PageRank研究报告

付星宇 15338046

中山大学数学学院

目录

1）PageRank 概述

1.1 传统搜索引擎与其缺陷

在 Google 之前，市面上已经出现了很多的搜索引擎，他们利用网络爬虫在互联网上爬取关键词(Term)，并构造倒序索引(Inverted Index)这种数据结构，建立了关键词与网页之间的映射(Mapping)关系。 当用户搜索某个关键词时，搜索引擎从倒序索引中提取出这个关键词所对应的所有网页，并通过关键词在各个网页中出现的频率，出现的位置，给网页进行排序，将排序靠前的网页推荐给用户。

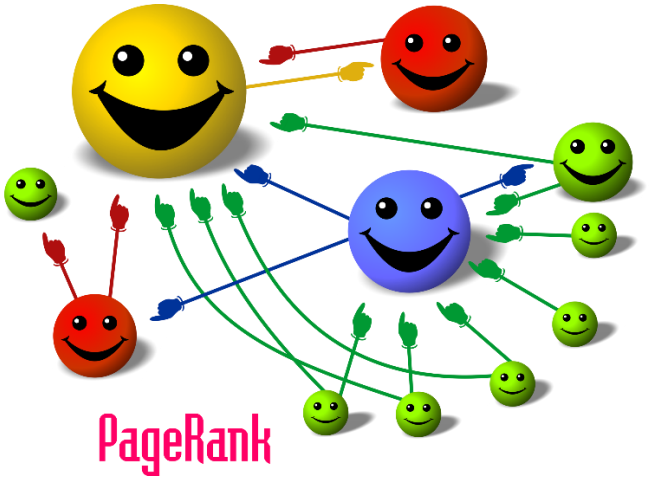
垃圾网站的出现几乎使早期搜索引擎完全丧失工作能力，这是因为早期搜索引擎的推荐策略非常简单，只考虑关键词与网页中文本的匹配次数与匹配位置，例如：在网页开头匹配到关键词，则搜索引擎认为大概率相关；关键词的匹配次数很高，则搜索引擎认为大概率相关。

这种单一的推荐策略使得构造垃圾网站十分简单，垃圾网站构造者只需将热门的搜索内容不停地复制到自己的网页上，然后将这些“幌子”全部设为背景色即可。

大量垃圾网站的出现，使得早期搜索引擎搜索得到结果几乎没什么意义，正是在这种大环境下，Google 采用了开创性 PageRank 算法，从众多搜索引擎当中脱颖而出。

1.2 PageRank：一种革命性的网页打分手段

尽管 Google 不是世界上第一家搜索引擎公司，但它是第一个成功战胜垃圾网站的搜索引擎。给定关键词，Google 不仅仅考虑网页中关键词出现的词频与词出现的位置，它还在网页的打分中加入了一个重要的成分 PageRank。



Google 的创始人拉里·佩奇和谢尔盖·布林于 1998 年在斯坦福大学发明了这项技术，PageRank 打分基于互联网复杂的拓扑结构，让互联网中的网页互相"投票"，最终得到一个稳定的网页重要性指标。

PageRank 有一下几个优点：

- 垃圾网页很难改变互联网的拓扑结构，故 PageRank 打分使传统垃圾网页被消灭。
- 各个网页的 PageRank 打分是由整个互联网决定的，稳定且客观。
- 从数学上可以严格证明 PageRank 算法的收敛性与稳定性。

可以说，正是因为 PageRank 的这些优点，使得谷歌成为当今世界上市场份额占额最大的搜索引擎：

Search engine	Market share in March 2017	
Baidu	5.94%	
Bing	6.92%	
Google	80.52%	
Yahoo!	5.35%	

2）PageRank 的工作原理与实验模*拟*

2.1 随机游走

PageRank 的数学基础是著名的随机游走(Random Walk)模型，RW 不仅在网页链接分析中起到革命性作用，也广泛应用在金融，物理学，生物学，社会学的建模工作中，其中比较著名的结果是金融理论的随机游走假说，它是有效市场假说的一个充分条件。本节我们将介绍 PageRank 是如何利用随机游走模型对网页进行打分的。

