PageRank

PageRank is a system for ranking web pages that Google's founders Larry Page and Sergey Brin developed at Stanford University. And what it is important to understand is that PageRank is all about links.

The higher the PageRank of a link, the more authoritative it is.

PageRanks is a link analysis algorithm and it assigns a numerical weighting to each element of a hyperlinked set of documents, such as the World Wide Web, with the purpose of "measuring" its relative importance within the set. The algorithm may be applied to any collection of entities with reciprocal quotations and references. The basic idea of PageRank is that the importance of a web page depends on the pages that link to it. For instance, a web page *i* that includes a hyperlink to web page j. If there are a lot of pages also linked to j,then j is considered important on the web. On the other hand, if j only has one in-link, however, this link is from an authoritative web page k (like www.google.com, www.yahoo.com, or www.bing.com), Then also j is important because k can transfer its popularity or authority to j.
A PageRank results from a mathematical algorithm based on the webgraph, created by all World Wide Web pages as nodes and hyperlinks as edges, taking into consideration authority hubs such as cnn.com or mayoclinic.org. The rank value indicates the importance of a particular page. A hyperlink to a page counts as a vote of support. The PageRank of a page is defined recursively and depends on the number and PageRank metric of all pages that link to it ("incoming links"). A page that is linked to by many pages with high PageRank receives a high rank itself.

Numerous academic papers concerning PageRank have been published since Page and Brin's original paper.[5] In practice, the PageRank concept may be vulnerable to manipulation. Research has been conducted into identifying falsely influenced PageRank rankings. The goal is to find an effective means of ignoring links from documents with falsely influenced PageRank.[6]
Other link-based ranking algorithms for Web pages include the HITS algorithm invented by Jon Kleinberg (used by Teoma and now Ask.com), the IBM CLEVER project, the TrustRank algorithm and the Hummingbird algorithm.

1.1 Transition Matrix and Page Rank Value Vector

The web can be represented like a directed graph where nodes represent the web pages and edges form links between them. Typically, when a web page *i* references *j*, a directed edge is added between node i and j in the graph. In PageRank model, each page should transfer evenly its importance to the pages that it links to In general, if a page has k out-links, it will pass on 1 k of its importance to each of the pages that it links to.

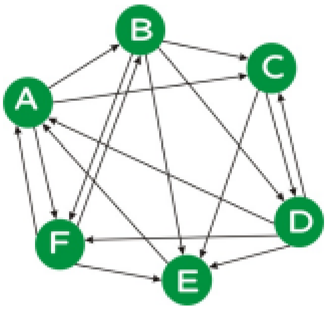


Fig 1: An example web with 6 pages

$$P = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{4} & 1 & \frac{1}{3} \\ \frac{1}{3} & 0 & 0 & 0 & 0 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 \end{bmatrix}$$

Transition Matrix for the graph above

Consider, for example, page A from the graph shown in Fig 1. A has three out-links to pass on 1/3 of its importance to B, C, and F. According to this transition rule, we can define the transition matrix of the graph.

Initially, the algorithm starts with a uniform distribution where the importance of each node is 1/k (here ⅙). This is denoted by π and is called the initial PageRank value vector. As each incoming link increases the PageRank value of a web page, the page rank of each page is updated by adding to the current value to the importance of the incoming links. This is the same as multiplying the matrix P with π.

1.2 Damping Factor

Dangling nodes and disconnected components are quite common on the Internet, considering the large scale of the web. For example, suppose that some pages do not have any out-links (we call them dangling nodes), assuming a random surfer that surfed such pages will inevitably get stuck on these pages. To deal with these problems, a positive constant d between 0 and 1 (typically 0.15) is introduced, called the damping factor.

The significance of the damping factor for the random surfer involves a small but positive percentage of the time, and it will dump the current page and choose a different page arbitrarily from the web and move there. The damping factor d reflects the probability that the surfer quits the current page and moves to a new one.

