CptS 591: Elements of Network Science, Spring 2021
PageRank
Lecture Notes

Assefaw Gebremedhin

March 2, 2021

Note on reference: This lecture notes are based on David Gleich's lecture notes on the same topic from 2013.

There are quite a few ways of deriving the PageRank equation. The Easley and Kleinberg book describes what is often called the "two fixes" approach. We prefer the random surfer derivation, which is what is presented here.

The goal of PageRank is to estimate an importance score for each node on the web-graph. In particular, the setup we have is a directed graph G that represents the web-graph. In comparison to the HITS algorithm, which we will see in next lecture, the graph G is the entire web-graph and not a query-focused subset. Thus, in PageRank, we are going to compute one set of importance scores for all nodes in the network.

PageRank uses the following model for the importance of a page:

a page is important if we are likely to find a web-surfer there.

If we could watch everyone surfing the web, then an ideal PageRank score would be to just average up all of the page-views and use those scores. But, back in 1997 when Google came up with the idea, this was not practical!

Instead, they did what every good computer science student does: create a hypothetical model for how people surf the web, and use that as a proxy.

The PageRank random surfer "browses the web" with the following strategy:

1. click a link on a page, or
2. *do something else.*

Even though it has two activities, this model roughly corresponds to how one typically browses the web! The "do something else" behavior is designed to capture anything that is not a click – these are activities such as doing a web-search, going to a bookmark page, and so on.

But, this model isn't quite specified yet. We need to be more precise about how the model encodes these two behaviors.

For instance, which link does the surfer pick? The basic PageRank model assumes a uniform random choice of which link to click. We'll describe this model. Later PageRank models incorporated measured click-through data based on logs.

And what happens when the surfer "does something else"? Again, in the basic PageRank model, we assume that the surfer jumps to a page uniformly at random over all pages on the web.

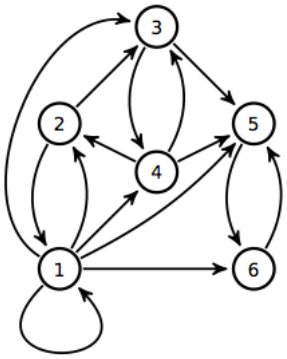
Again, later models used more targeted behaviors in the "do something else" activity to create "topic specific PageRanks" ideas.

Finally, the last detail of the model addresses how do we decide which activity to do? Again, the basic PageRank model makes this choice randomly and fixes a link-following probability α. Consequently, the surfer "does something else" with probability $1 - \alpha$.

This model describes a Markov chain! Recall that Markov chain is a probability model where the value of the random variable at the next time depends only on the value of the random variable at the current time:

$$\text{Probability}(X_{t+1} = i \mid X_t = j) = \text{Probability}(X_{t+1} = i \mid X_t = j, X_{t-1} = k, \ldots).$$

In this case, the random variable will be the page id that the random surfer is currently visiting. If you know that page, then you'll know the probability that the random surfer is on any subsequent page.



Let's see this for the graph displayed above.

If we know that the random surfer is on node 4, then we know $X_t = 4$. So

$\text{Probability}(X_{t+1} = 1 \mid X_t = 4)$	$= (1 - \alpha)/6$	no link but random jump
$\text{Probability}(X_{t+1} = 2 \mid X_t = 4)$	$= \alpha/3 + (1 - \alpha)/6$	link $4 \to 2$ and random jump
$\text{Probability}(X_{t+1} = 3 \mid X_t = 4)$	$= \alpha/3 + (1 - \alpha)/6$	link $4 \to 3$ and random jump
$\text{Probability}(X_{t+1} = 4 \mid X_t = 4)$	$= (1 - \alpha)/6$	no link but random jump
$\text{Probability}(X_{t+1} = 5 \mid X_t = 4)$	$= \alpha/3 + (1 - \alpha)/6$	link $4 \to 5$ and random jump
$\text{Probability}(X_{t+1} = 6 \mid X_t = 4)$	$= (1 - \alpha)/6$	no link but random jump

We can do this for all other nodes as well and build a transition matrix where:

$$P_{i,j} = \text{Probability}(X_{t+1} = j \mid X_t = i).$$

For the behavior of clicking links, the transition matrix is:

$$\mathbf{P} = \begin{bmatrix} 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ \alpha/2 + (1-\alpha)/6 & (1-\alpha)/6 & \alpha/2 + (1-\alpha)/6 & (1-\alpha)/6 & (1-\alpha)/6 & (1-\alpha)/6 \\ (1-\alpha)/6 & (1-\alpha)/6 & (1-\alpha)/6 & \alpha/2 + (1-\alpha)/6 & \alpha/2 + (1-\alpha)/6 & (1-\alpha)/6 \\ (1-\alpha)/6 & \alpha/3 + (1-\alpha)/6 & \alpha/3 + (1-\alpha)/6 & (1-\alpha)/6 & \alpha/3 + (1-\alpha)/6 & (1-\alpha)/6 \\ (1-\alpha)/6 & (1-\alpha)/6 & (1-\alpha)/6 & (1-\alpha)/6 & (1-\alpha)/6 & \alpha + (1-\alpha)/6 \\ (1-\alpha)/6 & (1-\alpha)/6 & (1-\alpha)/6 & (1-\alpha)/6 & \alpha + (1-\alpha)/6 & (1-\alpha)/6 \end{bmatrix}.$$

For the purposes of analysis and simplicity, we often decompose $\mathbf{P}$ into two pieces: transitions due to links and transitions due to "random jumps"

$$\mathbf{P} = \alpha \mathbf{P}_{\text{link}} + (1-\alpha)\mathbf{P}_{\text{jump}} = \alpha \underbrace{\begin{bmatrix} 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 1/3 & 1/3 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{P}_{\text{link}}} + (1-\alpha) \underbrace{\begin{bmatrix} 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \end{bmatrix}}_{\mathbf{P}_{\text{jump}}}.$$

One of the fantastic things about Markov chains is that we have a deep understanding of their properties. For instance, the probability of finding the Markov chain in a particular state after a very long time is given by something called the *stationary distribution*.