所谓随机游走，直观来说，我们想象一位旅行家站在有向图(Directed Graph)G的某个结点v上，然后他随机地选择一条从v出发的有向边e，并走到e的终点v'。之后，旅行家在v'点递归地重复上述过程。

互联网是一个巨大的有向图，这也就给了随机游走发挥作用的平台。搜索引擎利用网络爬虫的技术，将待打分的网页以及网页之间的链接关系爬取下来。将网页抽象为结点，将链接关系抽象为边，那么搜索引擎爬取下的复杂网络结构便可以抽象为一个有向图:

$$G = \langle V, E \rangle$$

其中V为结点集，E为有向边集，设$|V| = n$。

PageRank 将随机游走模型作用在图G上，采用迭代的计算方法，逐步找到每一时刻 t，各个网页被旅行家游走的概率。当$t \to \infty$时，游走概率向量收敛到一个向量$p \in R^n$，其中p的第i个分量的含义为第i个网页被旅行家游走的概率。网页被游走概率越大，网页的重要性也就越大。

假设在 0 时刻，旅行家均匀分布在这n个网页结点上，故有初始游走概率向量：

$$p_0 = \frac{\vec{e}}{n} \in R^n$$

其中$\vec{e} = (1\ 1 \ldots 1)^T$。

考虑 t 时刻旅行家到达第i号网页的概率，由全概率公式：

$$\mathbf{p}(\ \mathbf{t时刻旅行家到达i号网页}\ )$$

$$=\sum_{\mathbf{j=1}}^{\mathbf{n}}\mathbf{p}(\ \mathbf{从j号网页移动到i号网页|在j号网页})\times\mathbf{p}(\ \mathbf{（t-1）时刻在j号网页})$$

考虑所有的$i\in\{1,2,\dots,n\}$，可以将上式表达为矩阵的形式：

$$p^{(t)}\coloneqq Mp^{(t-1)}$$

其中$p^{(t)}$与$p^{(t-1)}$分别代表 t 时刻与(t-1)时刻旅行家的游走概率向量，M是一个$n\times n$的矩阵，M的 j 列 i 行的元素为旅行家从 j 号网页移动到 i 号网页的概率，我们称M为网络G的马尔科夫转移矩阵(Markov Transition Matrix)，如果网络G不存在出度为零的死结点(Dead End)，那么M每列的和均为 1，满足这种性质的矩阵称为随机矩阵(Stochastic Matrix)。

传统的 PageRank 算法利用$p^{(t)}\coloneqq Mp^{(t-1)}$不断更新游走概率向量，直到前后两次更新不再发生太大变化，即：

$$\left|p^{(t)}-p^{(t-1)}\right|_2<\varepsilon$$

其中$\varepsilon>0$为系统事先给定的一个收敛标准。

从更新方程可以看出，基于随机游走的 PageRank 算法是离散的马尔科夫过程(Markov Process)，因为它满足无记忆性(Memorylessness)，即 t 时刻的状态只与(t-1)时刻状态有关。

2.2 马尔科夫转移矩阵的谱半径与 PageRank 的收敛性

从 2.1 的讨论可以看到，我们实际上是用迭代的方法，求解向量方程：

$$p = Mp$$

其中M是$n \times n$的马尔科夫转移矩阵，p是$n \times 1$的游走概率向量，若上述方程有解，则p是M的特征值为 1 的特征向量。

那M存不存在值为 1 的特征值呢？如果不存在，那么我们在 2.1 中的迭代方法一定不可能收敛，整个 PageRank 算法也就失去了作用。下面的定理回答了这个问题：