The damping factor thus modifies the previous transition matrix based on d into P = (1−d)·P +d ·R
where r is given by:

$$R = \frac{1}{N} \cdot \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

1.3 Page Rank formula

Hence the complete page rank formula derived from the previous sections is given as:

$$PR(p_i) = \frac{1-d}{N} + d(\sum_{p_j \text{ links to } p_i} \frac{PR(p_j)}{L(p_j)} + \sum_{p_j \text{ has no out-links}} \frac{PR(p_j)}{N})$$

TrustRank

TrustRank is an algorithm that conducts link analysis to separate valuable web pages from spam and helps search engines rank pages in SERPs (Search Engine Results Pages). It is a semi-automated process that needs some human assistance to function correctly. Search engines have many different algorithms and ranking factors that they use when measuring the quality of web pages. TrustRank is one of them.

Because manual review of the Internet is impractical and very expensive, TrustRank was introduced to help achieve this task faster and cheaper.

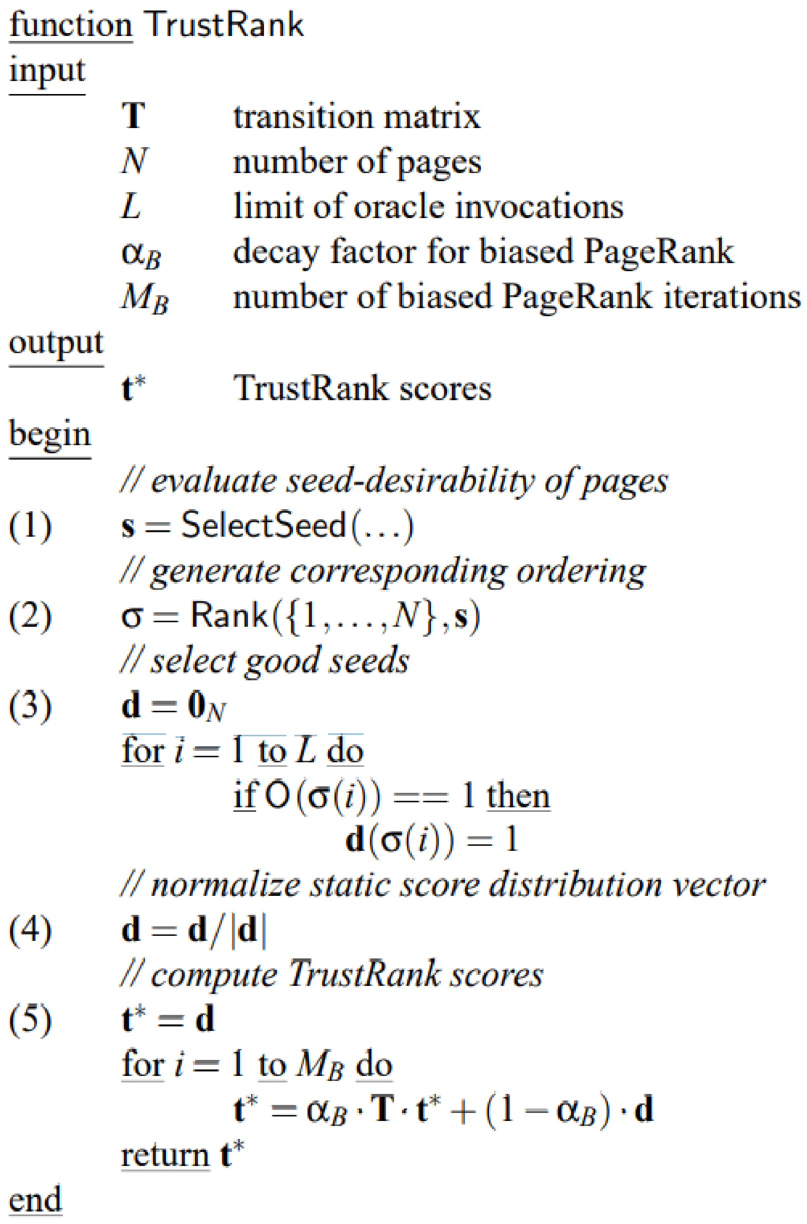
One of the most critical factors that help the web search engine determine a web page’s quality when returning results is backlinks. Search engines consider the number and quality of backlinks when assigning a place to a specific web page in SERPs. Many web spam pages are created only to mislead search engines. These pages, chiefly made for commercial reasons, use various techniques to achieve higher-than-deserved rankings in the search engines' result pages. While human experts can quickly identify spam, search engines are still being improved daily to do it without the help of humans.

