Google Pagerank

Justin Wyss-Gallifent

July 21, 2021

7.1 Introduction

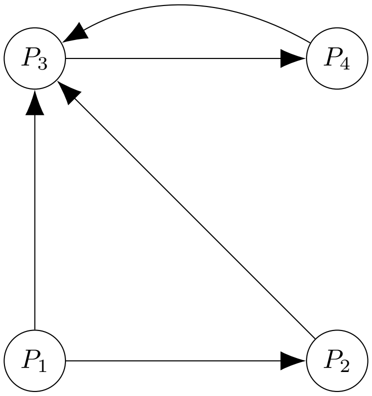
One of the principal requirements that Google has to deal with is ranking web pages. A web page should be ranked higher by some sort of criteria. So how can we go about doing this? Given a web page, the basic idea might be to look at how many pages are linking *to* this page; The more the better. However then we have to appreciate how important *those* pages are, since being linked to by a useless page is not as important as being linked to by an important one, and so the problem goes back and back.

Note the term "Pagerank" comes from the name of Larry Page, one of the founders of Google, not because it's related to web pages.

7.2 Relationship to Markov Chains

Let's examine a very basic internet, see how it connects to Markov Chains, and how the Google Pagerank method works.

Example 7.1. Suppose the internet consists of only four pages, P_1, P_2, P_3, P_4, linked as follows, where, for example, an arrow from P_1 to P_3 indicates a link from P_1 to P_3.



Notice that this looks very much like the diagram of a Markov Chain. If that's the case why don't we just assign probability values to the directions like we did with population diagrams? For example if a web page has two outbound links we could assign each a probability value of 0.5 and so on? If we did this then the steady-state vector would correspond to where a web surfer would end up in the long term, and this seems like a reasonable way to assign value to web pages.

One obvious problem is that a web page may have no outbound links. If that's the case we wouldn't know what to assign for the probabilities.

Another obvious problem is that this method doesn't really act like a web surfer. Web surfers don't just follow links, they also jump to other pages independently of where they were.

The Google Pagerank (GP) algorithm takes the following approach:

(1) We assume that a Random Websurfer (RW) starts at some page.

(2) If the page has outbound links then there is an 85% chance that WR chooses one of those links and those links are equally likely. There is a 15% chance that RW chooses a page at random from all possible pages.

(3) If the page has no outbound links then there is a 100% chance that RW chooses a page at random from all possible pages.

(4) RW will continue to do this forever.

After reading this it becomes fairly clear that this is exactly a Markov Chain. The picture above is not an exact representation of the movement of RW because we need to take into account the 15% chance that RW randomly chooses a page. We could connect every page to every other page in the diagram but that would be a bit silly so instead we just recognize that the connections are there.

Example 7.1 Revisited.

- There is a weight of 0.15/4 connecting P_i to P_j for all i, j.
- There is an additional weight of $0.85(1/n)$ connecting P_i to P_j provided that P_i links to P_j and that P_i links to n pages total.

This is particularly easy to see in terms of two separate matrices:

$$T = \frac{0.15}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} + 0.85\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 \\ 1/2 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$T = \begin{bmatrix} 0.0375 & 0.0375 & 0.0375 & 0.0375 \\ 0.4625 & 0.0375 & 0.0375 & 0.0375 \\ 0.4625 & 0.8875 & 0.0375 & 0.8875 \\ 0.0375 & 0.0375 & 0.8875 & 0.0375 \end{bmatrix}$$

Check your sanity - this should be a transition matrix. Not only that but it's a regular transition matrix because the first part of the sum forces all entries of T^1 to be nonzero. Consequently it obeys our theorem, having an eigenvalue of $\lambda = 1$ and a corresponding probability eigenvector.

The corresponding probability eigenvector is the *ranking vector* :

$$\begin{bmatrix} 0.0375 \\ 0.0534 \\ 0.4711 \\ 0.4379 \end{bmatrix}$$

We therefore rank the pages according to the probability that RW will end up there in the long run:

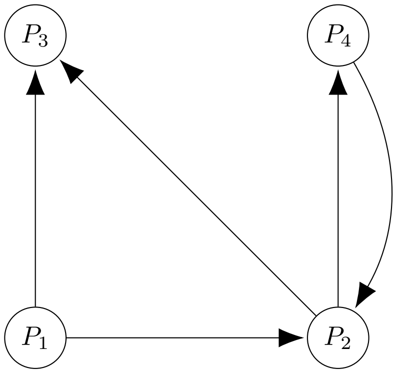
- P_1 has pagerank 0.0375
- P_2 has pagerank 0.0534
- P_3 has pagerank 0.4711
- P_4 has pagerank 0.4379

Think about why this makes sense in the context of the picture.

- The page P_3 is important because lots of pages link to it.
- The page P_4 only has P_3 linking to it but P_3 itself is important, so P_4 is too. Not quite as important though.
- Page P_1 seems least important since no other pages link to it.
- Page P_2 is only slightly more important than P_1 because it does have one page linking to it, that being P_1, but P_1 is not that important.

