vv214 final project: PageRank Algorithm

516370910214 Sun,Yi

April 19, 2019

1 Introduction

In this project, we are going to rank the importance of umji official websites (umji.sjtu.edu.cn) , so as to help freshmen of the Joint Institute to get familiar with online campus life.

2 Previous Studies

First developed by Larry Page and Sergey Brin in 1996, PageRank Algorithm is an eigenvalue problem[Page 97]. He ranked the web page according to the fundamental assumption that a page is more important if it is pointed to by more pages.

The algorithm soon brought him an idea of a large-scale hypertextual web search engine[Brin and Page 98], which is later named as Google. They introduced the natural model of PageRank Algorithm, including the damping factor α. To be exact, Brin and Page introduced a probability, at any step, that a random surfer would continue his trip from page i to page j. The residual probability is estimated from the frequency that an average surfer uses the bookmark feature in his browser. It is generally assumed that $\alpha \approx 0.85$.

The Google search engine was proved to be relatively efficient, for sake of the large eigengap of its transition matrix[Taher and Sepandar 03]. As a result, the stationary probability of a random visitor arriving at page i could be approximated through a few iteration of the power method.

3 Mathematical Background

3.1 Eigenvalue & Eigenvector

Let M be an $n \times n$ matrix. In linear algebra, we denotes that

- if $x^T M = \lambda x^T$, x is called a left eigenvector.

- if $Mx = \lambda x$, x is called a right eigenvector.
- a left eigenvector is the corresponding right eigenvector of M^T.

3.2 Markov Chain

3.2.1 Markov Property

In statistics and probabilities, Markov property denotes the momeoryless property of a stochastic / random process, i.e., the conditional probability distribution P$[X_{n+1} = x_{n+1}]$ at t=n+1 depends only upon the present state t=n.

$$P[X_{n+1=x_{n+1}}|X_n = x_n, \cdots X_0 = x_0] = P[X_{n+1=x_{n+1}}|X_n = x_n]$$

3.2.2 Discrete Time Markov Chain

A Discrete Time Markov Chain is a sequence of state spaces $S = P[X_i]$ connected by directed edges. The n-th edge of the state space is a transition probability matrix, with $p_{ij} = P[X_i- > X_j]$. By definition, the transition matrix satisfies $\sum_{i=1}^{dimS} [p_{ai}] = 1, \forall a$.

3.2.3 Perron-Frobenius Theorem

It is proved in class that, for a transition matrix $P_{n\times n}$, there exists a largest positive λ_{PF} s.t.

$$\overline{X_n} = (\frac{1}{\lambda_{PF}}P)^n\overline{X_0}$$

converges to an equilibrium state $\overline{X_\infty}$, i.e., $\overline{X_\infty}$ is the eigenvector associated with λ_{PF}.

3.3 Natural PageRank Model

The PageRank Algorithm ranks the importance of web pages. It is assumed that, the more outbound links a page receives, the more important the page is.

3.3.1 Ideal PageRank Model

The original idea of PageRank Model views the web as a directed graph G=(V,E), where each of the N pages $\in V$ is a node, and each hyperlink $\in E$ is an edge.

The adjacency matrix A of the graph stores the out-degree of vertex i, i.e., $a_{ij} = \#Edge$[page i $\rightarrow$ page j]. If $\sum_{j=1}^{n} a_{ij} = 0$, the corresponding page i is a dangling node with no out-links.

We construct a Markov Chain with a state space S=V for page 1 to N, with a row-stachastic vector $x_n = [P_{page1}\ P_{page2}\ \cdots\ P_{pageN}]^T$. A random surfer is supposed to pick a page $\in V$ at the The transition matrix P, which represents the probabilities of a random surfer picking a link at time t, is defined as $p_{ij} = \frac{1}{outdeg(i)}$ for every page j adjacent to i.

The stationary distribution $\overline{x_\infty}$ could be obtained from the following steps

- $x^T = x^T P$
- $\overline{x_\infty} = \frac{x}{||x||}$

3.3.2 Dangling Nodes

At the presence of dangling nodes, the random surfer cannot go to any other pages since there is no outbound links on the page, namely, $\sum_{j=i}^{n} p_{ij} = 0$. The iteration of Markov Chain is stucked in the ideal model.

We solve this problem by introducing a new transition matrix $\overline{P}$=P+D, where D=$\mathbf{d}\mathbf{v}^T$, $d_i = 1$ iff outdeg(i)=0, and $\mathbf{v}$ is a general surfer preference for pages $\in$ V.

3.3.3 Cyclic Paths

Similarly, the surfer might get trapped by a cyclic path in G. $\overline{x_n}$ would never converge to $\overline{x_\infty}$ under this circumstance.

It is artificially defined to avoid this reducibility that, at every time t, the surfer would continue to click the outbound links with a high probability $\alpha \approx 0.85$, while jump to all nodes with the probability $(1-\alpha) \cdot 100\%$.

The ultimate transition matrix for the natural PageRank Model is thus

$$\hat{P} = \alpha \overline{P} + (1-\alpha)\mathbf{e}\mathbf{v}^T$$

where $\mathbf{e}$ is a row vector with all entries equal to 1.

3.3.4 Eigenproblem

Up to now, the PageRank problem is simplified as a eigenproblem $\mathbf{z}^T \hat{P} = \mathbf{z}^T$, where $\mathbf{z}$ is the PageRank vector we are looking for.

$$\hat{P} = \alpha(P + \mathbf{d}\mathbf{v}^T) + (1-\alpha)\mathbf{e}\mathbf{v}^T$$

A common method to solve this eigenproblem is the power method. Previous studies have concluded that for the rank-one modification of αP, $\hat{P}$, few steps of iteration in power method would be enough for a relatively precise $\mathbf{z}$.

Note that $\mathbf{z}^T\mathbf{e} = \sum_{i=1}^{N} z_i = 1$, this eigenproblem can be rewritten into a linear system

$$S\mathbf{z} = (1-\alpha)\mathbf{v}$$

where S=I-$\alpha P^T - \alpha \mathbf{v}\mathbf{d}^T$.

We would further discuss the handling of the nonsparsity of S so as to reduce the complexity of PageRank Algorithm.

4 problem

5 Conclusion

6 Reference

Page, Larry, "PageRank: Bringing Order to the Web". Archived from the original on May 6, 2002. Retrieved 2016-09-11., Stanford Digital Library Project, talk. August 18, 1997 (archived 2002)

Brin, S.; Page, L. (1998). "The anatomy of a large-scale hypertextual Web search engine". Computer Networks and ISDN Systems. 30 (1–7): 107–117. CiteSeerX 10.1.1.115.5930. doi:10.1016/S0169-7552(98)00110-X. ISSN 0169-7552. Archived from the original on 2015-09-27.

Taher Haveliwala; Sepandar Kamvar (March 2003). "The Second Eigenvalue of the Google Matrix". Stanford University Technical Report: 7056. arXiv:math/0307056. Bibcode:2003math......7056N. Archived from the original on 2008-12-17.