One popular method for improving rankings is to increase the perceived importance of a document through complex linking schemes. Google's PageRank and other search ranking algorithms have been subjected to manipulation.

TrustRank seeks to combat spam by filtering the web-based upon reliability. The method calls for selecting a small set of seed pages to be evaluated by an expert. Once the reputable seed pages are manually identified, a crawl extending outward from the seed set seeks out similarly reliable and trustworthy pages. TrustRank's reliability diminishes with increased distance between documents and the seed set.

The logic works oppositely as well, which is called Anti-Trust Rank. The closer a site is to spam resources, the more likely it is to be spam.

2.1 TrustRank Algorithm



```
function TrustRank
input
        $\mathbf{T}$          transition matrix
        $N$          number of pages
        $L$          limit of oracle invocations
        $\alpha_B$         decay factor for biased PageRank
        $M_B$         number of biased PageRank iterations
output
        $\mathbf{t}^*$         TrustRank scores
begin
        // evaluate seed-desirability of pages
(1)     $\mathbf{s} = \text{SelectSeed}(\ldots)$
        // generate corresponding ordering
(2)     $\sigma = \text{Rank}(\{1,\ldots,N\},\mathbf{s})$
        // select good seeds
(3)     $\mathbf{d} = \mathbf{0}_N$
        for $i = 1$ to $L$ do
            if $\text{O}(\sigma(i)) == 1$ then
                $\mathbf{d}(\sigma(i)) = 1$
        // normalize static score distribution vector
(4)     $\mathbf{d} = \mathbf{d}/|\mathbf{d}|$
        // compute TrustRank scores
(5)     $\mathbf{t}^* = \mathbf{d}$
        for $i = 1$ to $M_B$ do
            $\mathbf{t}^* = \alpha_B \cdot \mathbf{T} \cdot \mathbf{t}^* + (1 - \alpha_B) \cdot \mathbf{d}$
        return $\mathbf{t}^*$
end
```

2.2 Assessing Trust

To discover good pages without invoking the oracle function on the entire web, we will rely on an essential empirical observation we call the approximate isolation of the good set: good pages seldom point to bad ones. This notion is reasonably intuitive—bad pages are built to mislead search engines, not to provide useful information. Therefore, people creating good pages have little reason to point to bad pages.

However, the creators of good pages can sometimes be “tricked,” so we find some good-to-bad links on the web. Given a good but unmoderated message board, spammers may include URLs to their spam pages as part of the seemingly innocent messages they post. Consequently, good pages of the message board would link to bad pages.

2.3 Computing Trust

Given a limited budget L of O-invocations, it is straightforward to select at random a seed set S of L pages and call the oracle on its elements. We denote the subsets of good and bad seed pages by S + and S −, respectively. Since the human expert does not check the remaining pages, we assign them a trust score of 1/2 to signal our lack of information. Therefore, we call this scheme the ignorant trust function T0 defined for any p ∈ V as follows:

$$\mathsf{T}_0(p) = \begin{cases} \mathsf{O}(p) & \text{if } p \in \mathcal{S}, \\ 1/2 & \text{otherwise.} \end{cases}$$

As the next step in computing trust scores, we take advantage of the approximate isolation of good pages. We still select the set S of L pages that we invoke the oracle on at random. Then, expecting that good pages point to other good pages only, we assign a score of 1 to all pages that are reachable from a page in S + in M or fewer steps.