数据科学导论 PageRank 算法实现和理论作业

中国人民大学信息学院

1 基本的 PageRank 函数实现

核心算法实现，不允许直接调包，比如 nx.pagerank(xxx)。

给定描述一个 graph 的邻接表文件，每一行是 ‘,’（键盘上的普通英文逗号）分隔的节点 ID。节点 ID 是不含逗号的字符串。每一行的含义是，行首节点有指向后边节点的一条有向边。例如，某个只有 $\{A, B, C\}$ 三个节点的图 G_1，边集为 $\{A \to B, B \to C\}$，则我们给定的输入文件如下：

```
A,B
B,C
```

边权都是 1。

输出按 PageRank 分数降序排序的节点 list。list 中的元素都是 str，**分数相同的两个节点，按字典序降序排列**。

【小测试集】我们提供了一个小测试集，g2，包含四个点，六条边，返回的正确答案 list 应该是['a','b','c','d']，他们的具体分数是[0.36, 0.34, 0.18, 0.12]（不需要返回具体分数，只返回节点列表，其内容元素是 str 型）。

代码中必须包含一个命名为 PageRank 的函数，名字不要搞错！所有代码必须写在这个函数内。代码函数模板如下：

```python
#<学号>-PageRank.py
#@param input_file_name: 描述一个graph的纯文本邻接表文件名，如'E:\graph.txt'。
#@param damping_factor
def PageRank(input_file_name, damping_factor):
    #your implementation
    return node_list_in_descending_order
```

上面的简单测试通过后，请自己下载 Epinions 数据集 `http://snap.stanford.edu/data/soc-Epinions1.html` ，首先预处理成上述输入文件格式。然后，运行自己的 PageRank 代码，并 damping factor 设为 0.85。运行过程和结果截图，程序运行时间，top-10 节点及其分数，写进实验报告。

注意：这里一个核心的问题是如何存储稀疏的图，请先自己尝试思考解决策略，上机课可以提出问题，进行研讨。

2 Personalized PageRank 函数实现

共有两个输入文件：图、种子。图输入文件同 1 小节。种子输入文件每一行是：一个种子节点, 种子分数。例如种子集为 $\{A:0.5, B:0.5\}$，种子集描述文件如下：

```
A,0.5
B,0.5
```

输出降序排序的节点 list。list 中的元素都是 str。

代码中必须包含一个命名为 PPR 的函数，名字不要搞错！所有 PPR 相关代码必须写在这个函数内。代码函数模板如下：

```
#@param input_Graph: 描述一个graph的纯文本邻接表文件名，如'E:\graph.txt'。
#@param input_Seed: 描述一个种子集的纯文本文件名，如'E:\seed.txt'。
#@param damping_factor
def PPR(input_Graph, input_Seed, damping_factor):
    #your implementation
    return node_list_in_descending_order
```

同样使用 Epinions 数据集 `http://snap.stanford.edu/data/soc-Epinions1.html` 运行自己的 Personalized PageRank 代码，damping factor 设为 0.85，种子集选取编号自 0 到 49 的 50 个节点，每个种子的初始分数为 $\frac{1}{50}$。运行过程和结果截图，程序运行时间，top-10 节点及其分数，写进实验报告。

3 Personalized PageRank 线性可加证明题

现有全连通的平凡图 $G =< V, E >, |V| = n$ 。

n 个 nodes（其中包含种子）的分数向量 $p^{(0)} = [\lambda_1, \lambda_2 \ldots \lambda_n]$，满足 $\sum_{i=1}^{n} \lambda_i = 1$，和 Personalized PageRank 函数 $PPR(p^{(0)}, G, \alpha)$。函数 PPR 输出是所有节点的最终收敛分数向量 p^*，即 $p^* \leftarrow PPR(p^{(0)}, G, \alpha)$。

令 $p_i^{(0)} = [0, 0, \ldots, 1, \ldots, 0]$ （长度和 $p^{(0)}$ 一样，但是第 i 个元素为 1，其余全为 0），formally,

$$p_i^{(0)}[j] = \begin{cases} 1 & if \quad j == i \\ 0 & if \quad j! = i \end{cases} \quad 1 \leq j \leq n$$

$p_i^* \leftarrow PPR(p_i^{(0)}, G, \alpha)$。
求证：$p^* = \sum_{i=1}^{s} \lambda_i p_i^*$
证明写进实验报告。

4 提交要求

本次所有作业打包成一个 zip ！！ 不是 rar！！，命名为 < 学号 >-< 姓名 >-PageRank.zip （去掉尖括号）。实验报告格式仅接受 pdf，命名为 < 学号 >-PageRank 实验报告.pdf（去掉尖括号）。代码命名为 < 学号 >-PageRank.py。命名不要加空格。学号请写全。统一命名是为了方便自动批改，生成成绩报表，请同学们理解，谢谢。

你提交的一份代码只能有 PageRank 和 PPR 两个函数以及一些你用到的 import。不要写 main 函数，不要带自己随手测试的代码单独交的代码只是供我们自动评测用的 < 学号 >-PageRank.py。< 学号 >-PageRank.py 里可以有其他自定义函数，可以用常见的包。但是不要写 main 函数，也不要把自己测试的代码带进去。助教导入你们的函数的时候那些你们自己测试的代码会被执行可能会报错。

总共需要提交 2 个文件：即 < 学号 >-PageRank.py、实验报告，总大小不超过 3M。上传到obe.ruc.edu.cn 。实验报告最后要有实验小结。

实验报告必需的章节：包括两个函数的代码实现思路，在 Epinions 数据集上的实验结果（详见1和 2），如何应对大规模图数据的设想和实现，Personalized PageRank 线性可加证明题，实验小结。

coding 的要求注意命名规范易读、各模块带有注释。

写明 OS，Python 语言版本。

注意：Epinions 数据预处理代码不需要单独交，可以写在实验报告里。Epinions 数据不需要在你提交的代码里体现，可以在实验报告里体现。