IOSR Journal of Engineering (IOSRJEN)
e-ISSN: 2250-3021, **p**-ISSN: 2278-8719
Vol. 3, Issue 5 (May. 2013), ||V4 || PP 26-31

Page Ranking: Summary

Sonia Gupta
Computer Science Department Iftm University Moradabad (India)

Abstract: World Wide Web is the source of enormous information and we have many numbers of means to understand it. There are number of algorithms that are developed to extract these data from the pool of Web. There are mainly two algorithms which are used for webpage ranking. They are HITS and pageranking. HITS focuses on mutual reinforcement between authority and hub webpages, while PageRank focuses on hyperlink weight normalization and web surfing based on random walk models. In the earlier pagerank algorithm for enhancing the quality of the search-results a single page rank vector was used. For more accuracy of the results then further onwards the set of vectors was used. The users interest, knowledge creates the importance of the webpages. When the webpages are rated objectically or mechanically then it is termed as pageranking. We can construct page ranking on the large subgraph with the limited memory with them. Web can be considered as graph where pages are nodes and edges can be termed as hyperlinks. Here we discuss some methods of converging pageranking based on ordering of pages. Here pageranking is compared to idealized random surfing of net.

Keywords: *Page Rank, Markov chains, HITS, decomposition, aggregation/disaggregation.*

I. INTRODUCTION

The network structure provide enormous source of information regarding our queries in such a way that we can understand it quiet easily. It is very vast and collection of different pages. The variation of pages is more worse than raw scale of the data. Number of pages have been drastically increased and doubled itself to 800 million pages. For such a work we need to apply some of the algorithms to extract the data from such a pool of knowledge. WWW is a summation of lot of complexity and its phenomenal rate continuously expands. There are certain types of queries:

- ❖ Specific queries. E.g., "Does Netscape support the JDK 1.1 code-signing API?"
- ❖ Broad-topic queries. E.g., "Find information about the Java programming language."
- ❖ Similar-page queries. E.g., "Find pages `similar' to java.sun.com."

Webpages provide free of quality control or publishing cost. There are many ways through which webpages can be differentiated from each other. To measure out the importance of the webpages Pagerank is used, it is method for computing the every webpage based on the graph of the web. There are certain applications of the webpages such as search, browsing and traffic estimation. There are number of problems that are encountered by pageranking. They are as follows:

- No control on content.
- Huge size of web.
- Enormous increase in the size.
- Webpages are not in a proper structure.

Our aim is to discover information sources and to extract the relevant information from them either entirely automatically, or with very minimal human intervention. There are different ways through which data could be extracted from the WWW they are as follows:

- **Finding relevant information**

To find the relevant information people usually browse or use the search specific methods. To search the information from the network people have to insert the query regarding their search. There are certain problems regarding the search made. They are low precision and low recall.

- **Creating new knowledge out of information available on the Web**

This can be considered as sub-problem of the above two problems. Recent research focuses on utilizing the Web as a knowledge base for decision making.

- **Personalization of the information**

This is the problem which deals with the type and presentation of the information.

- **Learning about consumers and individual users**

This is the problem which deals with the problem of presentation. Sub-problem which lies under this problem is mass customizing the information to intend consumers or even to personalize it to individual users, problem related to marketing etc.

Pagerank evaluates the “prestige score” of a page as roughly proportional to the sum of prestige scores of pages citing it using hyperlinks. Pagerank is preferred widely then HITS as as the query-time cost of incorporating the precomputed PageRank importance score for a page is low. Generation of pagerank needs the entire web graph whereas in HITS small subset of the graph is used. The PageRank algorithm precomputes a rank vector that provides a-priori “importance” estimates for all of the pages on the Web. The idea of biasing the PageRank computation was for the purpose of personalization, but was never fully explored. This biasing process involves introducing artificial links into the Web graph during the offline rank computation. PageRank is a nice solution to evaluate the importance of the nodes of a graph based on the resolution of a fixed point problem associated to the random surfer model and to the Markov chain associated to the random walk. It is of natural interest to search for the maximum or minimum PageRank that a node (e.g., a website) can have depending on the presence or absence of some of the edges (e.g., hyperlinks) in the graph. Another motivation is that of estimating the PageRank of a node in the presence of *missing information* on the graph structure.

