应用随机过程大作业

基于随机游走模型的 PageRank 算法及应用

自 71 屈晨迪 2017010928
自 75 蔡烨怡 2017010922

1 课题背景介绍

1.1 随机游走模型

随机游走也称随机漫步，其概念接近于布朗运动，可以看作布朗运动的理想数学状态。随机游走最早由 Karl Pearson 于 1905 年提出，它是一种不规则的变动形式，在变动过程当中的每一步都是随机的，下一步运动仅由当前所处的状态决定，每一次变动都不会影响过去的状态，即具有“无记忆性”，可以用马尔科夫链刻画。随机游走模型是一类重要的随机系统模型，在直线上的随机游走也称为一维随机游走，根据不同的数学条件限制分为无限制、带吸收壁、带反射壁等类别。

以无限制的随机游走为例，设有一个质点在数轴上随机游动，每隔单位时间Δt移动一次，每次只能向左或向右移动Δx单位，或原地不动，假设质点在 0 时刻位置为 a，向右移动的概率为 p，向左移动概率为 q，原地不动的概率为 r，有 p,q,r 均大于 0，且$p+q+r=1$，每次移动互相独立，用X_n表示质点 n 次移动后的位置，则$\{X_n, n \geq 0\}$为一马尔科夫链，转移概率$p_{i,i+1} = p, p_{i,i-1} = q, p_{ii} = r$，其余的$p_{ij} = 0$。著名的一维随机游走问题有赌徒输光问题和酒鬼失足问题。

随机游走也可以应用于图上，成为基于图的随机游走模型，给定一个包含节点和边的图，确定出发点，随机选择图上的一个邻居节点，移动到邻居节点上，然后将当前节点作为初始点，重复上述过程，则这样选出的一系列节点就构成了一个随机游走过程，如图 1 所示。

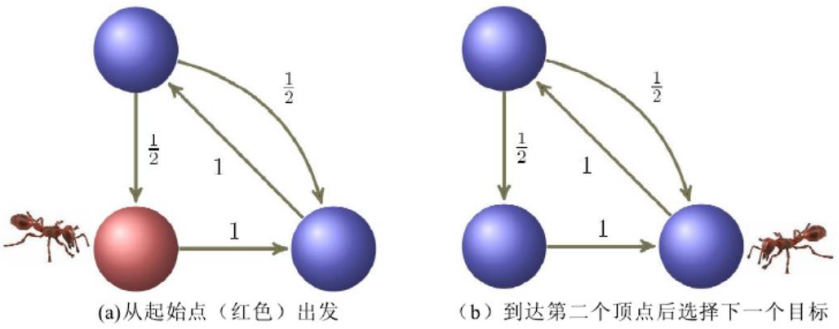

图 1 基于图的随机游走

1.2 PageRank 算法的来源

本次大作业主要研究的 PageRank 问题，就是一种基于图的随机游走问题。在搜索引擎最初出现的时候，多采用的是分类目录的方法，即人工对网页进行分类并整理出高质量的网

站。后来随着网页数量越来越多，人工分类难以实现，搜索引擎开启了文本检索的功能，即用户输入关键词，算法根据网页与关键词的相关程度来进行推荐，但常常有网页通过不断重复同一个关键词来提高自己的搜索率，于是两位谷歌的创始人提出了 PageRank 算法，对网页重要性进行排序。想象一个悠闲的上网者，他首先打开一个初始网页，浏览了数分钟后，随机选择当前界面上的一个链接，点进另一个网页浏览，如此反复，PageRank 就是该上网者分布在不同网页的概率。算法借鉴了通过论文引用次数来评判论文质量的方法，PageRank 的核心思路有以下两点：

- 若一个网页被很多其他网页链接到，则该网页较为重要，PageRank 值较高
- 若一个 PageRank 值较高的网页链接到其他网页，则被链接到的网页 PageRank 值也会相应地提高

本次课题首先对 PageRank 算法进行研究和实现，之后尝试将其拓展应用到其他领域。

2 PageRank 算法

2.1 问题建模

由上述算法来源的介绍可知，PageRank 问题被提出的背景是互联网和搜索引擎的发展，在网页发展的早期，搜索引擎需要对不同网页的重要性进行排序，以此确定在搜索结果中网页序列向用户呈现的先后顺序。为了解决这个问题，PageRank 算法由此定义了网页的 PR 值：PR 值越高，此网页的重要性越高，也就越应该被优先呈现。

