

**课 程 实 验 报 告**

**课程名称： 大数据分析**

**专业班级： 计科2003班**

**学 号： U202015374**

**姓 名： 张隽翊**

**指导教师： 崔金华**

**报告日期： 2022年12月28日**

**计算机科学与技术学院**

**目录**

[实验二 PageRank算法及其实现 1](#_Toc58252332)

[**2.1实验目的** 1](#_Toc58252333)

[**2.2 实验内容** 1](#_Toc58252334)

[**2.3 实验过程** 1](#_Toc58252335)

[2.3.1 编程思路 1](#_Toc58252336)

[2.3.2 遇到的问题及解决方式 2](#_Toc58252337)

[2.3.3 实验测试与结果分析 3](#_Toc58252338)

[**2.4 实验总结** 3](#_Toc58252339)

# 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]](#footnote-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

（1）构建邻接矩阵M

预处理代码输出的sent\_receive.csv文件给出了每一封邮件的发件人和收件人，从中可以得到通信对<sent\_id, receive\_id>，每个通信对代表在网络图中存在一条由用户sent\_id指向receive\_id的有向边。

初始创建一个N×N的全零矩阵M，当存在用户i到用户j的有向边时，令M[i, j] = 1。然后归一化矩阵M，先对M的每一列进行求和，再让这一列的元素除以这个和，使得矩阵的每一列元素之和为1（全0列除外）。

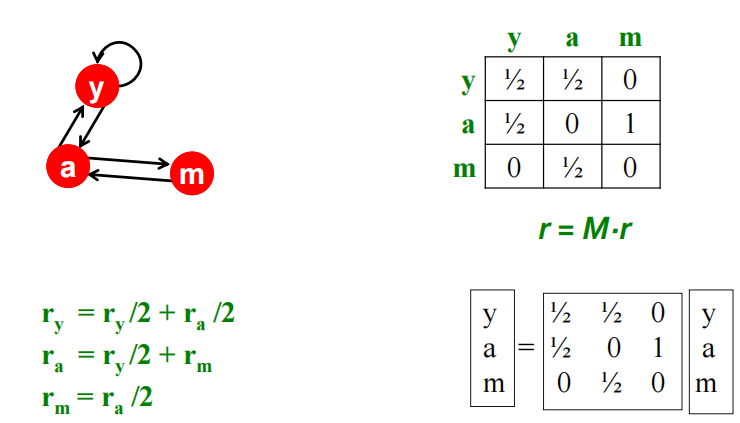


图 1.1 邻接矩阵构造过程

（2）迭代计算结果

对于基础版，将矩阵M带入基本公式计算r\_new；对于进阶版，利用矩阵M构建矩阵A，再带入计算。迭代至相邻两次的秩向量之差小于10-8时停止。每次迭代更新后将r\_new归一化，计算误差后用r\_new更新r再进入下一步迭代。

