PageRank is a ranking system designed to find the best pages on the web. A webpage is considered good if it is endorsed (i.e. linked to) by other good webpages. The more webpages link to it, and the more authoritative they are, the higher the page's PageRank score.

Note that this ranking is recursive, i.e., the PageRank score of one webpage depends only on the structure of the network and the PageRank scores of other webpages.

If one webpage links to a lot of webpages, each of its endorsements count less than if it had only linked to one webpage. That is, when calculating PageRank, the strength of a website's endorsement gets divided by the number of endorsements it makes.

0.1 Naive formulation of PageRank

In general, PageRank is a way to rank nodes on a graph.

Let r_i be the PageRank of node i, and d_i be its outdegree. Then we can define the PageRank of node j to be

$$r_j = \sum_{i \to j} \frac{r_i}{d_i}$$

That is, each of the neighbors that point to node j contribute to j's PageRank, and the contribution is based on how authoritative the neighbor is (i.e. the neighbor's own PageRank) and how many nodes the neighbor endorses.

If we write one of these equations for each node in the graph, we end up with a system of linear equations, and we can solve it to find the PageRank values of each node in the graph. This system of equations will always have at least one solution[1]. To constrain the scale of the solution, we stipulate that all of the PageRank values must sum to 1 (otherwise there would be an infinite number of solutions, since you could multiply the PageRank vector by any nonzero constant).

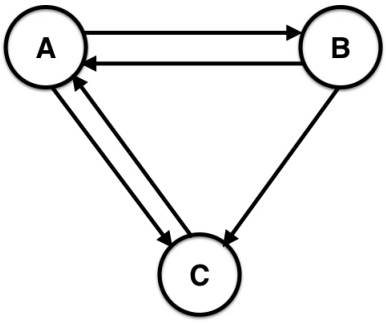

Figure 1: PageRank example

[1]This is because the solution to the PageRank equations can be interpreted as the stationary distribution of a Markov chain, which always exists: http://bit.ly/2eAqGWt

0.1.1 Example

The PageRank equations for the graph in Figure 1 are

$$\begin{aligned} r_A &= r_B/2 + r_C \\ r_B &= r_A/2 \\ r_C &= r_A/2 + r_B/2 \end{aligned}$$

(In addition, we enforce the constraint that $r_A + r_B + r_C = 1$.)

0.2 Matrix representation

We can keep all the PageRank values in a vector

$$\mathbf{r} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix}$$

In which case the PageRank equations become

$$\mathbf{r} = M\mathbf{r}$$

where M is a "weighted adjacency matrix" that contains the structure of the network. Specifically, we have

$$M_{ij} = \begin{cases} \frac{1}{d_j} & \text{if } j \text{ links to } i \\ 0 & \text{otherwise} \end{cases}$$

Note that the columns of M must sum to 1 (so M is a "column stochastic matrix").

0.2.1 Example

We can write the previous example in the form $\mathbf{r} = M\mathbf{r}$ by writing

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

and

$$\mathbf{r} = \begin{bmatrix} r_A \\ r_B \\ r_C \end{bmatrix}$$

0.3 Eigenvalue interpretation

Since $\mathbf{r} = M\mathbf{r}$, we know that assuming $\mathbf{r}$ exists, it must be an eigenvector of the stochastic web matrix M (where the eigenvalue is 1). We show that specifically, it must be the principal eigenvector of M (i.e. the eigenvector corresponding to the eigenvalue of largest magnitude).

Proof: Recall the definition of the L_1 vector norm:

$$||\mathbf{x}||_1 = \sum_{i=1}^{n} |x_i|$$

Using the L_1 vector norm, we can define an induced L_1 matrix norm, as follows:

$$||A||_1 = \max_{\mathbf{x}\neq 0;\ x\in\mathbb{R}^n} \frac{||A\mathbf{x}||_1}{||\mathbf{x}||_1}$$

It follows directly from the definition that $||A\mathbf{x}||_1 \leq ||A||_1||\mathbf{x}||_1$ for any matrix A and vector $\mathbf{x}$. However, this doesn't help much if we can't evaluate $||A||_1$.

Fortunately, there is an alternate, more convenient formula for evaluating the induced L_1 matrix norm:[2]

$$||A||_1 = \max_j \sum_{i=1}^{n} |A_{ij}|$$

That is, the induced L_1 matrix norm of a matrix A is the sum of the entries in the "largest" column.

How does this relate to the eigenvalues? Suppose that $\mathbf{x}$ is an eigenvector of M. We know that $||M\mathbf{x}||_1 \leq ||M||_1||\mathbf{x}||_1$. Since M is a column-stochastic matrix, all of its columns must sum to 1, so the convenient formula for $||M||_1$ gives us 1. Therefore $||M\mathbf{x}||_1 \leq ||\mathbf{x}||_1$.

However, the eigenvalue formula says that $M\mathbf{x} = \lambda\mathbf{x}$, and taking norms on both sides, we get $||M\mathbf{x}||_1 = \lambda||\mathbf{x}||_1$. Therefore, λ must be less than or equal to 1.

0.4 Power iteration

One way to solve for $\mathbf{r}$ is by using **power iteration**. The idea is we start by setting $\mathbf{r} = [1/n, 1/n, \ldots, 1/n]^T$. Then we keep multiplying it by M over and over again until we reach a steady state (i.e. the value of $\mathbf{r}$ doesn't change). This will give us a solution to $\mathbf{r} = M\mathbf{r}$.