原初描述：如果用$M = \{M_{ij}\}$表示站点之间的邻接矩阵（M是 01 矩阵，$M_{ij} = 1$表示i引用了j，$M_{ij} = 0$表示i没有引用j。）。则一个站点的PR值可以描述为：

$$PR(j) = \sum_{i, M_{ij}=1} \frac{PR(i)}{L(i)}$$

其中$L(i)$是站点i的出链总数。

修正 1：在实际情况中，站点i对不同站点的引用程度（比如链接的位置、大小）可能是不同的，所以用邻接矩阵M来描述问题并不准确。为了进一步地描述问题，用转移矩阵S来替代邻接矩阵M。$S = \{S_{ij}\}$表示从节点 j 到节点 i 的转移概率，因此有$\sum_i S_{ij} = 1$。那么节点 i 的 PR 值即所有进入 i 的节点的 PR 值的加权相加。

$$PR(j) = \sum_i S_{ji} PR(i)$$

令$P = [PR(1), PR(2) \dots PR(n)]^T$，则上式可以写成向量形式：

$$P = S \cdot P$$

修正 2：在实际情况中，可能存在终止点问题和陷阱问题。终止点问题是指，在互联网上可能有一些网页，它们不链接到任何一个页面，用户到达此页面后无法点击跳转到其他网页，导致前面得到的转移概率被清零，这样下去，最终得到的概率分布几乎都为 0；而陷阱问题是指一个网页不存在到其他网页的链接，只链接到自身，这样用户达到该网页后只能不断循环点开此网页，导致该网页概率逐渐趋于 1，而其他网页节点概率分布趋于零。

因此需要注意的是，在实际上网过程中，用户不会总是持续点击网页，而有可能在某个阶段放弃点击，直接输入站点网址进入其他页面。如果用P_n来表示在 n 时刻各个站点的概率

分布的话，$\{P_n, n \geq 0\}$是一个马尔科夫链。如果用户以概率α持续点击，以概率$1-\alpha$放弃浏览，那么P_n之间存在迭代关系：

$$P_{n+1} = \alpha S P_n + \frac{1-\alpha}{N} P_n = \left(\alpha S + \frac{1-\alpha}{N} I\right) P_n$$

令$A = (\alpha S + \frac{1-\alpha}{N} I)$为$\{P_n\}$的一步转移矩阵，显然$A$仍然是一个随机矩阵。则马尔可夫链由$A$和$P_0$唯一决定。

2.2 解存在的条件

跟据《应用随机过程》中的定理 3.5.6，离散马尔科夫链存在平稳分布的充分条件是马尔科夫链是不可约遍历链。下面分别说明这两个条件满足的情况：

不可约：不可约只有在转移矩阵 S 满足一定条件时才可以达到。由于$A = \alpha S + \frac{1-\alpha}{N} I$，后一项不对连通性做出贡献，因此 A 的不可约性仅仅取决于 S 的不可约性。当 S 满足对任意两个状态都存在 k 使$p_{ij}^{(k)} > 0$时，A定义的状态空间才是不可约的。

遍历链：马尔科夫链是遍历链要求转移概率矩阵A为素矩阵，即存在 k 使得$A^k > 0$。当 A 是不可约矩阵时，由于对任意i, j都能找到$k(i,j) > 0$，可以取K为$k(i,j), i \leq j, i, j \leq n$的最小公约数，则此时$A^K > 0$。