Here is a second example in which a page has no outbound link.

Example 7.2. Consider the internet:



Here we have:

$$T = \frac{0.15}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} + 0.85\begin{bmatrix} 0 & 0 & 1/4 & 0 \\ 1/2 & 0 & 1/4 & 0 \\ 1/2 & 1/2 & 1/4 & 1 \\ 0 & 1/2 & 1/4 & 0 \end{bmatrix}$$

Notice the column of 1/4. Since page 3 has no outbound links there is a 100% chance that RW will choose a page at random. Since 15% of that is accounted for in the first matrix we simply account for the other 85% in the second one.

We find the corresponding probability eigenvector to be:

$$\begin{bmatrix} 0.1347 \\ 0.1919 \\ 0.4572 \\ 0.2162 \end{bmatrix}$$

- P_1 has pagerank 0.1347
- P_2 has pagerank 0.1919
- P_3 has pagerank 0.4572
- P_4 has pagerank 0.2162

7.3 General Pagerank Matrix

In general then for an internet with n pages we have:

$$T = \frac{0.15}{n}\left[n \times n \text{ matrix of 1s}\right] + 0.85\left[\bar{v}_1 \;\; \bar{v}_n \;\; \ldots \;\; \bar{v}_n\right]$$

where $\bar{v}_i$ is given by:

- If page i has k outbound links then the j^{th} entry of $\bar{v}_i$ equals $1/k$ if page i has an outbound link to page j and 0 otherwise.
- If page i has no outbound links then every entry of $\bar{v}_i$ equals $1/n$.

7.4 Scalability

It's important that we understand that we never need to find the eigenvalues since we know that $\lambda = 1$ is there. This is good because finding the eigenvalues of an $n \times n$ matrix requires finding the roots of a polynomial of degree n and there is no closed formula for the roots of a polynomial of degree 5 or more.

Knowing that $\lambda = 1$ is an eigenvalue then requires us "only" to solve a system of n equations where n is the number of web pages on the internet.

7.5 Matlab

There's nothing particularly new related to Matlab in this chapter but it's worth noting that we can write a function m-file which creates the matrix for us. This is a slightly more sophisticated use of Matlab. The idea is that we'll first create a vector which indicates the links. In the following each row is a link from the first page to the second:

```
>> links = [1,2;1,3;2,3;3,4;4,3]
links =
     1     2
     1     3
     2     3
     3     4
     4     3
```

We also create a scalar containing the total number of pages:

```
>> pagecount = 4;
```

And then the following Matlab function m-file does the job:

```
function m = creategpmatrix(links,pagecount)
  % Usage:
  % links = [1,2;1,3;2,3;3,4;4,3];
  % pagecount = 4;
  % creategpmatrix(links,pagecount)
  part1 = ones(pagecount,pagecount);
  part2 = zeros(pagecount,pagecount);
  linksize = size(links);
  numlinks = linksize(1);
  for i = [1:numlinks]
    part2(links(i,2),links(i,1)) = 1;
  end
  for i = [1:pagecount]
    if (sum(part2(:,i)) > 0)
      part2(:,i) = part2(:,i)/sum(part2(:,i));
    else
      part2(:,i) = ones(pagecount,1)/pagecount;
    end
  end
  m = 0.15/pagecount*part1 + 0.85*part2;
end
```

As follows:

```
>> creategpmatrix(links,pagecount)
ans =
    0.0375    0.0375    0.0375    0.0375
    0.4625    0.0375    0.0375    0.0375
    0.4625    0.8875    0.0375    0.8875
    0.0375    0.0375    0.8875    0.0375
```

If you're curious about what's going on in the function m-file, let's walk through it.

First the command `ones(pagecount,pagecount)` creates a square matrix filled with 1s and the command `zeros(pagecount,pagecount)` creates a square matrix filled with 0s, both of the appropriate size. Next the number of links is calculated.

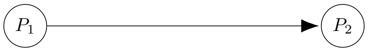
The first `for` loop goes through the links, here `length(links)` is the number of rows in the `links` matrix, hence the number of outbound links (since each link is a row). For each link from P_i to P_j (which are found in `link(i,1)` and `link(i,2)`) we place a 1 in the (j, i) entry of `part2`.

The second `for` loop goes through each column of `part2`. If the sum is nonzero, meaning there are outbound links, then it divides each column by its sum. If the sum is zero, meaning there are no outbound links, then it assigns each entry in the column to be the same and add to 1, which pretends that the page is linked to every other page equally.

Finally we assign the matrix `0.15/pagecount*part1 + 0.85*part2` to return.

7.6 Exercises

Exercise 7.1. Consider this mini-internet:

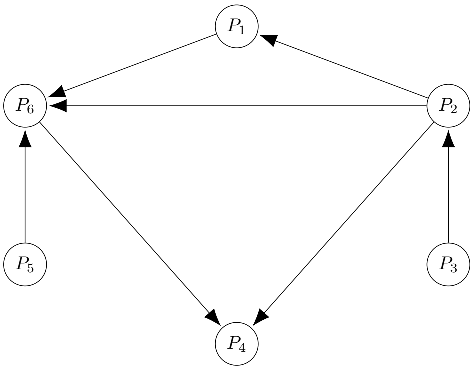


(a) Try to rank the pages in order of importance without doing any calculation.