Formally, we let $\mathbf{r}^{(0)} = [1/n, 1/n, \ldots, 1/n]^T$, then we iteratively compute $\mathbf{r}^{(t+1)} = M\mathbf{r}^{(t)}$ for each t until $|\mathbf{r}^{(t+1)} - \mathbf{r}^{(t)}|_1 < \epsilon$. (Note that $|\mathbf{x}|_1 = \sum_i |x_i|$ is the L_1 norm.) Then $\mathbf{r}^{(t-1)}$ is our estimate for the PageRank values.

[2] http://pages.cs.wisc.edu/ sifakis/courses/cs412-s13/lecture_notes/CS412_19_Mar_2013.pdf

0.4.1 Why power iteration converges to a principal eigenvector of the matrix M

We claim that the sequence $r^{(0)}, r^{(1)}, r^{(2)}, \ldots$ converges to the principal eigenvector of M (which are the PageRank values).

Proof: Assume that the n-by-n matrix M has n linearly independent eigenvectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$, with corresponding eigenvalues $1 = \lambda_1 > \lambda_2 > \cdots > \lambda_n$. (If this is not true, the proof is harder, and it can be found on Wikipedia.[3])

Then the vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ form a basis of $\mathbb{R}^n$, so we can write

$$\mathbf{r}^{(0)} = c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + \cdots + c_n\mathbf{x}_n$$

Since M is a linear operator, we have

$$\begin{aligned} M\mathbf{r}^{(0)} &= M(c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + \cdots + c_n\mathbf{x}_n) \\ &= c_1(M\mathbf{x}_1) + c_2(M\mathbf{x}_2) + \cdots + c_n(M\mathbf{x}_n) \\ &= c_1(\lambda_1\mathbf{x}_1) + c_2(\lambda_2\mathbf{x}_2) + \cdots + c_n(\lambda_n\mathbf{x}_n) \end{aligned}$$

By the same logic,

$$M^k\mathbf{r}^{(0)} = c_1(\lambda_1^k\mathbf{x}_1) + c_2(\lambda_2^k\mathbf{x}_2) + \cdots + c_n(\lambda_n^k\mathbf{x}_n)$$

Since $\lambda_1 = 1$ and $\lambda_2, \ldots, \lambda_n$ are all less than 1, we get $\mathbf{r}^{(k)} \to c_1\mathbf{x}_1$ as $k \to \infty$. That is, $\mathbf{r}$ approaches the dominant eigenvector of M.

0.5 Markov chain interpretation

One way to interpret PageRank is as follows. Imagine you are a web surfer who spends an infinite amount of time on the internet (which isn't too far from reality). At any time t, you are at a page i, and at time $t+1$, you follow an out-link from i uniformly at random, ending up at one of i's neighbors.

Let $\mathbf{p}(t)$ be the vector whose ith coordinate is the probability that the surfer is at page i at time t. ($\mathbf{p}(t)$ is a probability distribution over pages, and its entries sum to 1.)

Recall that M_{ij} is the probability of moving from node j to node i, given that you are already on node j, and $p_j(t)$ is the probability that you are on node j at time t. Therefore, for each node i, we have

$$p_i(t+1) = M_{i1}p_1(t) + M_{i2}p_2(t) + \cdots + M_{in}p_n(t)$$

[3] https://en.wikipedia.org/wiki/Power_iteration

Which means

$$\mathbf{p}(t+1) = M\mathbf{p}(t)$$

If the random walk ever reaches a state where $\mathbf{p}(t+1) = \mathbf{p}(t)$, then $\mathbf{p}(t)$ is a stationary distribution for this random walk. Recall that the PageRank vector $\mathbf{r} = M\mathbf{r}$. So the PageRank vector $\mathbf{r}$ is a stationary distribution for the random walk!

For graphs that satisfy certain conditions, this stationary distribution is unique, and will eventually be reached regardless of the initial probability distribution at time $t = 0$.

0.6 Final formulation of PageRank

One of the problems with the way we formulated PageRank is that some nodes might not have any out-links. In this case, the random web surfer gets stuck at a "dead end" and can't visit any more pages, ruining our plans. Similarly, the web surfer may get stuck in a "spider trap" of pages where all the links only point to pages inside the spider trap. In that case, the pages in the spider trap eventually absorb all the PageRank, leaving none of the PageRank for other pages.

In order to deal with the spider trap problem, we add an escape route. We say that with probability β (which is usually about 0.8 or 0.9), the web surfer follows an out-link at random, but with probability $1 - \beta$, he jumps to some random webpage. In the case of a dead end, the web surfer jumps to a random webpage 100% of the time.

With this modification, the new PageRank equation becomes

$$r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1 - \beta)\frac{1}{n}$$

where d_i is the outdegree of node i. (This formulation assumes there are no dead ends.)

Similar to our previous matrix M, we can define a new matrix

$$A = \beta M + (1 - \beta)\left[\frac{1}{n}\right]_{n \times n}$$

that reflects the new transition probabilities. Now we just have to solve the equation $\mathbf{r} = A\mathbf{r}$ instead of $\mathbf{r} = M\mathbf{r}$, and we can do that using power iteration.

1 References

"CS 246 Lecture 9: PageRank (2014)." `http://stanford.io/2fDoChT`

"Markov Chains." MIT OpenCourseWare, `http://bit.ly/2eAqGWt`

"CS 412 Lecture Notes: Linear Algebra." http://bit.ly/2fDyZ3d

"Power Iteration." https://en.wikipedia.org/wiki/Power_iteration