Introduction to AI and ML

Google PageRank

Manideep, EE17BTECH11046
Yashas, ES17BTECH11025

March 8, 2019

Google PageRank Algorithm

From the previous presentation:

$$PR(p_i;t+1) = \frac{1-d}{N} + d \sum_{p_j \epsilon M(p_i)} \frac{PR(p_j;t)}{L(p_j)}$$

where $p_1, p_2, ..., p_N$ are the pages under consideration,
$M(p_i)$ is the set of pages that link to p_i,
d is damping factor,(generally,d=0.85),
$L(p_j)$ is the number of outbound links on page p_j,
and N is the total number of pages.

Matrix Form

The PageRank matrix R,is represented as:

$$\mathbf{R} = \begin{bmatrix} PR(p_1) \\ PR(p_1) \\ \vdots \\ PR(p_N) \end{bmatrix}$$

$$\mathbf{R}(t+1) = \begin{bmatrix} (1-d)/N \\ (1-d)/N \\ \vdots \\ (1-d)/N \end{bmatrix} + d \begin{bmatrix} l(p_1,p_1) & l(p_1,p_2) & \cdots & l(p_1,p_N) \\ l(p_2,p_1) & \ddots & & l(p_2,pN) \\ \vdots & & l(p_i,p_j) & \vdots \\ l(p_N,p_1) & l(p_N,p_2) & \cdots & l(p_N,p_N) \end{bmatrix} \mathbf{R}(t)$$

where the adjacency function

$$l(p_i,p_j) = \text{is} \begin{cases} \dfrac{1}{L(p_j)} & \text{,if j links to i} \\ 0 & \text{,otherwise} \end{cases}$$

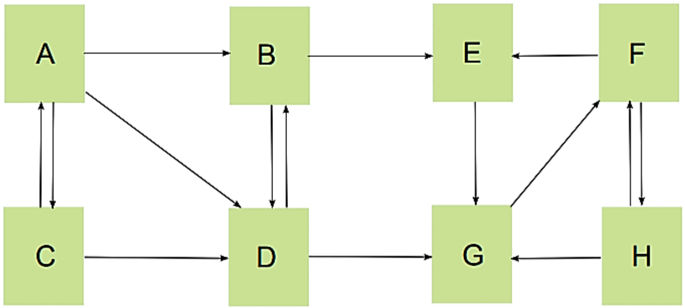
$$\sum_{i=1}^{N} l(p_i, p_j) = 1$$

i.e. the elements of each column sum up to 1

As a result of Markov theory, it can be shown that the PageRank of a page is the probability of arriving at that page after a large number of clicks.

Now t $\rightarrow \infty$ **R(t)**=**R** is the required solution

Web Graph



Methods of calculating PageRank

- Algebraic Method
- Power Method

Algebraic Method

We have:

$$\mathbf{R}(t+1) = \frac{1-d}{N}\mathbf{1} + d\mathcal{M}\mathbf{R}(t)$$

where:

$$\mathcal{M}_{ij} = \text{is} \begin{cases} \dfrac{1}{L(p_j)} & \text{,if j links to i} \\ 0 & \text{,otherwise} \end{cases}$$

$\mathbf{1}$ is the column vector of length N containing only ones.

Algebraic Method

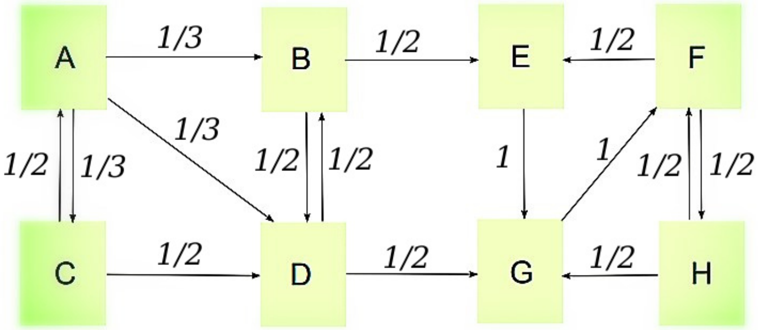
As t $\rightarrow \infty$:

$$\mathbf{R} = d\mathcal{M}\mathbf{R} + \frac{1-d}{N}\mathbf{1}$$

$$\mathbf{R} = (\mathbf{I} - d\mathcal{M})^{-1}\frac{(1-d)}{N}\mathbf{1}$$

We have the constraint that sum of elements of $\mathbf{R}$ is 1.
So by normalizing it,

$$\mathbf{R}_a = \frac{\mathbf{R}}{||\mathbf{R}||}$$


A
B
E
F
C
D
G
H
1/3
1/2
1/2
1/3
1/2
1/3
1/2
1/2
1
1
1/2
1/2
1/2
1/2
1/2

Now we have, $\mathcal{M} =$

$$\begin{bmatrix} 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1/2 \\ 0 & 0 & 0 & 1/2 & 1 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 0 \end{bmatrix}$$

$N = 8$
$d = 0.85$

Substituting the above values in the equation we get:

$$\mathbf{R}_a = \begin{bmatrix} 0.0304 & 0.0536 & 0.0274 & 0.0618 & 0.1621 & 0.2836 & 0.2419 & 0.1393 \end{bmatrix}^T$$

Power Method

$$\mathbf{R} = (d\mathcal{M} + \frac{1-d}{N}\mathbf{E})\mathbf{R}$$

where $\mathbf{E}$ is the matrix whose elements are all 1.(Here,$\mathbf{ER}=\mathbf{1}$)
Here we are making a constraint that sum of all PageRanks is 1

Writing $(d\mathcal{M} + \frac{1-d}{N}\mathbf{E})$ as the operator $\widehat{\mathcal{M}}$:

$$\mathbf{R} = \widehat{\mathcal{M}}\mathbf{R}$$

The power method involves applying the operator $\widehat{\mathcal{M}}$ in succession on the required vector.If the vector under consideration is $x(t)$ which starts from $x(0)$ (normalized vector):

$$x(t+1) = \widehat{\mathcal{M}}x(t)$$

This is done until:

$$|x(t+1) - x(t)| < \epsilon$$

After some finite number of iterations based on the value of ϵ ,we get the matrix **R**.

Taking ϵ as 10^{-8}

$$\mathbf{R} = \begin{bmatrix} .0304 & .0536 & .0274 & .0618 & .1623 & .2834 & .2416 & .1393 \end{bmatrix}^T$$