定理 1（随机方阵谱半径定理）

若M是 n×n 随机方阵（非负；所有列的列和均为 1），则：

（1）M存在值为 1 的特征值

（2）M的谱半径为 1

证明：

（1） 令A为M的转置矩阵，由M的定义，我们得到：

$$\sum_{j=1}^{n} a_{ij} = 1 \ , \ i \in \{1, \ 2, \dots, n\}$$

考虑向量$\mathrm{e} = (1\ 1\ 1 \dots 1)^{\mathrm{T}} \in R^n$，显然有$Ae = (1\ 1\ 1 \dots 1)^{\mathrm{T}} = e$，故 1 是A的一个特征值，而注意到 A 与 M 互为转置关系，故 1 也是 M 的特征值，证毕。

（2） 取 M 的列范数，$\|M\|_1 = \max\limits_{1 \le j \le n} \sum_{i=1}^{n} \left|a_{ij}\right|$，而 M 是随机矩阵，故 M 的列和均为 1，即$\|M\|_1 = 1$。由线性代数的知识可知，M 的谱半径小于等于 M 的列范数，故 M 的谱半径小于等于 1。而由（1）知，M 有值为 1 的特征值，故 M 的谱半径等于 1， 证毕。

随机方阵谱半径定理解决了理想情况下 PageRank 游走概率向量解的存在性问题，只要网络中没有出度为零的死结点，那么该网络的马尔科夫转移矩阵M一定是随机矩阵，故$p = Mp$有解。

Remark 如果网络中出现了死结点，则网络的马尔科夫转移矩阵会出现全零列，不能保证值为 1 的特征值的存在性，考虑如下网络：

其马尔科夫转移矩阵为：

$$\begin{pmatrix} 0.5 & 0 \\ 0.5 & 0 \end{pmatrix}$$

其特征值为 0 与 0.5，PageRank 算法不可能收敛。

目前，我们只是解决了：在网络中没有出度为零的结点的情况下，游走概率向量有解。我们没有证明，当取$p_0 = \left(\frac{1}{\mathrm{n}}\ \frac{1}{\mathrm{n}}\ \ldots \frac{1}{\mathrm{n}}\right)^{\mathrm{T}}$时，利用迭代公式$p^{(t)} \coloneqq Mp^{(t-1)}$，即$p^{(t)} \coloneqq M^t p_0$更新p，当 t 趋于无穷时，p 会收敛到$M$的特征向量，下面的定理解答了这个问题：

<u>定理 2（PageRank 幂次运算收敛性）</u>

设网络G没有死结点，M为G的马尔科夫转移矩阵，p_0是任意给定的 n 维实数向量，则 $p^{(t)} \coloneqq M^t p_0$ 迭代公式收敛到M的以 1 为特征值的特征向量。

证明：

由定理 1 我们知道，M的谱半径为 1，故设 1 = λ1 > |λ2| > …>|λm|，为M的 m 个不同的特征值。因为特征子空间直和分解R^n，故有对p_0的如下分解:

$$p_0 = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n$$

其中x_i为M的 n 个线性无关的特征向量，设特征值 1 的几何维数为 k，故$\{x_1, x_2, \ldots, x_n\}$中 k 个从属于特征值 1 的特征向量，不妨设为$\{x_1, x_2, \ldots, x_k\}$。

将M^t左乘到p_0的分解式中，因为$\{x_1, x_2, \dots, x_n\}$为M的特征向量，且$\{x_1, x_2, \dots, x_k\}$从属于特征值 1，则有:

$$M^t p_0 = (a_1 x_1 + a_2 x_2 + \cdots + a_k x_k) + (\lambda_j^t a_{k+1} x_{k+1} + \cdots + \lambda_i^t a_n x_n)$$

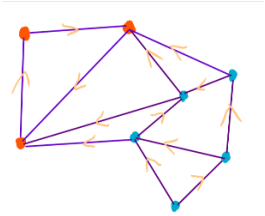
上述等式右边第二个括号中的每一项都乘以了某个绝对值小于 1 的特征值的 t 次方，当$t \to \infty$时，等式收敛到$(a_1 x_1 + a_2 x_2 + \cdots + a_k x_k)$。注意到$\{x_1, x_2, \dots, x_k\}$均为特征值 1 的特征向量，而同一个特征值的特征向量的全体构成一个线性空间，故$(a_1 x_1 + a_2 x_2 + \cdots + a_k x_k)$也是$M$的特征值为 1 的特征向量，证毕。