II. AN S-VALUED MARKOV PROCESS

is an infinite sequence of random variables $X_k = X_0, X_1, . . . \epsilon S$ if S is finite and the probability function

P satisfies: $P(X_{k+1} = b|X_0 = a_0, . . . , X_k = a_k) = P(X_{k+1} = b|X_k = a_k)$ is the same for all $k > 0$.

Its **transition function** is $\omega (a, b) = P(X_{k+1} = b|X_k = a)$.

Its **initial distribution** is $\sigma (a) = P(X_0 = a)$.

Usually a Markov chain is defined for a discrete set of times (i.e., a discrete-time Markov chain) although some authors use the same terminology where "time" can take continuous value.

1.1 Convergence of Markov processes

we review the conditions under which $\lim_{k\to\infty} P(X_k = a)$

converges. There s a property in most of the Markov process known as ergodicity.

Period of state Let $\{X_k\}$ be an S-valued Markov process.

The period of a state $a \in S$ is the largest d satisfying: $(\forall k, n \in N)$

$P(X_{n+k} = a|X_k = a) => 0)$ d divides n

If d = 1, then the state a is aperiodic.

Closed subset Let $\{X_k\}$ be an S-valued Markov process.

The subset $A \subseteq S$ is called closed subset if $\forall a \in A, \forall b$ Ï $A) \omega (a, b) = 0$.

Irreducible closed subset Let $\{X_k\}$ be an S-valued Markov

process. The subset $A \subseteq S$ is called irreducible closed subset iff A is a closed subset, and no proper subset of A is closed subset.

Irreducible Markov process Let $\{X_k\}$ be an S-valued Markov process. The Markov process is called irreducible Markov process iff S is a irreducible closed subset.

Ergodic Markov process An ergodic Markov process is

a Markov process $\{X_k\}$ that is both

- **irreducible:** every state is reachable from every other state.
- **aperiodic:** the greatest common divisor of the states periods is 1.

1.2 The Markov model of the Web

Consider the hyperlink structure of the Web as a directed graph. The nodes of this digraph represent webpages and the directed arcs represent hyperlinks

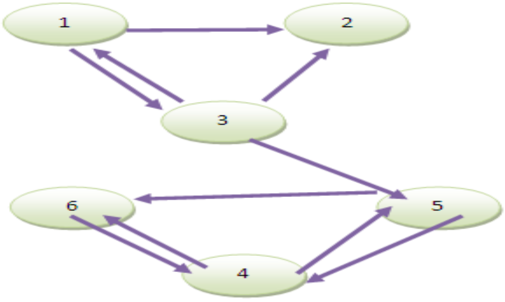


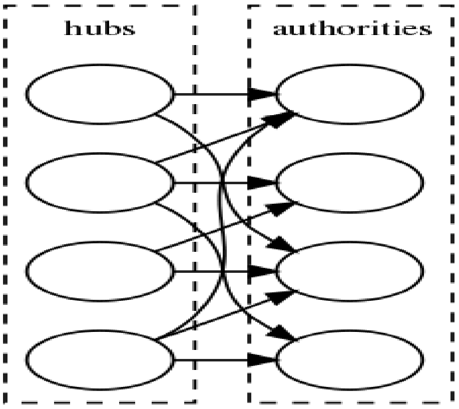
Fig: Directed graph sowing 6 webpages

The Markov model represents this graph with a square matrix P whose element P_{ij} is the probability of moving from state i (page i) to state j (page j) in one time step.

III. HITS ALGORITHM

This algorithm make use of the link structure of the web in order to discover and rank pages relevant for a particular topic. **HITS *(hyperlink-induced topic search)*** is now part of the **Ask** search engine. This was developed by Jon Kleinberg. HITS is applied on a subgraph after a search is done on the complete graph. Uses hubs and authorities to

define a recursive relationship between web pages. Uses hubs and authorities to define a recursive relationship between web pages. A hub is a page that links to many authorities.



Jon Kleinberg's algorithm called **HITS** identifies good authorities and hubs for a topic by assigning two numbers to a page: an authority and a hub weight. These weights are defined recursively.

ALGORITHM

The first step is to retrieve the most relevant pages to the search query. This set is called the *root set* and can be obtained by taking the top n pages returned by a text-based search algorithm. A *base set* is generated by augmenting the root set with all the web pages that are linked from it and some of the pages that link to it.

