Extrapolation For Pagerank

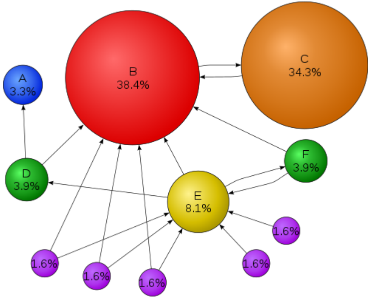
Ankush Dey MDS202108
Ananya S MDS202105
Anjali Pugalia MDS202107
Aniket Santra MDS202106

Chennai Mathematical Institute

May 4, 2022

Introduction

- PageRank (PR) is an algorithm used by Google Search to rank web pages in their search engine results.
- this algorithm outputs a probability distribution.
- It is a link analysis algorithm.
- The goal of PageRank's evaluations is to push not only the most relevant but the most authoritative pages on a given subject to the top of the search engine results pages (SERPs).



Abstract

- The original PageRank algorithm uses the Power Method.
- It computes the principal eigenvector of the Markov matrix representing the hyperlink structure of the Web.
- For Web graphs containing a billion nodes, computing a PageRank vector can take several days.
- Therefore here we are presenting a algorithm called **quadratic extrapolation** that accelerates the pagerank.
- It speeds up PageRank computation by 25–300% on a Web graph of 80 million nodes, with minimal overhead.

Power Method

Power Method

- We can use power method to calculate the principal eigen vector of the matrix.
- The Power Method is the oldest method for computing the principal eigenvector of a matrix.
- Proof for Power Method

$$
\begin{aligned}
\vec{x}^{(0)} &= \vec{u}_1 + \alpha_2\vec{u}_2 + \alpha_3\vec{u}_3 \ldots\ldots + \alpha_{m-1}\vec{u}_{m-1} + \alpha_m\vec{u}_m \\
\vec{x}^{(1)} = A\vec{x}^{(0)} &= \vec{u}_1 + \alpha_2\lambda_2\vec{u}_2 + \ldots\ldots + \lambda_m\alpha_m\vec{u}_m \\
\vec{x}^{(n)} = A^n\vec{x}^{(0)} &= \vec{u}_1 + \alpha_2\lambda_2^n\vec{u}_2 + \ldots\ldots + \lambda_m^n\alpha_m\vec{u}_m
\end{aligned}
$$

Power Method

- algorithm for power method:

function $\vec{x}^n$ = powermethod(){
$\vec{x}^{(0)} = \vec{v}$
k=1
repeat
$\vec{x}^{(k)} = A\vec{x}^{(k-1)}$
$\delta = ||\vec{x}^{(k)} - \vec{x}^{(k-1)}||_1$
k=k+1
until $\delta < \epsilon$
}

Power Method

Operation Count: A single iteration of the Power Method consists of the single matrix-vector multiply $A\vec{x}^{(k)}$. Generally, this is an $O(n^2)$ operation. now for a Markov matrix the largest eigen value is 1. so if the 2nd largest eigen value is is close to 1,then the power method is slow to converge, because n must be large before λ_2 is close to 0, and vice versa.

Quadratic Extrapolation

Quadratic Extrapolation

- We assume that the Markov matrix A has only 3 eigenvectors, and that the iterate $\vec{x}^{(k-3)}$ can be expressed as a linear combination of these 3 eigenvectors.
$$\vec{x}^{(k-3)} = \vec{v}$$
where v is an arbitrary initial vector.
- We then define the successive iterates as
$$\vec{x}^{(k-2)} = A\vec{x}^{(k-3)} \tag{1}$$
$$\vec{x}^{(k-1)} = A\vec{x}^{(k-2)} \tag{2}$$
$$\vec{x}^{(k)} = A\vec{x}^{(k-1)} \tag{3}$$
- Since we assume that A has 3 eigenvectors the characteristic polynomial is given by
$$p_A(\lambda) = \gamma_0 + \gamma_1\lambda + \gamma_2\lambda^2 + \gamma_3\lambda^3$$