定理 1 与定理 2 完全证明了：在无死结点的情形下，PageRank 算法的解的存在性与迭代更新的收敛性，我们从理论的层面上，严格验证了 PageRank 的有效性。

2.3 远移：对抗不良网络结构

我们在 2.2 中看到，PageRank 的解的存在性与算法的收敛性均依赖于一个条件，那就是爬取下来的网络关系中，不存在死结点。若是存在死结点，那该网络的马尔科夫转移矩阵就会存在全零列，从而不能保证值为 1 的特征值的存在性。

除了死结点之外，还有一种不良的网络结构：蜘蛛陷阱(Spider Traps)。这种网络结构是一个网页结点的集合，集合中每一个点均不是死结点，但是这个集合没有指向外部的边，下图中三个橙色的点便构成了一个蜘蛛陷阱。



蜘蛛陷阱会使旅行者在随机游走的过程中被困在里面，永远逃逸不出去，使得 PageRank 的打分几乎全部都分散在蜘蛛陷阱之中，而蜘蛛陷阱之外的网络结构，无论结构多么复杂，功能多么重要，都只会分得小部分 PageRank 打分，严重影响 PageRank 的工作效果。

而远移(Teleport)技术则有效地将上述两种不良网络结构的影响降低，它在随机游走的基础上，允许旅行者以一定概率不走当前页面结点的对外超链接，而是远移到另外一个随机网页上。这个操作使得困在蜘蛛陷阱或者死结点中的旅行者摆脱束缚，继续在网络中游走。

PageRank 用如下迭代公式计算带远移的游走概率向量：

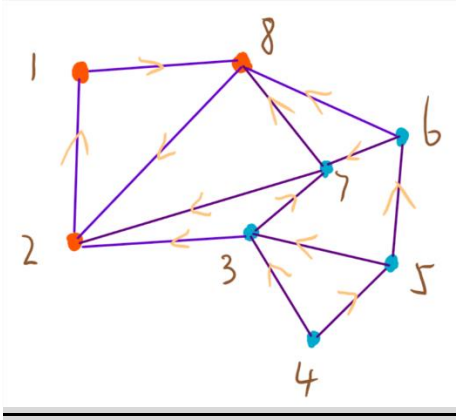
$$p^{(t)} := \beta M p^{(t-1)} + \frac{1-\beta}{n} e$$

其中β是旅行者继续沿着当前页面的超链接移动的概率，e 为元素全为 1 的 n 维向量。迭代公式所表达的含义是：旅行者以$1-\beta$的概率进行远移，等概率远移到 n 个网页中的某一个页面。

我们将在 2.4 节，用实验对比 PageRank 带远移与不带远移的效果差别。

2.4 基于 Python 的远移实验

本节我们考虑如下的网络结构：



这个网络的马尔科夫转移矩阵为：

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 1 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 1/2 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix}$$

此网络结构成病态，这是因为橙色的三个点构成了蜘蛛陷阱，我们分别用传统的 PageRank 算法与带远移的 PageRank 算法计算这个网络的网页重要性排序。

A. 传统 PageRank：

利用迭代公式 $p^{(t)} := Mp^{(t-1)}$更新游走概率向量，其中初始值$p^{(0)}$设为 $\left(\frac{1}{8}\ \frac{1}{8}\ \ldots\frac{1}{8}\right)^{\mathrm{T}}$，Python 代码为：

```
"""
Computer Network -------Midterm Essay
Random Walk
"""

"""Import Libraries"""
import numpy as np


"""Define Markov Transition Matrix"""
M = np.array([[0,1,0,0,0,0,0,0],
              [0,0,0.5,0,0,0,0.5,1],
              [0,0,0,0.5,0.5,0,0,0],
              [0,0,0,0,0,0,0,0],
              [0,0,0,0.5,0,0,0,0],
              [0,0,0,0,0.5,0,0,0],
              [0,0,0.5,0,0,0.5,0,0],
              [1,0,0,0,0,0.5,0.5,0]])


"""Traditional Random Walk"""
Iteration = 10
p = np.array([ 1/8, 1/8, 1/8, 1/8, 1/8, 1/8, 1/8, 1/8 ])
for i in range(Iteration):
    p1 = M.dot(p)
    print(p1)
    print(" ")
    p = p1
```