2.3 求解方法

下面假设A满足以上两个条件。

（1）特征值法

平稳分布P^*满足：

$$P^* = AP^*$$

也即P^*是矩阵A关于特征值 1 的特征向量。只要求解特征向量就可以解得平稳分布P^*。

（2）解析法

将$A = (\alpha S + \frac{1-\alpha}{N} I)$代入，可以得到上述问题的解析解表达式：

$$P^* = \left(\alpha S + \frac{1-\alpha}{N} I\right) P^*$$

$$\Rightarrow (I - \alpha S) P^* = \frac{1-\alpha}{N} e$$

$$\Rightarrow P^* = \frac{1-\alpha}{N} (I - \alpha S)^{-1} e$$

（3）迭代法

解析解中的求逆运算当矩阵阶数高是较难求解的。因此，也可以用迭代的方法逐次逼近。

$$P_{n+1} = A \cdot P_n$$

P_n 的收敛值即为原问题的解。

3 实验仿真

3.1 迭代法 python 代码实现

这里采用迭代法进行求解，假定的网页之间链接关系如图 2 所示，共有 A-E 五个节点，首先给每个网页随机赋予初始的 PR 值，此处统一设置为 1/N，N 为总网页数，即初始 PR 均为 0.2，之后按照公式$P_{n+1} = A \cdot P_n$不断迭代计算，其中$A = (\alpha S + \frac{1-\alpha}{N} I)$，直到满足终止条件$|P_{n+1} - P_n| < \epsilon$停止，此时的 PR 值为最终所求。具体流程图如下所示。

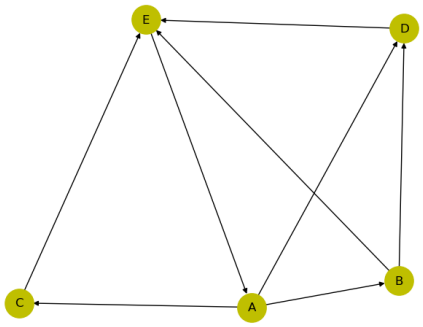

图 2 设定的网页链接关系

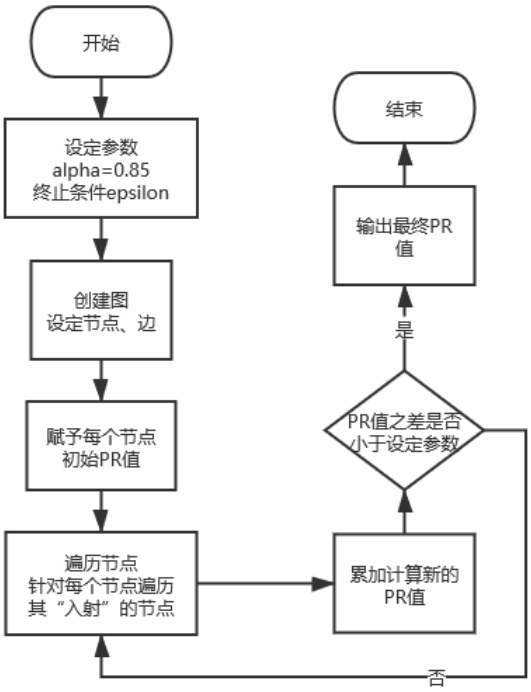

图 3 迭代法程序流程

最终结果如下所示，与解析法计算的结果一致，改变初始 PR 值对结果影响不大。绘制结果如图 4，节点大小反映了相应 PR 值的高低，观察下列结果，可以看到节点 E 的 PR 值最高，这与其被最多网页链接到的事实相符，而被高 PR 值的节点 E 链接到的节点 A PR 值也较高，很好地体现了 PageRank 算法的两点核心思想。

```
The final page rank is
 {'A': 0.2963352045696236, 'B': 0.11396164129472669, 'C': 0.11396164129472669, 'D': 0.16239533884498553, 'E': 0.31333713066901425}
```

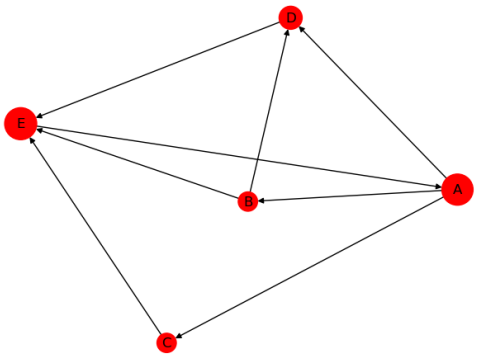

图 4 仿真结果

3.2 MapReduce 方法实现

在上述迭代算法中，只有 5 个网页节点，经过二十多次的迭代就能得到最终结果，但在实际应用中，网页总数可以达到百亿个，单纯用上述算法显然不能应对如此庞大的计算量，于是需要使用 MapReduce 方法，分布式计算大规模网页的 PageRank。MapReduce 方法包含了 Map 和 Reduce 两个过程，其中 Map 指对集合里的每个目标应用同一个操作，Reduce 指遍历 Mapping 返回的集合中的元素并返回一个综合的结果，如此可以把大规模计算部署到多个计算机上进行，大大提高运算效率，具体流程如下图。