Quadratic Extrapolation

- Since A is a Markov matrix $\lambda_1 = 1$

$$p_A(1) = 0 \implies \gamma_0 + \gamma_1 + \gamma_2 + \gamma_3 = 0 \tag{4}$$

- Using Cayley-Hamilton Theorem where let z be any arbitrary vector we get:

$$p_A(A)z = 0 \implies [\gamma_0 I + \gamma_1 A + \gamma_2 A^2 + \gamma_3 A^3]z = 0$$

- Letting $z = \vec{x}^{(k-3)}$,

$$[\gamma_0 I + \gamma_1 A + \gamma_2 A^2 + \gamma_3 A^3]\vec{x}^{(k-3)} = 0$$

- From equations 1,2 and 3 we get

$$\gamma_0 \vec{x}^{(k-3)} + \gamma_1 \vec{x}^{(k-2)} + \gamma_2 \vec{x}^{(k-1)} + \gamma_3 \vec{x}^{(k)} = 0$$

Quadratic Extrapolation

- Using equation 4 we get

$$(-\gamma_1 - \gamma_2 - \gamma_3)\vec{x}^{(k-3)} + \gamma_1\vec{x}^{(k-2)} + \gamma_2\vec{x}^{(k-1)} + \gamma_3\vec{x}^{(k)} = 0$$

- It can be rewritten as:

$$(\vec{x}^{(k-2)} - \vec{x}^{(k-3)})\gamma_1 + (\vec{x}^{(k-1)} - \vec{x}^{(k-3)})\gamma_2 + (\vec{x}^{(k)} - \vec{x}^{(k-3)})\gamma_3 = 0 \quad (5)$$

- Let us define a few new terms as following:

$$\vec{y}^{(k-2)} = \vec{x}^{(k-2)} - \vec{x}^{(k-3)}$$

$$\vec{y}^{(k-1)} = \vec{x}^{(k-1)} - \vec{x}^{(k-3)}$$

$$\vec{y}^{(k)} = \vec{x}^{(k)} - \vec{x}^{(k-3)}$$

Quadratic Extrapolation

- We write equation 5 in the form of a matrix:

$$\begin{pmatrix} \vec{y}^{(k-2)} & \vec{y}^{(k-1)} & \vec{y}^{(k)} \end{pmatrix} \vec{\gamma} = 0$$

- We want to solve for $\vec{\gamma}$. Since we are not interested in the trivial solution $\vec{\gamma} = 0$, we constraint the leading term of the characteristic polynomial:

$$\gamma_3 = 1$$

- Doing this does not affect the zeroes of the polynomial. Therefore equation can be rewritten as:

$$\begin{pmatrix} \vec{y}^{(k-2)} & \vec{y}^{(k-1)} \end{pmatrix} \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = -\vec{y}^{(k)}$$

Quadratic Extrapolation

- As this is an overdetermined system of equations we solve the corresponding least-square problem.

$$\begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = Y^{+} \vec{y}^{(k)}$$

Where Y^{+} is the psuedoinverse of the matrix Y.

- We divide the characteristic polynomial by $\lambda - 1$ to get a new polynomial $q_A(\lambda)$ where:

$$q_A(\lambda) = \beta_0 + \beta_1 \lambda + \beta_2 \lambda^2$$

- Polynomial Division gives the following values:

$$\begin{aligned} \beta_0 &= \gamma_1 + \gamma_2 + \gamma_3 \\ \beta_1 &= \gamma_2 + \gamma_3 \\ \beta_2 &= \gamma_3 \end{aligned}$$

