On the Google PageRank Algorithm

Ryan Joo Rui An

2024

Abstract

PageRank is an algorithm used by Google Search to rank web pages in their search engine results. It is named after both the term "web page" and co-founder Larry Page. PageRank is a way of measuring the importance of website pages.

§1 Introduction

A *search engine* aims to rank web pages effectively and efficiently. It *sorts* and *ranks* the sites containing a certain keyword, such that the first few sites are the most relevant.

The key assumption made is that the most important (authoritarial) sites receive more links from other sites.

§2 How It Works

Let S be the set containing four sites that contain a certain keyword. Then

$$S = \{s_1, s_2, s_3, s_4\}.$$

It is given that

- s_1 references s_2, s_3 and s_4;
- s_2 references s_4;
- s_3 references s_1 and s_4;
- s_4 references s_1 and s_3.

We can form an *adjacency matrix* $A = (a_{ij})$ defined as

$$a_{ij} = \begin{cases} 1 & \text{if } s_j \text{ references } s_i, \\ 0 & \text{if otherwise.} \end{cases}$$

Then

$$A = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}.$$

Interpreting this for s_1, for instance, it references s_2, s_3 and s_4, so $(2,1)$-, $(3,1)$- and $(4,1)$-entries are 1's.

Next we form the *probability transition matrix* $P = (p_{ij})$ defined as

$$p_{ij} = \frac{a_{ij}}{\sum_{k=1}^{n} a_{kj}}.$$

Basically this transforms A such that the sum of entries in a column is 1.

Hence we have

$$P = \begin{pmatrix} 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & \frac{1}{2} \\ \frac{1}{3} & 1 & \frac{1}{2} & 0 \end{pmatrix}.$$

Suppose a person visits s_3, then his *state vector* is given by

$$\mathbf{x}_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}.$$

His next state vector is given by

$$\mathbf{x}_2 = P\mathbf{x}_1 = \begin{pmatrix} \frac{1}{2} \\ 0 \\ 0 \\ \frac{1}{2} \end{pmatrix}$$

which means that he has equal probabilities of $\frac{1}{2}$ of ending up at s_1 and s_4.

Subsequently, assuming the person randomly refers to other sites, his next state vector is given by

$$\mathbf{x}_3 = P^2\mathbf{x}_1 = \begin{pmatrix} \frac{1}{4} \\ \frac{1}{6} \\ \frac{5}{12} \\ \frac{1}{6} \end{pmatrix}.$$

After multiple clicks, the *resultant state vector* in the run, if the person starts at s_3, is

$$\mathbf{x}_\infty = \begin{pmatrix} 0.267 \\ 0.100 \\ 0.300 \\ 0.333 \end{pmatrix}.$$

This means that s_4 has the highest probability of being visited in the long run, with random clicks.

If the person starts at s_1, we will eventually get the same resultant state vector, regardless of the initial state vector.

Therefore we can rank the sites in descending order of relevance:

$$s_4, s_1, s_3, s_2$$

Since the resultant state vector remains constant in the long-run, we have the following equation which relates the probability transition matrix and resultant state vector:

$$P\mathbf{x}_\infty = \mathbf{x}_\infty \tag{1}$$

Notice that the stochastic matrix P has eigenvalue 1. Hence given P, in order to rank sites, we simply need to compute the eigenvector $\mathbf{x}_\infty$ (also known as equilibrium vector) associated with eigenvalue 1.