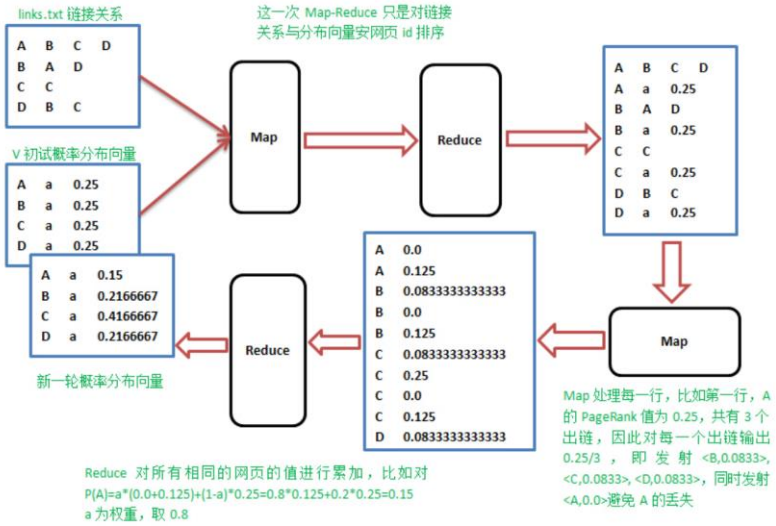

图 5 MapReduce 算法思路[5]

同样用 MapReduce 对 3.1 中设计的网页结构进行了仿真，得到的结果与迭代法相同。

3.3 存在的问题及改进

PageRank 算法存在一些明显的缺点，如没有过滤掉界面上的广告链接和功能链接，没有区分导航链接，一些网页是专门的导航页面，上面有很多其他站点的链接，这种链接对网页重要性的影响显然应该比其他链接要小，而后者更能体现出 PageRank 值的传递关系。

同时，该算法对新网页不太友好，一般新网页上线后入链较少，即使其重要性很高，仍

需要一段时间进行推广，而一些旧网页会随时间变得过时，但很多旧网页不会被删掉，它们在过去长时间中积累了很高的 PR 值，会给搜索带来麻烦。为了应对这种情况，Timed-PageRank 算法被提出，该方法在 PageRank 的基础上加了一个时间维度，它不再将阻尼系数α设为常量，而是引入一个随时间递减的函数$\mathrm{f(t)}(0 \le \mathrm{f(t)} \le 1)$来惩罚旧网页，此处 t 为该网页上次更新时间与当前时间的差值。

此外，还有人提出了 TrustRank 算法，TrustRank 最初用于检测垃圾网站，其主要原理是先通过人工判断来识别出高质量的页面，作为“种子”页面，与“种子”页面直接连接的页面更有可能是高质量网页，TR 值较高，与“种子”页面连接越远，TR 值逐渐降低。通常，将 PR 值与 TR 值结合起来可以更加准确地判断网页的重要性。

4 PageRank 在其它领域的应用——以神经科学为例

人类大脑的神经系统连接是极为重要我们又知之甚少的网络。因此很多的网络模型被应用到人类大脑神经连结的建模上，PageRank 算法就是其中之一。在采集到一定脑区的神经活动时间序列之后，PageRank 算法被用作衡量一个体素或核团的重要性。

4.1 PageRank 衡量核团的全局中心性

在 Zuo 2011[1]的工作中，作者对 fMRI 成像中的体素(4mm)计算了两两体素之间的相关性。每个体素的时间信息表示为$V_{ij}(t)$，计算两个时间序列之间 Pearson 相关系数。

$$r(i,j) = \frac{\sum_{t=1}^{T}[V_i(t) - \bar{V}_i(T)][V_j(t) - \bar{V}_j(T)]}{\sqrt{\sum_{t=1}^{T}[V_i(t) - \bar{V}_i(T)]^2}\sqrt{\sum_{t=1}^{T}[V_j(t) - \bar{V}_j(T)]^2}}$$

$r(i,j)$组成的矩阵称为相关矩阵R，$R = (r_{ij})$。对R矩阵以r_0（如取 0.0001）为阈值做硬阈值分割，得到连结矩阵$A = (a_{ij})$

$$a_{ij} = \begin{cases} 0, & r_{ij} \le r_0 \\ r_{ij}, & r_{ij} > r_0 \end{cases}$$

