Programming Assignment 3

Estimating PageRank Values of Wikipedia Articles using Apache Spark

Due: October 30th By 11:59PM
Submission: via Canvas, Individual submission

Objectives
The goal of this programming assignment is to enable you to gain experience in:

- Installing and using analytics tools such as HDFS and Apache Spark
- Implementing iterative algorithms to estimate PageRank values of Wikipedia articles using Wikipedia dump data

(1) Overview

In this assignment, you will design and implement a system that calculates PageRank[1] values of currently available Wikipedia articles. The PageRank algorithm is used by Google Search to rank web pages in their search engine query results. PageRank measures the importance of web pages. The underlying assumption is that more important web pages are likely to receive more links from other web pages.

In this assignment, you will implement algorithms to estimate PageRank values over the Wikipedia dataset. These results can be used to rank search results within Wikipedia. Most of the Wikipedia articles contain links to other articles or external web contents. We will consider only internal Wikipedia articles to calculate the PageRank. Currently Wikipedia contains more than 4.9 million articles[2]. Wikipedia publishes data dumps in various formats periodically[3]. In this assignment, you will perform:

- Estimation of PageRank values under ideal conditions
- Estimation of the PageRank values while considering dead-end articles
- Analysis of the above results: Creating a Wikipedia Bomb (Extra Credit)

To perform the PageRank algorithm over the Wikipedia dump, you are required to use Spark in an iterative fashion.

1.Larry Page, "PageRank: Bringing Order to the Web" Stanford Digital Library Project, 1997, https://web.archive.org/web/20020506051802/www-diglib.stanford.edu/cgi-bin/WP/get/SIDL-WP-1997-0072?1
2.https://en.wikipedia.org/wiki/Wikipedia:Statistics

3.https://meta.wikimedia.org/wiki/Data_dumps

(2) Requirements of Programing Assignment 3

Your implementation of this assignment should provide:

A. A sorted list of Wikipedia pages based on their ideal PageRank value in descending order. Each row should contain the title of the article and its PageRank value. This computation should be performed in your own Spark cluster with 5 machines. You should present results from at least 25 iterations.

B. A sorted list (in descending order) of Wikipedia pages based on their PageRank value with taxation. Each row should contain the title of the article and its PageRank value. This computation should be performed on your own Spark cluster with 5 machines. You should present results from at least 25 iterations.

C. **Extra Credit (2 points)**: A Wikipedia Bomb that returns the "Rocky Mountain National Park" wikipedia page for the search key word "surfing" (see 3.C). In order to do that, you should modify the link data file and show that the "Rocky Mountain National Park" page generates highest PageRank among all of the pages containing "surfing" as part of their titles.

There are many algorithms to estimate PageRank. You should use the algorithms included in this description. You are not allowed to use existing PageRank implementations.

Demonstration of your software should be on the machines in CSB120. Demonstration will include an interview discussing implementation and design details. Your submission should include your source codes and outputs. Your submission should be via Canvas.

(3) PageRank

The term PageRank comes from Larry Page, a founder of Google. PageRank is a function that assigns a real number to each page in the Web (in our case Wikipedia articles). A page with a higher PageRank value will be considered as a more important page. There are many algorithms to assign PageRank. In this assignment, we will consider two algorithms[4]: (1) Idealized PageRank and (2) Taxation-based PageRank.

A. Idealized PageRank

Suppose that our Web is a directed graph where pages are the nodes and there is (are) edge(s) from page p_1 to page p_2. The edges between nodes represent links between the web pages. Figure 1 depicts the links between 4 pages (A, B, C and D) as a graph. Page A has links to each of the other three pages B, C and D. Page B has links to A and D. Page C has a link only to A. Finally, page D has links to B and C.

4. Jure Leskovec, Anand Rajaraman, and Jeff Ullman, "Chapter 5: Link Analysis", Mining of Massive Datasets, Cambridge University Press, http://infolab.stanford.edu/~ullman/mmds/ch5.pdf

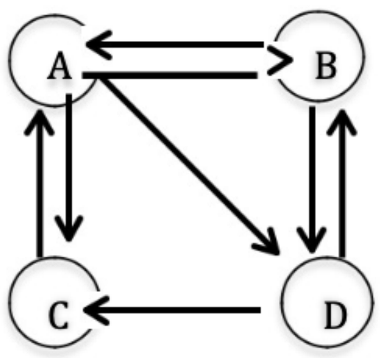

Figure 1. Web graph of example

Suppose a random surfer starts at page A (in Figure 1). The probability that the random surfer will visit each of the pages B, C, and D in his next stop is $\frac{1}{3}$. Note that if a page contains a link to itself, it should be represented in the graph as a loop. The random surfer that arrived at page B will likely to visit A with probability of $\frac{1}{2}$.

This graph can be defined as the *transition matrix* of the Web (here, Wikipedia), and the transition matrix for the previous example depicted in the Figure 1 is:

$$M = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$

In the above matrix, the order of the pages is the natural one, A, B, C, and D. The first column shows that a random surfer at A has probability of $\frac{1}{3}$ of visiting each of the other pages.