### 2.3.2 遇到的问题及解决方式

（1）构造归一化前的邻接矩阵M较慢

直接由<sent\_id, receive\_id>通信对构造邻接矩阵M比较缓慢，可以先利用Pandas读取，然后利用其groupby方法快速构建。

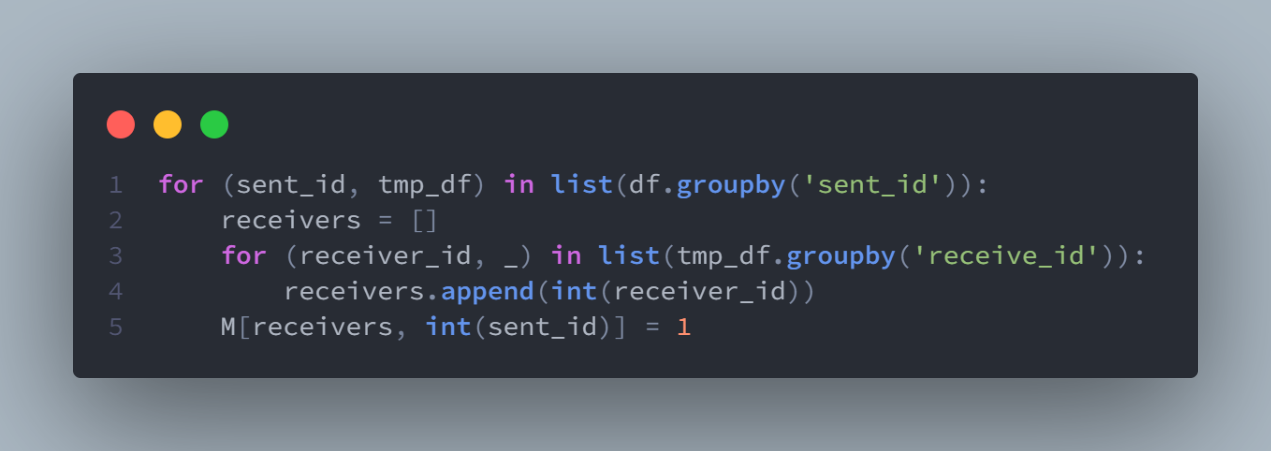


图 1.2 构造邻接矩阵的代码

（2）对更新后的r\_new也要进行归一化操作

这是由计算误差的方式决定的。误差在r\_new与r之差的基础上计算得到，需要r\_new与r的计算规则一致，而r初始时进行了归一化，因此r\_new也要进行归一化操作。

### 2.3.3 实验测试与结果分析

（1）基础pagerank测试

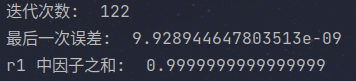


图 1.3 基础pagerank测试结果

由测试结果可知，最终得到的r向量因子之和在误差允许的范围内等于1，迭代次数较多，退出时的误差小于10-8，满足要求。

（2）进阶pagerank测试

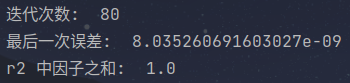


图 1.4 进阶pagerank测试结果

由测试结果可知，最终得到的r向量因子之和在误差允许的范围内等于1，迭代次数较少，退出时的误差小于10-8，满足要求。

## **2.4 实验总结**

本次实验实现了pagerank算法的基础版本的进阶版本。

在基础版本的pagerank算法中，仅使用归一化的邻接矩阵M更新秩向量r。在这种情况下，当M非满秩时会导致r在若干次迭代后变成0。导致M非满秩的可能情况有Spider traps（蜘蛛陷阱问题）和Dead ends（死角问题）。在Spider traps情况下，pagerank算法的随机选择局限在某些结点组成的集团内部，因为集团内部的这些结点只有指向集团内部结点的边，而没有指向集团外部结点的边，在若干次迭代后集团内部的结点会聚集大多数权重；在Dead ends情况下，pagerank算法的随机选择无路可走，直接导致r变为0。为解决这一问题，Google提出了改进的pagerank算法。

在进阶版本的pagerank算法中，引入了随机跳转系数beta，每次随机选择有1-beta的概率通过teleport进行rank传递。这一机制的引入实质上是在邻接矩阵M中引入了一个元素值较小的常量矩阵，对于存在Spider traps和Dead ends情况的结点进行修正，从而使得修正后的矩阵有极大概率满秩。

在实践中，要注意及时对秩向量r进行归一化，否则会得出错误的结果。算法迭代过程中误差的变化趋势如图1.5所示。

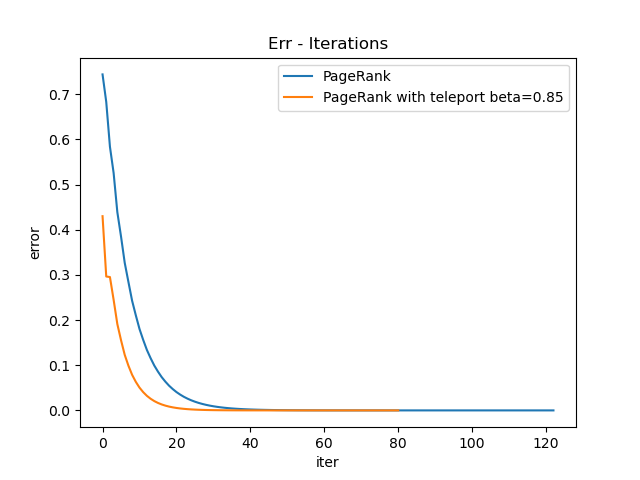


图 1.5 误差随迭代次数变化趋势图

1. 基本pagerank公式r=Mr

   2 进阶版pagerank公式：r，其中为阻尼系数，常见值为0.85 [↑](#footnote-ref-1)