而后作者衡量了节点在不同尺度上的重要性，包括局部重要性（度）、亚尺度重要性（子图分析）、全局重要性（特征值分析和 page-rank 算法）。作者采用的是以一定概率游走的 page-rank 模型（未做归一化）

$$PC(i) = r(i) = 1 - \alpha + \alpha\sum_{j=1}^{N}\frac{a_{ij}r(j)}{DC(j)}$$

其中$DC(j)$是 j 的出链的数量。

由于神经回路的建模中涉及的体素数量大于 1000，故作者还采用了 inner-outer 迭代来加速算法。在这项研究中，作者发现核团的局部重要性会随着年龄的增加而降低，但是全局的重要性却不会出现这种随年龄的下降。这说明局部回路和全局回路在生理上的作用可能会有不同。

4.2 PageRank 衡量核团的层级关系

在 Crofts 2011[2]的工作中，作者用 PageRank 算法来挖掘神经网络当中的层级关系。

“层级关系”指的是在网络结构当中存在的“分发式”、“下行式”的结构。在一个给定的网络

中，一个节点的层级由下述优化目标决定：

$$\min_{\substack{p\in\mathbb{R}^N \\ \|p\|_2=1 \\ p^Te=0}} \sum_{i=1}^{N}\sum_{j=1}^{N}(p_i-p_j)^2 a_{ij}$$

其中$P=(p_i)$是对 1,2…N 的一个重排列，a_{ij}是邻接矩阵中的元素，$a_{ij}=1$表示节点 i 到节点 j 有连结，$a_{ij}=0$表示节点 i 到节点 j 没有连接。元素重排完成之后，编号为 1 的结点的层级最高，编号为 N 的结点的层级最低。

PageRank 是这个优化问题的求解方法之一，即用 PageRank 的 PR 值的大小来指定这里的 P 值。P 值由以下等式给出：

$$P=\left(I-\theta A^T D^{out^{-1}}\right)^{-1}e$$

其中θ是 0 到 1 之间的常数，$D^{out}=diag(L(i))$，即由每个节点的出度构成的对角阵。这个式子相当于在经典 PageRank 算法中，令转移矩阵 S 为邻接矩阵的行归一化矩阵。如果令$\hat{A}=A^T D^{out^{-1}}$，则在θ足够小时，上式可以用泰勒展开式表示：

$$P=(I+\theta\hat{A}+\theta^2\hat{A}^2+\theta^3\hat{A}^3+\cdots)e$$

除了用 PageRank 求解之外，此优化目标还可以用 Katz 分数、交流度（communicability）等方法求解。作者在线虫的神经元连接上验证比较了这些方法，发现 PageRank 相较其它方法在优化目标上表现不佳。作者认为这可能是模型的不匹配和游走参数的错误带来的。这说明 PageRank 虽然有很强的普适性，但是也应该被小心妥善地运用到实际问题上。

5 反思与总结

本课题从随机游走模型出发，选定了 PageRank 算法作为研究对象，该算法可用于解决基于图的随机游走问题，我们主要从来源、问题建模、原理证明和计算方法等几个方面做了讨论和探究，并编写 python 程序进行了简单模型下的实验仿真，之后调研了该算法在其他领域的应用，看到该算法可以拓展应用到社交网络、文献管理、推荐系统等多个领域中去，报告以神经科学为例做了深入的探讨。通过本次大作业，我们对随机游走模型、转移概率矩阵、平稳分布存在条件等随机过程中的典型问题都有了更为清晰的理解，也了解了很多与网页排序和推荐算法相关的知识，收获颇丰。最后，感谢老师和助教本学期的指导和帮助!

6 参考文献

[1] X.-N. Zuo, R. Ehmke, M. Mennes, D. Imperati, F. X. Castellanos, O. Sporns, and M. P. Milham. Network centrality in the human functional connectome. Cerebral Cortex, 2011. doi:10.1093/cercor/bhr269.

[2]J. J. Crofts and D. J. Higham. Googling the brain: Discovering hierarchical and asymmetric network structures, with applications in neuroscience. Internet Mathematics, 7, pp. 233{254, 2011. doi:10.1080/15427951.2011.604284.

[3] https://www.cnblogs.com/rubinorth/p/5799848.html

[4] https://blog.csdn.net/liujh845633242/article/details/103504499

[5] https://www.cnblogs.com/fengfenggirl/p/pagerank-introduction.html