pageRank

1. 算法简介

该工程通过 java 语言实现了简单的 pageRank 算法，该算法的核心公式如下所示：

$$G=\alpha S+(1-\alpha)\frac{1}{n}U \tag{1}$$

$$q^{next}=Gq^{cur} \tag{2}$$

α：0－1之间的任意值

S：根据网页的指向得到的原始矩阵

n：网页的数量

U：全1的n阶矩阵

q：表示初始网页的权重向量

其中 α 表示阻尼系数，其意义是在任意时刻用户到达某页面后并继续向后浏览的概率。而 $1-\alpha$ 表示用户到达某个页面后停止向后浏览，随机跳到新 URL 的概率（可以理解在浏览器的地址栏输入新的网址）。

http://blog.csdn.net/hguisu/article/details/7996185（该博客讲的更加详细）。

2. 实例讲解

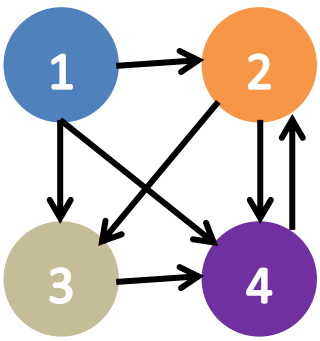


图 2.1

图 2.1 中显示了一个网站中四个网页之间的相互链接关系。从图中可以得到 S 矩阵如下所示：

$$S=\begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 1 \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 1 & 0 \end{bmatrix}$$

S_{ij} :表示从第 j 个页面跳转到第 i 个页面时，权重的分配系数。

3. 工程文件简介

src 文件夹下是源代码：

PkDataProc.java => 读取训练数据，并把网址映射成从 0 开始的整数。

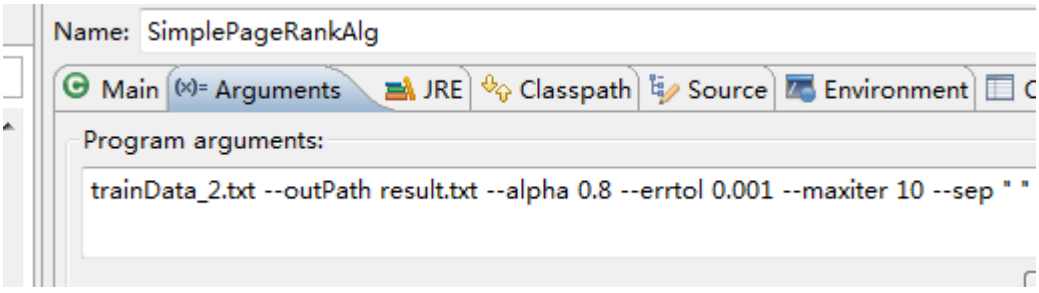
PkOption.java => 参数类，该 pagerank 算法支持的参数。

PkGraph.java => 通过训练数据，构造矩阵 G。

SimplePageRankAlg.java => 迭代计算，并获得收敛的 q 向量。

4. 程序调用方式

- 在 **windows** 下可以用 eclipse 打开该工程，在 Run =>Run Configurations 界面设置程序需要的参数，参数细节在后面会给出解释。



- 在 **Linux** 下可以可以通过如下命令进行调用。

  把文件夹下的 SimplePageRankAlg.jar 拷贝到 linux 下某个目录，通过如下命令进行调用。

  Java –jar SimplePageRankAlg.jar trainDataFilePath [--sep sepValue] [--alpha alphaValue] [--maxIter maxIterValue] [--errTol errTolValue] [--outPath outPathValue]

方括号中表示可选参数。

trainDataFilePath：表示训练数据文件路径

sep：表示训练数据每条记录的字段分隔符，默认是空格或 tab 键

alpha: 表示任意时刻用户到达某页面后并继续向后浏览的概率，默认是 0.85

maxIter: 计算的最大迭代次数，默认是 20

errTol: 结束迭代计算的误差参数，该参数与相邻两次计算的 q 相邻的距离进行比较，默认是 0.0001

outPath：把最终计算的 q 向量输出到目标文件中。如果不传，默认为空，则输出到终端。

5. 数据集

在文件夹下有两个数据集文件 trainData_1.txt、trainData_2.txt，第一个文件是描述图 2.1 节点之间的关系，第二个数据文件是也是描述网页之间的链接关系，数据量都比较小，主要用于测试程

序的正确性。数据文件中每行表示一条记录，数据格式如下：

FromId ToId

程序的时间复杂度是$O(n^3)$,其中 n 表示节点的个数。