WEB PAGE RANKING SOLUTION THROUGH sNorm (p) ALGORITHM IMPLEMENTATION

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
In
Computer Science & Engineering

Thapar University, Patiala

By:
Munish Kumar
(80632015)

Under the supervision of:
Mr. Ravinder Kumar
Lecturer, Computer Science and Engineering Department,
Thapar University, Patiala.

MAY 2008

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

WEB PAGE RANKING SOLUTION THROUGH sNorm (p) ALGORITHM IMPLEMENTATION

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
In
Computer Science & Engineering

Thapar University, Patiala

By:
Munish Kumar
(80632015)

Under the supervision of:
Mr. Ravinder Kumar
Lecturer, Computer Science and Engineering Department,
Thapar University, Patiala.

MAY 2008

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

Certificate

I hereby certify that the work which is being presented in the thesis entitled, **"Web Page Ranking Solution through sNorm(p) Algorithm Implementation"**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in Computer Science & Engineering submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Mr.Ravinder Kumar and refers other researcher's works which are duly listed in the reference section.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

(**Munish Kumar)**

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Mr. Ravinder Kumar
Computer Science and Engineering Department
Thapar University
Patiala

Countersigned by

Dr. SEEMA BAWA
Professor & Head
Computer Science & Engineering. Department
Thapar University
Patiala

Dr. R.K.SHARMA
Dean (Academic Affairs)
Thapar University,
Patiala.

Acknowledgement

I wish to express my sincere gratitude to Mr. Ravinder Kumar, Lecturer, Computer Science & Engineering Department for providing invaluable guidance and suggestions which inspired me to submit this thesis report.

I would also like to thank all the staff members of Computer Science & Engineering Department who were always there at the need of the hour and provided with all the help and facilities, which I required, for the completion of this work.

Special thanks goes to Mr.Rajiv Jain, Lecturer at MIMIT, Malout for their valuable contributions, corrections and discussions to this work.

Last, but not the least I wish to thank my colleagues, who have given me moral support and their relentless advice throughout the completion of this work.

Munish Kumar

Abstract

Ranking has always been an important component for information retrieval system. The web is a massive collection of static and dynamic pages. It is an infinite source of information which includes countless hyperlinks. The collection of information provides a rich and unprecedented information retrieval source. Search Engine database contains the bulk of web pages so ranking of web page is essential for fulfill the user needs. Page Rank Linking is used to measure the importance and behavior of web page. The PageRank algorithm, which is a technique used to improve the relevance of results returned by search engine graph structure. Page Rank Algorithem calculates the rank of individual web page and Hypertext Induced Topic Search (HITS) depends upon the hubs and authority framework. A fast and efficient page ranking mechanism for web retrieval remains as a challenge. Several web page ranking algorithms Hypertext Induced Topic Search (HITS) and Page Rank have been proposed. Ranking has always been an important component of any information retrieval system. In the case of Web search its importance becomes critical. Due to the size of the Web, it is imperative to have ranking functions that capture the user needs. To this end the Web offers a rich context of information which is expressed through the hyperlinks

In thesis the analysis of the importance of Web Page by using ranking algorithms is done and has proposed the new sNorm (p) Ranking algorithm. And its quality has been compared with existing algorithms like Hypertext Induced Topic Search (HITS), Stochastic Approach for Link-Structure Analysis (SALSA) and Norm (p). The algorithms are tested over multiple hyperlinks graph.

Keywords: Search Engine, Page Rank, HITS, Hyperlink, SALSA, HUBAVG, sNorm(p), Graph.