(b) Find the pagerank of each of the pages.

(c) If there are any disparities between your answer to (a) and (b) explain (if you can) what the cause of this disparity might be.

Exercise 7.2. Consider this mini-internet:



(a) Try to rank the pages in order of importance without doing any calculation.

(b) Find the pagerank of each of the pages.

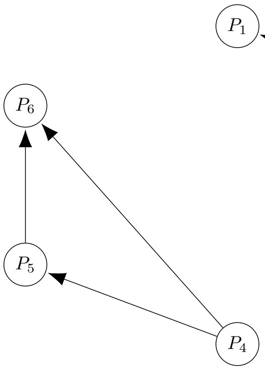
(c) If there are any disparities between your answer to (a) and (b) explain (if you can) what the cause of this disparity might be.

Exercise 7.3. Suppose that the rule that there is a 15% probability that RW jumps to a random page were removed, and instead the full 100% (instead of 85%) from each node were distributed across all outbound links. If there are no outbound links then there is still a 100% probability that RW jumps to a random page. This could lead to the possiblility that T is not regular.

(a) Give an example of an internet with a non-regular T such that there is no $\bar{x}^*$ such that $T^k\bar{x}_0$ converges to $\bar{x}^*$ for all $\bar{x}_0$.

(b) Give an example of an internet with a non-regular T such that there is some $\bar{x}^*$ such that $T^k\bar{x}_0$ converges to $\bar{x}^*$ for all $\bar{x}_0$ but that this causes serious problems with the ranking of the pages. Hint: Can you design an internet where some of the pages will end up with a pagerank of zero?

Exercise 7.4. The Google Pagerank algorithm works even if the internet is disconnected. Consider this example:



Find the pagerank of each of the pages.

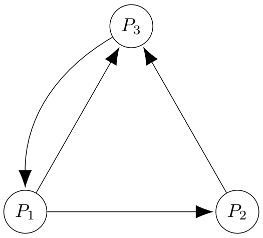
Exercise 7.5. Given the transition matrix T we generally find the pagerank vector by solving the eigenvector equation $(A - I)\bar{x} = \bar{0}$, meaning we find the eigenvector corresponding to the eigenvalue $\lambda = 1$. However it's also possible simply to pick an arbitrary $\bar{x}_0$ and then find $T^k\bar{x}_0$ for successive values of k until successive entries of $T^k\bar{x}_0$ all differ by a predetermined number. Starting with $\bar{x}_0 = [1, 0, 0, 0, 0, 0]^T$ and using the T from the previous problem, find the smallest k so that all entries in $T^k\bar{x}_0$ and $T^{k-1}\bar{x}_0$ are equal up to and including the fourth decimal place.

Exercise 7.6. In an internet with n pages all of which link to one another it makes sense that all of the pages have the same pagerank. Show that this is the case - find the matrix T and show that the vector with all entries equal (and adding to 1) is an eigenvector corresponding to eigenvalue $\lambda = 1$.

Exercise 7.7. A valid question is whether the 85/15 split has an impact not just on the pagerank but on the order of the pages in terms of ranking. For example if we used 60/40 instead would a higher ranked page using 85/15 still be higher ranked using 60/40.

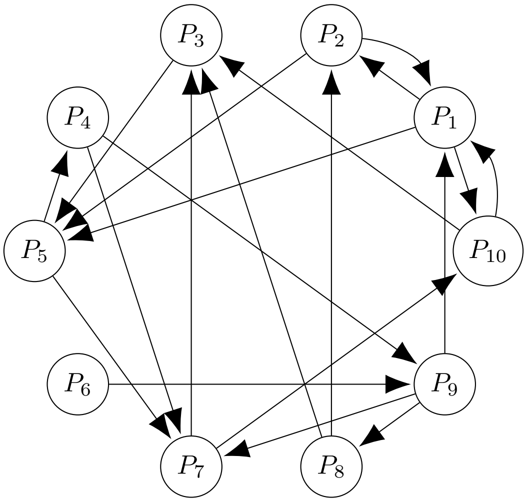
(a) Justify informally why it seems reasonable that the order of the pages in terms of ranking would not be affected.

(b) Test this assumption on the following internet by finding the pageranks using both 85/15 and 60/40 splits.



(c) The above internet can also be analyzed with a 1/0 split. Find the pageranks using this split.

Exercise 7.8. Find the pagerank of the pages in the following internet. You will definitely want to use technology for this!



Exercise 7.9. Explain why having outbound links on your webpage will not affect your Google Pagerank.

Exercise 7.10. Write down the transition matrix for an internet with n pages for which the only links are from page 1 to page 2, page 2 to page 3, ... page $n-1$ to page n.

Exercise 7.11. Why does the Google Pagerank produce more reasonable results than simply assigning a page a ranking in accordance with the number of pages that link to it?