

**课 程 实 验 报 告**

**课程名称： 大数据分析**

**专业班级： cs1902**

**学 号： U201914945**

**姓 名： 顾璨**

**指导教师： 杨驰**

**报告日期： 2021/12/15**

**计算机科学与技术学院**

**目录**

[实验二 PageRank算法及其实现 1](#_Toc90027732)

**[2.1实验目的](#_Toc90027733)** [1](#_Toc90027733)

**[2.2 实验内容](#_Toc90027734)** [1](#_Toc90027734)

**[2.3 实验过程](#_Toc90027735)** [1](#_Toc90027735)

[2.3.1 编程思路 1](#_Toc90027736)

[2.3.2 遇到的问题及解决方式 1](#_Toc90027737)

[2.3.3 实验测试与结果分析 1](#_Toc90027738)

**[2.4 实验总结](#_Toc90027739)** [1](#_Toc90027739)

# 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的名称，但这些名称包含许多别名，因此需要对邮件中的名称进行统一并映射到唯一id。（提供预处理代码preprocess.py参考）。

完成这些后，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。

输出人名id及其对应的pagerank值。

加分项：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。

## **2.3 实验过程**

1. 将给定数据进行预处理, 得到id-id 表格

2. 将之转化为接邻矩阵

3. 将每个人的分数平分给他信件的接收者

4. 构造初始化向量其中每人的权重相同

5. 用向量反复点乘矩阵, 直到向量变化小于阈值

### 2.3.1 编程思路

1. 调用老师给的预处理文件

2. 初始化矩阵

mat = np.zeros((513, 513))

for (sent\_id, part\_df) in *list*(df.groupby('sent\_id')):

    receivers = []

    for (receiver\_id, ppt\_df) in *list*(part\_df.groupby('receive\_id')):

        receivers.append(*int*(receiver\_id))

    count = **len**(receivers)

    mat[receivers, *int*(sent\_id)] = 1 / count

1. 初始化向量

r = np.ones((513, 1)) / 513

1. 迭代

while 1:

    new\_r = np.dot(mat, r)

    diff = (new\_r - r)\*\*2

    if np.sum(diff) < 1e-8:

        break

    r = new\_r

### 2.3.2 遇到的问题及解决方式

对线性代数的掌握不够熟练,一开始未区分左乘和右乘导致出错, 后续进行了订正

### 2.3.3 实验测试与结果分析

## **2.4 实验总结**

通过本次实验我了解并实现了 PageRank 算法, 复习了基本线性代数知识, 对于其中的 teleport 部分我并没有完全理解, 后续我将继续学习尝试掌握并实现这部分内容