Quadratic Extrapolation

- Again using Cayley-Hamilton let z be any vector then:

$$q_A(A)z = \vec{u}_1$$

- This happens because:

$$\begin{aligned} q_A(\lambda)(\lambda - 1) &= p_A(\lambda) \\ q_A(A)(A - 1) &= p_A(A) \\ q_A(A)(A - 1)z &= p_A(A)z \end{aligned}$$

As $p_A(A)z = 0$

$$\begin{aligned} q_A(A)Az - q_A(A)z &= 0 \\ A(q_A(A)z) &= 1.q_A(A)z \end{aligned}$$

It is of the form

$$A\vec{v} = \lambda\vec{v}$$

Quadratic Extrapolation

- Here $\vec{u}_1$ is the eigenvector of A corresponding to the eigenvalue 1. Let z $= \vec{x}^{(k-2)}$ we get:

$$\vec{u}_1 = q_A(A)\vec{x}^{(k-2)} = [\beta_0 I + \beta_1 A + \beta_2 A^2]\vec{x}^{(k-2)}$$

- Using equations 1,2 and 3 we get

$$\vec{u}_1 = \beta_0\vec{x}^{(k-2)} + \beta_1\vec{x}^{(k-1)} + \beta_2\vec{x}^{(k)} \tag{6}$$

- However, since this solution is based on the assumption that A has only 3 eigenvectors, equation 6 gives only an approximation to $\vec{u}_1$

Quadratic Extrapolation

- Algorithm for Quadratic Extrapolation:

function $\vec{x}^* = QuadraticExtrapolation(\vec{x}^{(k-3)} ......\vec{x}^{(k)})\{$

for j=k-2:k do

$\vec{y}^{(j)} = \vec{x}^{(j)} - \vec{x}^{(k-3)};$

end

$Y = (\vec{y}^{(k-2)} \quad \vec{y}^{(k-1)})$

$\gamma_3 = 1$

$\begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = -Y^{+}\vec{y}^{(k)}$

$\gamma_0 = -(\gamma_1 + \gamma_2 + \gamma_3);$

$\beta_0 = \gamma_1 + \gamma_2 + \gamma_3;$

$\beta_1 = \gamma_2 + \gamma_3;$

$\beta_2 = \gamma_3;$

$\vec{x}^* = \beta_0\vec{x}^{(k-2)} + \beta_1\vec{x}^{(k-1)} + \beta_2\vec{x}^{(k)};$

$\}$

Quadratic Extrapolation

function $\vec{x}^n$ = quadraticpowermethod(){
$\vec{x}^{(0)} = \vec{v}$
k=1
repeat
$\vec{x}^{(k)} = A\vec{x}^{(k-1)}$
$\delta = ||\vec{x}^{(k)} - \vec{x}^{(k-1)}||_1$
periodically,
$\vec{x}^k = QuadraticExtrapolation(\vec{x}^{(k-3)}......\vec{x}^{(k)})$
k=k+1
until
$\delta < \epsilon$
}

Worked Examples

- For Power Method:

$$A=\begin{bmatrix}\frac{1}{2} & 0 & \frac{2}{3}\\ \frac{1}{4} & 1 & 0\\ \frac{1}{4} & 0 & \frac{1}{3}\end{bmatrix} \qquad \vec{x}^{(0)}=\begin{bmatrix}\frac{1}{3}\\ \frac{1}{3}\\ \frac{1}{3}\end{bmatrix}$$

$$\vec{x}^{(1)} = A\vec{x}^{(0)} = \begin{bmatrix}0.385\\ 0.416\\ 0.194\end{bmatrix}$$

$$\vec{x}^{(2)} = A^2\vec{x}^{(0)} = \begin{bmatrix}0.324\\ 0.513\\ 0.162\end{bmatrix}$$

$$\vec{x}^{(3)} = A^3\vec{x}^{(0)} = \begin{bmatrix}0.210\\ 0.594\\ 0.135\end{bmatrix}$$

......

