Page Ranks and Random Walks

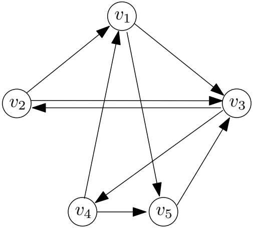
Yufei Tao

Department of Computer Science and Engineering
Chinese University of Hong Kong

We will discuss how to compute **page ranks** on a directed graph, which provide valuable information regarding the "importance" of each vertex in the graph. We will also take the opportunity to have a good look at the theory of **random walks** (a.k.a. **Markov chains**), which generalize the stochastic process underlying page ranks.

Modeling the Internet as a Graph

Let us model WWW as a directed graph $G = (V, E)$.
Each webpage is represented as a node in V. Given two nodes $v_1, v_2 \in V$, E has an edge from v_1 to v_2 if webpage v_1 contains a hyperlink to webpage v_2.



For simplicity, we assume that every node has at least one out-going link.

Random Surfing

Consider the following process that mimics the behavior of a user surfing randomly in WWW:

1. Let u be the webpage that the user is currently at (initially, set u to an arbitrary webpage).
2. With probability α:
   the user clicks on a random hyperlink in u, and set u to the new webpage that opens up.
3. With probability $1 - \alpha$:
   the user sets u to a random webpage in WWW; we will refer to this as a **reset**.
4. Repeat from Step 1.

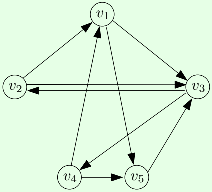
We refer to the above process as **random surfing**.

Page Rank

The **page rank** of a webpage equals the probability that it is the t-th webpage visited by the user when t tends to ∞.

- α is often set to 0.85 in practice.
- You may be wondering how come the page-rank definition says nothing on the **first page** of the user. It turns out that **it does not matter**. The page rank of a page remains the same (when $t \to \infty$) regardless of which is the first page visited.

Example:

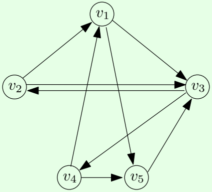


Assume that the first webpage chosen by the user is v_1. Let us analyze the probability that the second page is v_3. For this to happen, one of the following disjoint events must take place:

- A reset occurs, and happens to pick v_3. The probability for this is $0.15 \cdot (1/5) = 0.03$.
- The user follows the link from v_1 to v_3. The probability for this is $0.85 \cdot (1/2) = 0.425$.

Hence, the probability for v_3 to be the second webpage is $0.03 + 0.425 = 0.455$.

Example (cont.):



Let us now analyze the probability that the third webpage is v_4. For this to happen, one of the following disjoint events must occur:

- A reset occurs, and happens to pick v_4. The probability for this is $0.15 \cdot (1/5) = 0.03$.
- v_3 is the second page, and the user follows the link from v_3 to v_4. The probability for this is $0.455 \cdot 0.85 \cdot (1/2) = 0.193$.

Hence, the probability for v_4 to be the third webpage is $0.03 + 0.193 = 0.223$.

Given a vertex $v \in V$, define $p(v, t)$ to be the probability that v is the t-th webpage visited. Then, we have the following recurrence from the above discussion:

$$p(v, t+1) \quad = \quad \frac{1-\alpha}{|V|} + \alpha \cdot \sum_{u \in in(v)} \frac{p(u, t)}{outdeg(u)}$$

where

- $in(v)$ is the set of **in-neighbors** of v (i.e., nodes with links to v).
- $outdeg(v)$ is the **out-degree** of v (i.e., how man links v has).

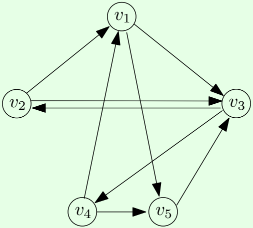
Remark: The above assumes that every webpage has at least one hyperlink. To satisfy this assumption, if a webpage does not contain any hyperlinks, add a hyperlink to itself. The above equation holds after such modifications.