The algorithm performs a series of iterations, each consisting of two basic steps:

***Authority Update**:* Update each node's *Authority score* to be equal to the sum of the *Hub Scores* of each node that points to it. That is, a node is given a high authority score by being linked to by pages that are recognized as Hubs for information.

***Hub Update**:* Update each node's *Hub Score* to be equal to the sum of the *Authority Scores* of each node that it points to. That is, a node is given a high hub score by linking to nodes that are considered to be authorities on the subject.

The Hub score and Authority score for a node is calculated with the following algorithm:

1. Start with each node having a hub score and authority score of 1.
2. Run the Authority Update Rule
3. Run the Hub Update Rule
4. Normalize the values by dividing each Hub score by square root of the sum of the squares of all Hub scores, and dividing each Authority score by square root of the sum of the squares of all Authority scores.
5. Repeat from the second step as necessary.

1.3 Authority Update Rule

$\forall$p, we update auth(p) to be the summation: $\sum_{i=1}^{n} hub(i)$where n is the total number of pages connected to p and i is a page connected to p. That is, the Authority score of a page is the sum of all the Hub scores of pages that point to it.

1.4 Hub Update Rule

$\forall$p, we update hub(p) to be the summation:

$\sum_{i=1}^{n} auth(i)$where n is the total number of pages p connects to and i is a page which p connects to. Thus a page's Hub score is the sum of the Authority scores of all its linking pages.

IV. DECOMPOSITION

Decomposition of a matrix as a sum or linear combination of outer product matrices underlies both the bilinear methods and fundamental concept of matrix rank. The PageRank vector is very long and it is good idea try to divide it on several components and find each component separately and, after that, find the whole PageRank vector. The matrix P is represented in block structure for the purpose.

$\begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{n1} & \cdots & P_{nn} \end{bmatrix} = P$where N < n. The PageRank vector is

□ = (□$_1$, □$_2$, . . . , □$_N$),

where □ $_I$ is row vector with dim(□ $_I$) = n_I and

$$\sum_{I=1}^{n} n_I = n$$

4.1 Block-diagonal case

Lets consider the case when the matrix P is block-diagonal

$$P = \begin{bmatrix} P_1 & 0 & \cdots & 0 \\ 0 & P_2 & & 0 \\ & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & P_N \end{bmatrix}$$

For block I define the perturbed matrix

$G_I = cP_I + (1 - c)1/n_I E$,

and let vector □ $_I$ be the PageRank of □ $_I$ such that

□ $_I$ = □ $_I G_I$

□ $_I$e = 1

4.2 2 × 2 case

Let us consider the case

$$P = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix}$$

□ = (□$_1$, □ $_2$)

where P_{11} and P_{22} are square. The equation can be rewritten as

□ (I − P) = 0.

The partition of P was considered in. Assume that P is irreducible. Hence, I−P is an singular and irreducible, the non-trivial leading principal submatrix. I − P_{11} is non-singular.

V. AGGREGATION/DISAGGREGATION

When power method is used for finding PageRank different components of the PageRank vector can converges with different speed. And while the appropriate accuracy is achieved for some components we have to continue computation to reach a good accuracy for components converging slowly. Aggregation/disaggregation methods are based on the idea. Partitioning of a Google matrix is used

$$G = \begin{pmatrix} G_{11} & G_{12} & \cdots & G_{1N} \\ G_{21} & G_{22} & & G_{2N} \\ & \vdots & \ddots & \vdots \\ G_{N1} & G_{N2} & \cdots & G_{NN} \end{pmatrix}$$

Aggregation/disaggregation can be classified into the following three types:

1.5 Exact Aggregation/Disaggregation

Aggregation/disaggregation method is an improved form of pagerank algorithm. The aggregation algorithm computes the components of □ as stationary distributions of smaller matrices.

Partition the irreducible stochastic matrix

$P = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix}, \pi = \begin{bmatrix} \pi_1 \\ \pi_2 \end{bmatrix}$

so that P_{11} and P_{22} are square.