Worked Example

....

$$\vec{x}^{(18)} = A^{18}\vec{x}^{(0)} = \begin{bmatrix} 0.017 \\ 0.973 \\ 0.008 \end{bmatrix}$$

$$\vec{x}^{(19)} = A^{19}\vec{x}^{(0)} = \begin{bmatrix} 0.014 \\ 0.999 \\ 0.007 \end{bmatrix} \approx \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Worked Example

- For Quadratic Extrapolation

$$A=\begin{bmatrix}\frac{1}{2} & 0 & \frac{2}{3}\\ \frac{1}{4} & 1 & 0\\ \frac{1}{4} & 0 & \frac{1}{3}\end{bmatrix} \qquad \vec{x}^{(0)}=\begin{bmatrix}\frac{1}{3}\\ \frac{1}{3}\\ \frac{1}{3}\end{bmatrix}$$

$$\vec{x}^{(1)} = A\vec{x}^{(0)} = \begin{bmatrix}0.385\\ 0.416\\ 0.194\end{bmatrix}$$

$$\vec{x}^{(2)} = A^2\vec{x}^{(0)} = \begin{bmatrix}0.324\\ 0.513\\ 0.162\end{bmatrix}$$

$$\vec{x}^{(3)} = A^3\vec{x}^{(0)} = \begin{bmatrix}0.210\\ 0.594\\ 0.135\end{bmatrix}$$

....

Worked Example

$$\vec{y}^{(1)} = \vec{x}^{(1)} - \vec{x}^{(0)} = \begin{bmatrix} 0.055 \\ 0.083 \\ -0.138 \end{bmatrix}$$

$$\vec{y}^{(2)} = \vec{x}^{(2)} - \vec{x}^{(0)} = \begin{bmatrix} -0.009 \\ 0.180 \\ -0.171 \end{bmatrix}$$

$$\vec{y}^{(3)} = \vec{x}^{(3)} - \vec{x}^{(0)} = \begin{bmatrix} -0.063 \\ 0.261 \\ -0.198 \end{bmatrix}$$

$$\gamma = \begin{bmatrix} -8.881\text{x}10^{-16} \\ -1 \end{bmatrix}$$

$$\beta_0 = -6.661x10^{-16} \approx 0 \qquad \beta_1 = 2.2204x10^{-16} \approx 0 \qquad \beta_2 = 1$$

Worked Exampple

$$\vec{x}^{(3)} = quadraticextrapolation(\vec{x}^{(1)}, \vec{x}^{(2)}, \vec{x}^{(3)}) = \begin{bmatrix} 0.270 \\ 0.594 \\ 0.135 \end{bmatrix}$$

iteration 2

$$\vec{x}^{(4)} = A\vec{x}^{(3)} = \begin{bmatrix} 0.225 \\ 0.662 \\ 0.112 \end{bmatrix}$$

$$\gamma = \begin{bmatrix} -0.475 \\ -0.655 \end{bmatrix}$$

$$\beta = \begin{bmatrix} -0.130 \\ -0.344 \\ 1 \end{bmatrix}$$

$$\vec{x}^{(4)} = quadraticextrapolation(\vec{x}^{(2)}, \vec{x}^{(3)}, \vec{x}^{(4)}) = \begin{bmatrix} 0.275 \\ 0.800 \\ 0.137 \end{bmatrix}$$

.....

iteration 17:

$$= \begin{bmatrix} 0.01 \\ 1.017 \\ 0.006 \end{bmatrix} \approx \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Conclusion

- Web search has become an integral part of modern information access, posing many interesting challenges in developing effective and efficient strategies for ranking search results.
- Although PageRank is a offline computation it has become increasingly desirable to speed up this computation.
- Quadratic Extrapolation is an implementationally simple technique that requires little additional infrastructure to integrate into the standard Power Method.
- In particular, Quadratic Extrapolation works by eliminating the bottleneck for the Power Method, namely the second and third eigenvector components in the current iterate, thus boosting the effectiveness of the simple Power Method itself.