# Page Rank Algorithm

Pawan Singh Faujdar (03CS3006)

15/01/2007

## 1 Page Rank Algorithm:

- PR (A) = PR(B) + PR(C) + PR(D)
- PR(A) = PR(B)/C(B) + PR(C)/C(C) + PR(D)/C(D) —Surfing Pattern

Where C(X) = Links going out from X.
PR(X) = Page Rank of X.

## Page Rank Algorithm of Google:

Page Rank is a numeric value that represents how important a page is on the web, which is calculated by analyzing the link (hyperlink) structure of the web.

One page links (hyperlinks) to another page, it is effectively casting a vote for the other page. Thus more votes that are cast for a page, the more important the page must be and is recursive one. So that the pages linked with the pages with PageRank (higher eigenvector centrality) has a higher Page Rank.

We assume page A has pages T1...Tn which point to it (i.e., are citations). The parameter d is a damping factor which can be set between 0 and 1. We usually set d to 0.85. There are more details about d in the next section. Also C(A) is defined as the number of links going out of page A. The PageRank of a page A is given as follows:

- PR(A) = d( PR(t1)/C(t1) + PR(t2)/C(t2) +........+ PR(tn)/C(tn)) + (1-d)

The PageRanks form a probability distribution over web pages, so the sum of all web pages' PageRanks will be one.

PageRank or PR(A) can be calculated using a simple iterative algorithm, and corresponds to the principal eigenvector of the normalized link matrix of the web.

1. PR(Tn) - Each page has a notion of its own self-importance. That's "PR(T1)" for the first page in the web all the way up to "PR(Tn)" for the last page

2. C(Tn) - Each page spreads its vote out evenly amongst all of it's outgoing links. The count, or number, of outgoing links for page 1 is "C(T1)", "C(Tn)" for page n, and so on for all pages.

3. PR(Tn)/C(Tn) - so if our page (page A) has a backlink from page "n" the share of the vote page A will get is "PR(Tn)/C(Tn)"

4. d(... - All these fractions of votes are added together but, to stop the other pages having too much influence, this total vote is "damped down" by multiplying it by 0.85 (the factor "d")

5. (1 - d) - The (1 - d) bit at the beginning is a bit of probability math magic so the "sum of all web pages' PageRanks will be one": it adds in the bit lost by the d(.... It also means that if a page has no links to it (no backlinks) even then it will still get a small PR of 0.15 (i.e. 1 - 0.85). (Aside: the Google paper says "the sum of all pages" but they mean the "the normalised sum" - otherwise known as "the average" to you and me.

## 2 How is Page Rank calculated:

The PR of each page depends on the PR of the pages pointing to it. But we won't know what PR those pages have until the pages pointing to them have their PR calculated and so on. And when you consider that page links can form circles it seems impossible to do this calculation.

But actually it's not that bad. Remember this bit of the Google paper:

PageRank or PR(A) can be calculated using a simple iterative algorithm, and corresponds to the principal eigenvector of the normalized link matrix of the web.

What that means to us is that we can just go ahead and calculate a page's PR without knowing the final value of the PR of the other pages. That seems strange but, basically, each time we run the calculation we are getting a closer estimate of the final value. So all we need to do is remember the each value we calculate and repeat the calculations lots of times until the numbers stop changing much.

Lets take the simplest example network: two pages, each pointing to the other:

Each page has one outgoing link (the outgoing count is 1, i.e. C(A) = 1 and C(B) = 1).

Guess 1:

We don't know what their PR should be to begin with, so let's take a guess at 1.0 and do some calculations:

d = 0.85

PR(A) = (1 - d) + d(PR(B)/1)

PR(B) = (1 - d) + d(PR(A)/1)

i.e.

PR(A) = 0.15 + 0.85 * 1 = 1

PR(B) = 0.15 + 0.85 * 1 = 1

We can see that the numbers aren't changing at all. So it looks like we started out with a good guess.

Guess 2:

Let's start the guess at 0 instead and re-calculate:

PR(A) = 0.15 + 0.85 * 0 = 0.15

PR(B) = 0.15 + 0.85 * 0.15 = 0.2775

NB. we've already calculated a "next best guess" at PR(A) so we use it here

And again:

PR(A) = 0.15 + 0.85 * 0.2775 = 0.385875

PR(B) = 0.15 + 0.85 * 0.385875 = 0.47799375

And again

PR(A) = 0.15 + 0.85 * 0.47799375 = 0.5562946875

PR(B) = 0.15 + 0.85 * 0.5562946875 = 0.622850484375

and so on. The numbers just keep going up. But will the numbers stop increasing when they get to 1.0. What if a calculation over-shoots and goes above 1.0 ?

Guess 3

Let's start the guess at 40 each and do a few cycles:

PR(A) = 40

PR(B) = 40

First calculation

PR(A) = 0.15 + 0.85 * 40 = 34.25

PR(B) = 0.15 + 0.85 * 0.385875 = 29.1775

And again

PR(A) = 0.15 + 0.85 * 29.1775 = 24.950875

PR(B) = 0.15 + 0.85 * 24.950875 = 21.35824375

Those numbers are heading down alright. It sure looks the numbers will get to 1.0 and stop.

* Principle: it doesn't matter where you start your guess, once the PageRank calculations have settled down, the "normalized probability distribution" (the average PageRank for all pages) will be 1.0.

References:

Link: http://www.iprcom.com/papers/pagerank/