基于 MapReduce 的搜索引擎算法

PageRank

如何表示图：邻接表/邻接矩阵

PageRank 是一种在搜索引擎中根据网页之间的链接关系计算网页排名的技术

基本思想：被优质网页链接的网页多半也是优质网页。将互联网上各个网页之间的链接关系看作有向图，对于任意网页 P_i ，其 PageRank 值为

$$R(P_i) = \sum_{P_j \in B_i} \frac{R(P_j)}{L_j}$$

其中 B_i 为所有连接到 P_i 的网页集合，L_j 为 P_j 对外链接的数目

简化模型

简化模型：可以定义一个超链接矩阵 H 满足

$$H_{ij} = \begin{cases} 1/L_j & P_j \in B \\ 0 & P_j \notin B \end{cases}$$

并且 $R = [R(P_i)]$ ，则 $R = HR$

如果使用上述简化的模型，会面临两个问题

- rank leak：没有出度的节点会导致迭代收敛到全 0，解决方法有两种：递归地将无出度的节点从图中去掉，计算完成后再加入，或是对无出度的节点添加边指向链接其的顶点
- rank sink：没有入度的节点迭代后 PageRank 值会收敛到 0

随机访问模型

上网者从随机的网页开始浏览，不断点击至别的网页直至开启新的随机网页，将以上述方式访问新网页的概率设置为其 PageRank 值

在图中任意两点间添加新的通路，每个顶点处按概率 d 以链接的方式转移，按概率 $1-d$ 以随机的方式转移（新通路）

则令 $H' = dH + (1-d)[1/N]_{N \times N}$ ，有 $R = H'R$

其满足马尔科夫链的性质，如果马尔科夫链收敛，则 R 有唯一解

随机访问模型的计算公式为

$$PR(P_i) = \frac{1-d}{N} + d \sum_{P_j \in M(P_i)} \frac{PR(P_j)}{L(P_j)}$$

随机访问模型一定程度上解决了 rank leak 和 rank sink 的问题

用 MapReduce 实现 PageRank

GraphBuilder

建立网页间的链接图

Map：输出 `<URL, (PR_init, link_list)>`，以网页 URL 为 key，PageRank 初始值和出度的列表作为 value

Reduce：原样输出

PageRankIter

迭代计算 PageRank，以阶段一的输出为输入

Map：输出两种 key

- 对于 `link_list` 中的每个链接 u，输出 `<u, cur_rank/[link_list]>`
- 输出 `<URL, link_list>` 维护图结构

Reduce：对于所有 `<URL, cur_rank/[link_list]>`，对其求和，乘上概率 d，加上 $\frac{1-d}{N-1}$ 得到新的 PageRank 值，输出 `<URL, (new_rank, link_list)>` 作为下一轮迭代的输出

可以选择以下迭代终止条件

- 各网页 PageRank 值不变
- 各网页 PageRank 排名不变
- 迭代至固定次数

RankViewer

将输出排序即可