An idealized PageRank supposes that a random surfer may start from any of the n pages with equal probability. The initial vector v_0 for our previous example is $(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$. After one step, the distribution of the surfer will be Mv_0, after two steps it will be $M(Mv_0) = M^2v_0$, and so on. Multiplying the initial vector v_0 by M a total of i times will provide us with the distribution of the surfer after i steps.

After enough number of iterations, the vector will show little change at each round. For this assignment your iteration count is as specified earlier.

B. Taxation

Wikipedia contains dead-end articles [5]. A dead-end article is a page that contains no links to other Wikipedia articles. Since there is no link to go out, once a random surfer visits the dead-end article, this random surfer never leaves the article. This often results in 0 probabilities in many of the pages as the number of iteration increases.

To deal with dead-end articles, the PageRank algorithm modifies the idealized process by which random surfers are assumed to move without following the links. This method is referred to as "*taxation*". Taxation allows each random surfer to have a small probability of teleporting to a random page, rather than following an out-link from their current page.

With taxation, the new vector for estimating the PageRank will be,

$$v' = \beta M v + (1-\beta) e / n$$

where β is a chosen constant specifying the probability that the random surfer decides to follow an out-link from their present page. Usually in the range of 0.8 to 0.9, e is a vector of all 1's with the appropriate number of components (i.e. Wikipedia pages), and n is the number of nodes in the Web graph. In this assignment, you should use β as 0.85.

You should perform this processing using Apache Spark. Please ignore all of the possible errors due to the data segmentation by HDFS.

C. Wikipedia Bomb (Extra Credit)

The term Wikipedia Bomb is derived from the Google bombs for this assignment. The Google bomb refers to the practice of causing a web site to rank highly in web search engine results for unrelated or off-topic search terms by linking heavily [6].

In this assignment, you should generate a Wiki Bomb for "surfing" in Wikipedia and the result should be the "Rocky Mountain National Park" page in Wikipedia.

To generate this, you are allowed to modify the **link data file**. You should show the results and discuss during the demo.

The term Wikipedia Bomb is derived from the concept of Google Bombs.
A Google Bomb refers to the practice of manipulating search engine rankings by causing a website to appear highly for unrelated or off-topic search terms through the use of many hyperlinks pointing to it.

5 https://en.wikipedia.org/wiki/Wikipedia:Dead-end_pages
6 https://en.wikipedia.org/wiki/Google_bomb

The key mechanism behind a Google Bomb lies in how anchor text is used. In addition to the PageRank value - Inverted index of terms, the final result also consider the votes from the incoming links. Each link acts as a vote for another page. When many pages link to a target page using the same anchor text:

- The PageRank algorithm counts those links as “votes” for that target page.
- The anchor text associated with those links helps the search engine associate that page with the specific linked phrase.

Together, this combination can make a page rank higher for that search term — even if the page’s content is unrelated to the term.

In this bonus task, you will simulate this behavior within Wikipedia data. You are not required to implement a general-purpose anchor text voting mechanism. If your system successfully generates a “Wikipedia Bomb” for the term “surfing”, making the Rocky Mountain National Park page appear as the result, that will be sufficient.

However, note that hard-coded answers will not receive bonus credit; your approach must logically demonstrate how link structure and anchor text manipulation lead to this outcome.

(4) Programming Environment

A. Requirements

All of the software demonstration of this assignment will be done in CSB120 machines. You should make sure your software works on the machines in CSB120. Your spark configuration should already be set up (look into ~/sparkConfig folder). You should stick with the same configuration for this assignment.

(5) Dataset

The original dump dataset from Wikipedia is 12.1GB as a bz2 file, and about 50GB when it is decompressed. It follows an XML format and includes other information that is not relevant to this task. To optimize your work on this assignment, we will use Wikipedia dump [7] generated by Henry Haselgrove. Please find the instruction to download the data file from CS120 (instead of the external server) in the course assignment web page. If you have any problem with storage capacity, you should contact the Prof. Pallickara immediately.

The dataset contains two files:

links-simple-sorted.zip (323MB) [Link]
titles-sorted.zip (28MB) [Link]

These files contain all links between proper Wikipedia pages. This includes disambiguation pages and redirect pages. It includes only English language Wikipedia.

In *links-simple-sorted.txt*, there is one line for each page that has links from it. The format of the lines is:
$from_1$: to_{11} to_{12} to_{13}
$from_2$: to_{21} to_{22} to_{23} ...
…..

where $from_1$ is an integer labeling a page that has links from it, and to_{11} to_{12} to_{13} ... are integers labeling all the pages that the page links to. To find the page title that corresponds to integer n, just look up the n-th line in the file *titles-sorted.txt*, a UTF-8 encoded text file.

(6) Grading

Please look at the rubric for PA3 posted on Canvas assignment page.

(6) Late Policy

Please check the late policy posted on the course web page and in the PA3 rubric.

7 http://haselgrove.id.au/wikipedia.htm