It is guaranteed that, when $t \to \infty$:

$$p(v, t+1) \quad = \quad p(v, t)$$

holds for **all** $v \in V$. The value of $p(v, t)$ at that moment is the of v.

Example:



The page ranks of $v_1, ..., v_5$ are 0.1716, 0.1666, 0.3214, 0.1666, and 0.1737, respectively.

The following algorithm – called the **power method** – computes the page ranks of all vertices:

1. $v \leftarrow$ an arbitrary node in V.
2. set $p(v, 1) \leftarrow 1$, and $p(u, 1) \leftarrow 0$ for all vertices $u \neq v$.
3. $t \leftarrow 1$
4. use the equation of Slide 8 to calculate $p(v, t+1)$ for all $u \in V$
5. **if** $p(u, t) = p(v, t+1)$ for all $u \in V$ **then return**
6. go to Line 4

Remark: In practice, we may terminate the algorithm after t has become large enough (e.g., 100).

Next, we will discuss how page ranks relate to the well-established theory of random walks. In particular, we will see that page ranks form an eigenvector of a matrix that depends only on the WWW graph G and α.

An $n \times n$ matrix M is called a **stochastic matrix** if both the following hold:

- Every value in M is non-negative.
- The values of every row sum up to 1.

Every stochastic matrix M defines a "random walk" process, formally known as a **random walk**.

- Consider that we have a directed graph G_{markov} of n nodes: $v_1, ..., v_n$. For every non-zero entry $M[i,j]$ of M $(1 \leq i,j \leq n)$, G_{markov} has an edge from v_i to v_j (note: j can be i, namely, there can be self-loop edges).
- At the beginning of the random walk, you stand at any vertex of your choice – this is your **first stop**.
- Then, inductively, assuming you are at a node v_i at the t-th stop $(t \geq 1)$, you move to a neighbor v_j with probability $M[i,j]$. The new node you are standing at now is the **$(t+1)$-th stop**.

Remark: The above stochastic process is also called a **Markov chain**.

The random walk on the previous slide is **irreducible** if, for all $1 \le i, j \le n$, there is a path from v_i to v_j in G_{markov}.

The random walk on the previous slide is **aperiodic** if the following statement is true regardless of the first stop: every vertex in G_{markov} has a non-zero probability of being visited at every step $t \ge t_0$ for some finite value t_0.

An $n \times 1$ vector P is a **probability vector** if both the following are true:

- Each component in P is a value between 0 and 1.
- All components of P sum up to 1.

Theorem: Let M be a stochastic matrix describing an irreducible and aperiodic random walk. Let M^T be the transpose of M. Then, there is a unique probability vector P satisfying $P = M^T P$.

The proof is non-trivial and omitted.

Google's random surfing can be regarded as a random walk. Specifically, assume that WWW has n webpages $v_1, ..., v_n$. If you are currently at webpage v_i, then you jump to webpage v_j as the next stop with probability:

- $\frac{1-\alpha}{n}$, if v_i does not have a hyperlink to v_j.
- $\frac{1-\alpha}{n} + \frac{\alpha}{outdeg(v_i)}$, if v_i has $outdeg(v_i)$ hyperlinks, one of which points to v_j.

You can view the above process as a random walk on a graph G_{markov}, where each v_i corresponds to a webpage, and there is a link from every v_i to every v_j (even for $i = j$). Let M be the matrix for this random walk. Then, $M[i, j]$ is set as the probability of jumping from v_i to v_j as discussed above.

Think: Verify by yourself that M describes an irreducible and aperiodic random walk.

As before, let $p(v_i, t)$ $(1 \leq i \leq n)$ be the probability that webpage v_i is the t-th one visited by the random surfer. Let $P(t)$ be an $n \times 1$ vector such that:

$$P(t) \quad = \quad (p(v_1, t), p(v_2, t), ..., p(v_n, t))^T$$

where the superscript T stands for "transpose".

