们按照PageRank值的大小从大到小将结果写入result.csv文件。结果符合预期值，结点入度越多，出度适当较小PageRank值越高，代表其结点更重要。

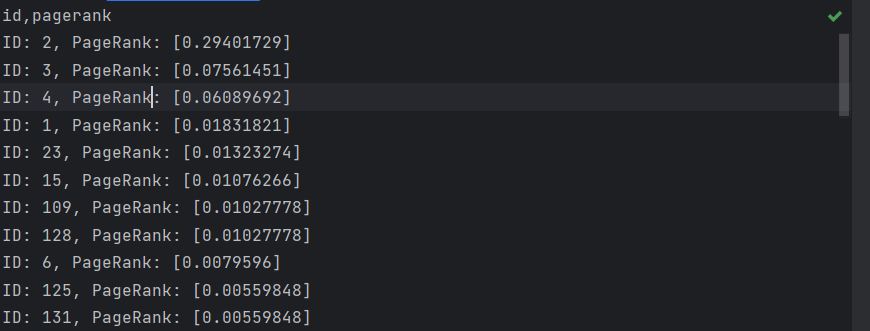


图2.5部分输出结果

## **2.4 实验总结**

关于实验部分的提出的问题思考：

preprocess.py文件实现了以下的步骤：一是读取别名信息，构建别名到PersonId的映射关系。二是读取人名信息，构建PersonId到Name的映射关系。三是对每个邮件中的MetadataTo和MetadataFrom进行格式化处理，将所有人名转换为小写并去除逗号，并根据别名信息将别名转换为对应的PersonId。完成了映射。

在查询PageRank算法中，我在网上了解了一下信息：

Pagerank算法在搜索引擎领域中被广泛应用，是Google搜索引擎排名算法的核心之一。它通过对网页之间的链接关系进行分析和评估，对每个网页赋予一个权重，从而使得搜索引擎能够更好地为用户提供有用的搜索结果。Pagerank算法基于网络中节点之间的链接关系，通过随机游走模型计算每个节点的重要性，其计算过程比较简单，但结果准确性较高，尤其在大型复杂网络中的应用效果显著。