运行的结果为：

```
[ 0.125   0.25    0.125   0.      0.0625  0.0625  0.125   0.25  ]
[ 0.25    0.375   0.03125 0.      0.      0.03125 0.09375 0.21875]
[ 0.375   0.28125 0.      0.      0.      0.      0.03125 0.3125 ]
[ 0.28125 0.328125 0.     0.      0.      0.      0.      0.390625]
[ 0.328125 0.390625 0.    0.      0.      0.      0.      0.28125 ]
[ 0.390625 0.28125 0.     0.      0.      0.      0.      0.328125]
[ 0.28125 0.328125 0.     0.      0.      0.      0.      0.390625]
[ 0.328125 0.390625 0.    0.      0.      0.      0.      0.28125 ]
[ 0.390625 0.28125 0.     0.      0.      0.      0.      0.328125]
[ 0.28125 0.328125 0.     0.      0.      0.      0.      0.390625]
```

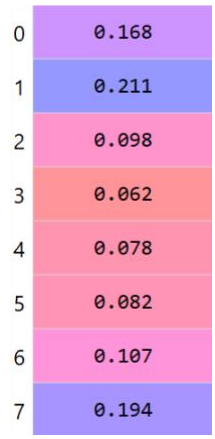
可以看出，从第四次迭代开始，PageRank 的分数全部集中在第 1，2，8 号网页，这正是这个网络结构中的蜘蛛陷阱。

B. 带远移 PageRank：

利用迭代公式 $p^{(t)} := \beta M p^{(t-1)} + \frac{1-\beta}{n} e$ 更新游走概率向量，其中初始值 $p^{(0)}$设为$\left(\frac{1}{8}\ \frac{1}{8}\ \ldots\frac{1}{8}\right)^{\mathrm{T}}$，$\beta$设为 0.5，Python 代码如下:

```python
"""Random Walk With Teleporting"""
Iteration = 100
e = np.array([ 1/8, 1/8, 1/8, 1/8, 1/8, 1/8, 1/8, 1/8 ])
beta = 0.5
p = e
for i in range(Iteration):
    p1 = beta * M.dot(p) + (1-beta) * e
    print(p1)
    print(" ")
    p = p1
```

运行最终结果为：



最终 57%的 PageRank 分数集中在 1，2，8 号网页结点上，这说明了：一方面 1，2，8 号结点确实很重要，另一方面带远移的 PageRank 避免了 3，4，5，6，7 号 PageRank 无分的极端情况。此外，7 号与 3 号是除 1，2，8 号之外分数最高的两个结点，这和 7 号，3 号结点度数最高的事实是一致的。

3）PageRank 的改进

3.1 带主题偏好的随机游走

我们在第 2 章介绍的 PageRank 算法有一个缺陷，那就是它完全没有把各个网页所包含的内容主题考虑在内，而只是分析了这个网络的结构问题，这种分析太过单调，在实际工作中必须加以改进。

举个例子，假如小麦在搜索与“时尚”相关的关键词，我们希望网页的 PageRank 得分不仅和网络的拓扑结构有关，也和时尚这个主题相关，即某个网页的主题与时尚越接近，该网页的 PageRank 得分越可能高，如果可以达到这个目的，那么搜索引擎在推荐网页时，就会有更私人化的返回结果。

正是基于上面这个想法，很多搜索引擎公司不只维护一个 PageRank 的打分向量，而是对每一个不同的主题，维护一个不同的 PageRank 向量，例如：体育 PageRank 向量，时尚 PageRank 向量，学术 PageRank 向量，美食 PageRank 向量等。