From Slide 8, we know:

$$P(t+1) \quad = \quad M^T \cdot P(t).$$

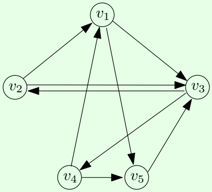
When $P(t+1) = P(t)$, the values in $P(t)$ give the page ranks of the vertices $v_1, ..., v_n$. At this moment, $P(t)$ is the solution of P from the following equation:

$$P = M^T \cdot P.$$

Namely, P (which is a probabilistic vector) is an eigenvector of M of eigenvalue 1. By the theorem in Slide 15, P exists and is unique.

P is commonly referred to as the **stationary probability vector** of the random walk described by M.

Example:



The matrix describing the random walk is:

$$M = \begin{bmatrix} 0.03 & 0.03 & 0.455 & 0.03 & 0.455 \\ 0.455 & 0.03 & 0.455 & 0.03 & 0.03 \\ 0.03 & 0.455 & 0.03 & 0.455 & 0.03 \\ 0.455 & 0.03 & 0.03 & 0.03 & 0.455 \\ 0.03 & 0.03 & 0.88 & 0.03 & 0.03 \end{bmatrix}$$

You can verify that $P = (0.1716, 0.1666, 0.3214, 0.1666, 0.1737)^T$ is an eigenvector of M^T with eigenvalue 1. It is the stationary probability vector of the random walk described by M.

With everything said, we can now re-state the power method in a concise manner:

1. Set $P(1) \leftarrow (1, 0, ..., 0)^T$, and $t \leftarrow 1$.
2. Compute

$$P(t+1) \quad = \quad M^T \cdot P(t).$$

3. $t \leftarrow t + 1$.
4. Repeat from Step 2.

Theorem (the Convergence Theorem): In the power method, $\lim_{t \to \infty} P(t) = P$.

We will prove the theorem in the next few slides.

Proof of the Convergence Theorem

Recall that $P(t) = (p(v_1, t), ..., p(v_n, t))^T$. Define r_i $(1 \leq i \leq n)$ as the page rank of v_i, namely, $P = (r_1, r_2, ..., r_n)^T$.

Define

$$Err(t) = \sum_{i=1}^{n} |p(v_i, t) - r_i|. \quad (1)$$

We will prove that $Err(t) \leq \alpha \cdot Err(t-1)$. This implies that $Err(t) \leq \alpha^t \cdot Err(0)$, which tends to 0 as t goes to infinity. This will prove our claim.

Proof of the Convergence Theorem

By definition of stationary vector, we know that for each $i \in [1, n]$,

$$r_i = \frac{1-\alpha}{n} + \alpha \cdot \sum_{\text{in-neighbor } v_j \text{ of } v_i} \frac{r_j}{outdeg(v_j)}.$$

By how the power method runs, we know:

$$p(v_i, t) = \frac{1-\alpha}{n} + \alpha \cdot \sum_{\text{in-neighbor } v_j \text{ of } v_i} \frac{p(v_j, t-1)}{outdeg(v_j)}.$$

Therefore:

$$|p(v_i, t) - r_i| \le \alpha \cdot \sum_{\text{in-neighbor } v_j \text{ of } v_i} \frac{|p(v_j, t-1) - r_j|}{outdeg(v_j)}. \qquad (2)$$

Proof of the Convergence Theorem

By combining (1) and (2), we have:

$$Err(t) \quad \le \quad \alpha \cdot \sum_{v_i} \sum_{\text{in-neighbor } v_j \text{ of } v_i} \frac{|p(v_j, t-1) - r_j|}{outdeg(v_j)}.$$

Observe that $\frac{|p(v_j,t-1)-r_j|}{outdeg(v_j)}$ is added exactly $outdeg(v_j)$ times, once for every out-neighbor of v_j. Therefore, we conclude:

$$Err(t) \quad \le \quad \alpha \cdot \sum_{v_i} |p(v_i, t-1) - r_i| = \alpha \cdot Err(t-1)$$

completing the proof. $\square$

Remark: our proof suggests that $Err(t) \le \epsilon$ after only $t = O(\log 1/\epsilon)$ rounds.