ALGORITHM

1. Determine the stationary distribution σ of $s \equiv P_{22} + P_{21}(I - P_{11})^{-1}P_{12}$.
2. Determine the stationary distribution α of

$$A \equiv \begin{pmatrix} P_{11} & P_{12}1 \\ \sigma^T P_{21} & \sigma^T P_{22}1 \end{pmatrix}$$

3. Partition $\propto = \begin{pmatrix} \pi_1 \\ \rho \end{pmatrix}$, and set $\pi = \begin{pmatrix} \pi_1 \\ \rho\sigma \end{pmatrix}$

The first two steps of Algorithm can be interpreted as an aggregation because they take place on 'aggregated' matrices of smaller size, while the third step represents a disaggregation that produces a long vector□ of original size.

1.6 Approximate Aggregation/Disaggregation

The approximate aggregation/disaggregation algorithm does away with the time consuming computations involving the stochastic complement S. The approximate aggregation matrix is

$$\bar{A} \equiv \begin{pmatrix} P_{11} & P_{12}1 \\ \bar{\sigma}^T P_{21} & \bar{\sigma}^T P_{22}1 \end{pmatrix}$$

and it differs from the exact matrix A only in the last row.

ALGORITHM

1. Select a vector $\bar{\sigma}$with $\bar{\sigma} > 0$ and $\bar{\bar{\sigma}}^T 1 = 1$.
2. Determine the stationary distribution $\bar{\propto}$of

$$\bar{A} \equiv \begin{pmatrix} P_{11} & P_{12}1 \\ \bar{\sigma}^T P_{21} & \bar{\sigma}^T P_{22}1 \end{pmatrix}$$

3. Partition $\bar{\propto} = \begin{pmatrix} \omega_1 \\ \bar{\rho} \end{pmatrix}$, and set$\omega \equiv \begin{pmatrix} \omega 1 \\ \bar{\rho} \quad \bar{\sigma} \end{pmatrix}$
4. Multiply $\bar{\pi}^T \equiv \omega^T P$

1.7 Iterative Aggregation/Disaggregation (IAD)

The iterative aggregation/disaggregation (IAD) method improves the PageRank algorithm. For simplicity we view the IAD method as an alternative to the power method rather than an updating algorithm.

ALGORITHM

1. Select a vector $\pi^{(0)} = (\pi_1{}^{(0)} \quad \pi_2{}^{(0)})$ with $\pi_2{}^{(0)} > 0$

2. Do k = 1, 2 . . .

(a) Normalize

$$\sigma^{(k)} \equiv {\pi_2{}^{(k-1)}}\Big/{[\pi_2{}^{(k-1)}]^T 1}$$

(b) Determine the stationary distribution $\propto^{(k)}$ of $A^{(k)} \equiv \begin{pmatrix} P_{11} & P_{12}1 \\ [\sigma^{(k)}]P_{21} & [\sigma^{(k)}]^T P_{22} \end{pmatrix}$

(c) Partition

$\propto^{(k)} = \begin{pmatrix} \omega_1{}^{(k)} \\ p_k \end{pmatrix}$, and set $\omega^{(k)} \equiv \begin{pmatrix} \omega_1{}^{(k)} \\ p_k \sigma^{(k)} \end{pmatrix}$

(d) Multiply $[\pi^{(k)}]^T \equiv [\omega^{(k)}]^T P$

VI. CONCLUSION AND FUTURE WORK

In the new PageRank algorithm, pages have the right PageRank values and the iteration process always converges to a fixed point. We also described efficient implementation issues of our algorithm. The implementation of our algorithm does not require a large amount of spatial and computational overhead. The PageRank algorithm, is related to the concept of finding a canonical vertex ordering. Since a PageRank vector can be obtained by applying the power method, which simply computes an iterated dot product, vertices contained in the same block yield equal PageRank values. The quotient matrix can be constructed more efficiently. The current versions also use dense matrices, but sparse matrices must be used to process large graphs, e.g., web graphs. Adding sparse matrix support also motivates supporting personalization vectors, where provided a probability vector, **v**, the PageRank matrix, **S**, yielded by applying the PageRank perturbation.