当用户输入一个关键词进行搜索时，我们对关键词用 NLP 的手段进行语义分析，将此关键词归为某一主题，在推荐网页时，我们调用该主题所对应的 PageRank 向量组成网页打分的一部分，再综合其他多种因素，最终返回搜索结果。

还是以“时尚”这个主题作为例子，我们的直观是：如果一个网页被公认的著名时尚网站所指向，那么这个网页很可能会与时尚主题相关，即著名时尚网站的附近，会聚集着与时尚极其相关的网站。网页的时尚主题相关度会随着其与公认的著名时尚网站的距离增大而减小。

带主题偏好的随机游走算法(Topic-Biased Random Walk)很好地表达了上述的直观，旅行者在沿着当前链接随机游走的同时，以一定概率，远移(teleport) 到某个公认的时尚网站上，如此一来，与公认时尚网站距离越近的网站，就会有越高的时尚 PageRank 打分。

带主题偏好的随机游走算法的迭代公式为：

$$p^{(t)} = \beta M p^{(t-1)} + \frac{(1-\beta)}{sum(T)} T$$

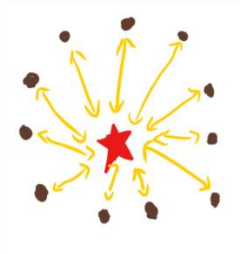
其中$(1-\beta)$为发生远移的概率，T是一个布尔 n 维向量，它的第 i 个分量代表网页 i 是否是公认的时尚主题网站(0 代表不是；1 代表是)。

可以看出：带主题偏好的 PageRank 是对带远移的 PageRank 的一种修改，旅行家不再是随机远移到任意某个网页，而是随机远移到某些公认的时尚网页上。

T的标记可以采用人工标记的办法，即用人力将著名的时尚网站进行标记，但是人力成本太过昂贵，且人工标记难免有带有个人主观色彩与大量的遗漏，在 21 世纪，这种方法是不可取的。我们可以采用 NLP 的语义分析技术等方法，用人工智能的办法标记T，不仅更加准确，也大大节约成本。

3.2 带信任偏好的随机游走

虽然 PageRank 通过分析网络结构的方法，给各个网页打出了比较合适的分数，战胜了传统垃圾网站，但是专门针对 PageRank 所设计的垃圾网站也层出不穷，下面介绍一种新型垃圾网站的构造办法：



垃圾网站维护者发现，网站处于互联网越中心的位置，PageRank 的打分也就越高，于是他们建立了很多卫星垃圾网站(上图中的小褐点)去和目标垃圾网站(上图中的红星)相互指向，这一构造明显提高了目标垃圾网站的 PageRank 得分。

带信任偏好(Trust-Biased)的随机游走模型是 3.1 中算法的变式，它很大程度上消灭这些新型的垃圾网站所带来的问题，算法的直观与 3.1 类似：如果一个网站被一些公认的门户网站所指向，那么这个网站很可能不是垃圾网站，于是我们在运行随机游走算法时，允许旅行家以一定概率远移到某些公认的门户网站上，这种修改使得权威网站及其附近的网站 PageRank 得分会大大提高。

搜索引擎公司标记了 n 维布尔向量A，用来指示哪些网站是公认的非垃圾网站，并采用如下迭代公式计算 PageRank 得分：

$$p^{(t)} = \beta M p^{(t-1)} + \frac{(1-\beta)}{sum(A)} A$$

新的 PageRank 得分向量将大部分分数集中在权威网站附近，垃圾网站的得分会急剧下降，这样搜索返回的结果将不太可能包含垃圾网站。

此外，带信任偏好的随机游走还可以用于垃圾网站的识别工作，我们可以先运行一次传统的带远移的 PageRank 算法，得到游走概率向量$p^{(传)}$，再运

行一次带信任偏好的 PageRank 算法，得到游走概率向量 $p^{(信)}$，我们可以考察那些在$p^{(传)}$中分数很高，但在$p^{(信)}$中分数很低的网站，他们很大概率是垃圾网站。