The stationary distribution has all sorts of properties. First, for PageRank, it is the unique solution of the eigenvector problem:

$$\mathbf{P}^T\mathbf{x} = \mathbf{x}.$$

If we write out the components of this equation, we find that if

$$x_i = \text{Probability}(X_t = i)$$

then

$$[\mathbf{P}^T\mathbf{x}]_i = \sum_j P_{ji}x_j = \sum_{pages j} \text{Probability}(X_{t+1} = i | X_t = j)\text{Probability}(X_t = j) = \text{Probability}(X_{t+1} = i).$$

Stationarity implies:

$$[\mathbf{P}^T\mathbf{x}] = \text{Probability}(X_{t+1} = i) = x_i = \text{Probability}(X_t = i).$$

So we can view a solution $\mathbf{x}$ as the probability of finding the surfer at a node after a very long time.

Solving the system

We want to solve $\mathbf{P}^T\mathbf{x} = \mathbf{x}$. The Perron-Frobenius theorem tells us that this system has a unique solution. We also know that $\mathbf{x}$ is a probability distribution vector over pages, so $\sum_i x_i = 1, x_i \geq 0$. Let's write the equation out in terms of the link and jump matrices:

$$\mathbf{P}^T\mathbf{x} = (\alpha\mathbf{P}_{\text{link}}^T + (1-\alpha)\mathbf{P}_{\text{jump}}^T)\mathbf{x} = \alpha\mathbf{P}_{\text{link}}^T\mathbf{x} + (1-\alpha)\mathbf{P}_{\text{jump}}^T\mathbf{x}.$$

Now, the jump matrix $\mathbf{P}_{\text{jump}}^T = \frac{1}{n}\mathbf{E}$ where $\mathbf{E}$ is a matrix of all ones and n is the total number of pages. This means that $\mathbf{E}\mathbf{x}$ where $\mathbf{x}$ is a probability vector yields: $\mathbf{E}\mathbf{x} = \frac{1}{n}\mathbf{e}$ where $\mathbf{e}$ is a vector of all ones. This means we can rewrite the equation for $\mathbf{x}$ as:

$$\mathbf{x} = \alpha\mathbf{P}_{\text{link}}^T\mathbf{x} + (1-\alpha)\frac{1}{n}\mathbf{e}.$$

Then the iteration

$$\mathbf{x}^{(k+1)} = \alpha\mathbf{P}_{\text{link}}^T\mathbf{x}^{(k)} + (1-\alpha)\frac{1}{n}\mathbf{e}$$

is guaranteed to converge to the unique solution! This is how to compute a PageRank vector. This will converge for each starting point $\mathbf{x}^{(0)}$, but most people use $\mathbf{x}^{(0)} = \frac{1}{n}\mathbf{e}$.

It is equivalent to the iteration $\mathbf{x}^{(k)} = (\mathbf{P}^T)^k\mathbf{x}^{(0)}$ when $\mathbf{x}^{(k)}$ is always a probability vector too.

The dreaded dangling nodes

Above, we have assumed that each node has at least one outlink. If this isn't the case, then we call such a node a *dangling node*. Determining what to do with dangling nodes is difficult.

The standard fix is to assume that the surfer will execute a random jump from these nodes with probability 1. You can capture this behavior without ever adding any links to your graph using the iteration:

$$\mathbf{x}^{(k+1)} = \alpha \mathbf{P}_{\text{link}}^T \mathbf{x}^{(k)} + (1 - \alpha + \gamma^{(k)}) \frac{1}{n} \mathbf{e}$$

where $\gamma^{(k)}$ is chosen such that $\mathbf{e}^T \mathbf{x}^{(k)} = 1$ always. Usually, people would use the procedure:

```
y = alpha*P'*x;
gamma = 1-sum(y);
x = y + gamma*v
```

where $\mathbf{v} = \frac{1}{n}\mathbf{e}$. Note that $\gamma = 1 - \mathbf{e}^T\mathbf{y} - (1 - \alpha)$ in this formulation.

Another option is adding self-loops to all dangling nodes. This may be appropriate for some applications. But you should think about it.

The two fixes derivation

Easley and Kleinberg present PageRank from the perspective of the thesis:

important pages link to other important pages,

which follows naturally from their treatment of HITS.

This gives rise to the iteration:

$$\mathbf{x}^{(k+1)} \mathbf{P}_{\text{link}}^T \mathbf{x}^{(k)}$$

which we then "fix" into the iteration:

$$\mathbf{x}^{(k+1)} c \tilde{\mathbf{P}}_{\text{link}}^T \mathbf{x}^{(k)} + (1 - c)\mathbf{e}/n.$$

where $\tilde{\mathbf{P}}$ is some fix for dangling nodes.