Artificial Intelligence can offer tremendous help. Instead of providing a PCS guarantee, Bayesian approaches attempt to allocate a finite data budget to maximize the posterior PCS of the selected system. we will investigate ranking formulas from other IR models such as the Set-based Model to extract

new terminals. We will utilize the structural information within documents to compare our approach to others, such as , for Web search. We also will investigate the influence of the mutation factor, tree depth, and population length on the discovery of good ranking functions, considering the use of meaningful terminals and statistical information.

REFRENCES

[1]. S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. Proc. of 7th WWW Conferece, 1998.

[2]. J. M. Kleinberg. Authoritative sources in a hyperlinked environment. J. ACM, 48:604{632, 1999}.

[3]. J. Kleinberg, Authoritative sources in a hyperlinked environment, In Proceedings of the Ninth ACM-SIAM Symposium on Discrete Algorithms (1998) 668–677.

[4]. M.E.J. Newman, The structure and function of complex network, SIAM Review, 45:2 (2003) 167–256.

[5]. A. N. Langville and C. D. Meyer, Deeper inside PageRank, Internet Mathematics, 1:3 (2005) 335–380.

[6]. Nancy Blachman, Eric Fredricksen, and Fritz Schneider. How to Do Everything with Google. McGraw-Hill, 2003.

[7]. The PageRank citation ranking: Bringing order to the web. L. Page, S. Brin, R. Motwani and T. Winograd Technical Report, Stanford University, 1998.

[8]. Krishna Bharat and George A. Mihaila. When experts agree: using non-affiliated experts to rank popular topics. ACM Transactions on Information Systems, 20(1):47–58, 2002.

[9]. Abraham Berman and Robert J. Plemmons. Nonnegative Matrices in the Mathematical Sciences. Academic Press, Inc., 1979.

[10]. Monica Bianchini, Marco Gori, and Franco Scarselli. Inside PageRank. ACM Transactions on Internet Technology, 5(1), 2005. To appear.

[11]. Paolo Boldi and Sebastiano Vigna. The WebGraph framework II: Codes for the World Wide Web. Technical Report 294-03, Universita di Milano, Dipartimento di Scienze dell' Informazione Engineering, 2003.

[12]. Zheng Chen, Jidong Wang, Liu Wenyin, and Wei-Ying Ma. A unified framework for web link analysis. In Proceedings of Web Information Systems Engineering, page 63, New York, 2002. ACM Press.

[13]. Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine.Computer Networks and ISDN Systems, 33:107–117, 1998.

[14]. Grace E. Cho and Carl D. Meyer. Markov chain sensitivity measured by mean first passage times. Linear Algebra and its Applications, 313:21–28, 2000.

[15]. David Cohn and Huan Chang. Learning to probabilistically identify authoritative documents. In Proceedings of the 17th International Conference on Machine Learning, pages 167–174, Stanford, CA, 2000.

[16]. C.F.Ipsen and S.Kirklad. Convergence analysis of the Langville-Meyer PageRank algorithm.

[17]. K.Avrachenkov and N.Litvak. Decomposition of the Google PageRank and Optimal Linking Strategy. Inria Sophia Antipolis, University of Twente, 2004.

[18]. A.Berman and R.J.Plemmons. Nonnegative Matrices in the Mathematical Sciences. SIAM Classics In Applied Mathematics, SIAM, Philadelphia, 1994.

[19]. S. Brin and L. Page, The anatomy of a large-scale hypertextual web search engine, Comput. Networks and ISDN Systems, 30 (1998), pp. 107–17.

[20]. A. Langville and C. Meyer, Updating PageRank using the group inverse and stochastic complementation, (2002). CRSC-TR02-32 at www.ncsu.edu/crsc/reports/reports02.html.

[21]. T. H. Haveliwala: Efficient Computation of PageRank, unpublished manuscript, Stanford University (1999).

[22]. A. Y. Ng, A. X. Zheng, and M. I. Jordan: Stable Algorithms for Link Analysis, Proceedings of the 24th ACM SIGIR Conference (2001), 258-266

[23]. B. Pˆossas, N. Ziviani, J. Wagner Meira, and B. Ribeiro-Neto. Set-based vector model: An efficient approach for correlation-based ranking. ACM TOIS, 23(4):397–429, 2005.

[24]. 24. W. Fan, M. D. Gordon, and P. Pathak. Genetic programming-based discovery of ranking functions for effective web search. Journal of Manag. Inf. Syst., 21(4):37–56, 2005

.