TABLE OF CONTENTS

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 Web Search Motivation

Now a days searching on the internet is most widely used operation on the World Wide Web. The amount of information is increasing day by day rapidly that creates the challenge for information retrieval. There are so many tools for perform efficient searching. Due to the size of web and requirements of users creates the challenge for search engine page ranking [1]. Ranking is the main part of any information retrieval system. In the Search Engine

1. Crawler: used for retrieves the web pages and web contents
2. Indexer: stores and indexes information on the retrieved pages
3. Ranker: Measure the importance of Web Page
4. Retrieval Engine: performs lookups on index tables against query

The web has a hierarchical structure: every day pages are added, deleted, and modified. The size of the web is on the order of more than a billion pages, and many of those pages contain redundant or incorrect information. A search tool on the web must be able to distinguish high-quality pages from low-quality pages. In addition, users of web search tools also present a challenge to IR researchers and developers. The average web IR user enters very short queries, does not make use of system feedback to revise the query, seldom performs a search using advanced search options, and generally views only the top few documents returned by the search

If user sends the query for particular topic, then Web can have hundred, even thousands results regarding that query. But if the Ranking algorithm does not provide the result within the top few positions of the ranking then that search engine is useless

and not efficient. The users have no patience to go through the hundred pages to find the one which they want. So the quality ranking of web page becomes essential. The needs of users are different, so random page may be highly relevant to the query [2]. The main task of ranking of web page is to identify the importance of web page. Mainly In links to the pages and out links from the page can give idea about the context of the page. In this thesis we will discuss three algorithms for Ranking of Web Pages which is Page Rank, HITS and SALSA. We have proposed the sNorm (p) webpage ranking algorithm.

1.2 Analysis of Ranking

Analysis of ranking start with the collection of pages and algorithm precedes the hyperlinks between the pages and constructing the hyperlink graph. The hyperlink Graph used the node as web page and edge for hyperlink between various web pages. The algorithms operate on the hyperlink graph and produce the weight for each page. This weight is used for ranking of particular web page.

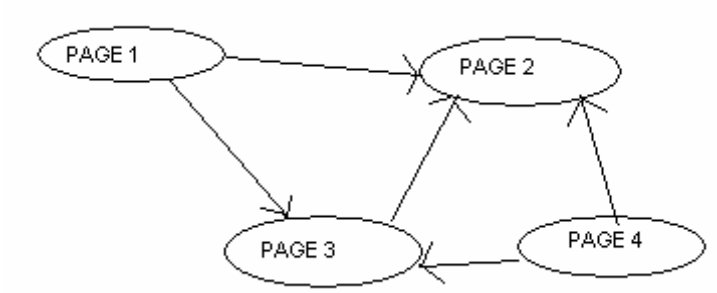

Figure 1.1: Hyperlink Graph

The Page Rank is introduced by Brin and Page (1998), later becomes a integrate component of Google etc search engine [7]. Kleinberg (1999) introduced the hubs and authorities for web pages [4]. Each and every page is associated with hub or authority page. Authority weight of a web page is the sum of the total weights of the pages that are pointed to this page and the hub weight of web page is the sum of weights of pages that are pointed by this page. Kleinberg proposed the HITS (Hyperlink Induced

Topic Search) Algorithm for computing the weight for web page [6]. HITS algorithm has two properties symmetry and equality [5]. First property is the symmetry means hub and authority weights are computed as the same way authority weight is the sum of total of hub weights and hub weight is the sum of total of authority weight. In the case of Google and PageRank, performing a user search requires two general steps:

1. Perform text query: group and locally rank pages according to traditional IR methods and
2. Merge these results with the global PageRank scores to order the documents returned by the search

Traditional IR methods involve a textual search of the pages in the web where query terms are matched to the documents and a relevancy set is created. The relevancy set is the group of pages most closely related to the query terms

1.3 Topic Specification

Topic specification can de defined by quality of documents related to query. Generally users give the ambiguous queries to search engines. So the resulting pages returned by search engine may or may not be related to query. For example, user will put the query like Database then it can be several hyperlinks like related to database research papers, database books, database engineering software etc. So by topic specification user can get the links about the all aspects of given query topic. Search Engines like Yahoo! Get vast information on various topics arranged hierarchically. The Page Rank algorithm introduced by Brin and Page later becomes a commercial success story as an integral component of Google Search Engine [8]. The linking between web pages provides the useful information. Whenever a page author create link it is as he is recommending the destination page.

Kleinberg introduced the HITS Algorithm for Topic specification [6]. Kleinberg provide the hubs and authorities paradigm where every page is associated with hub and authority weight. The authority weight of page is the sum of the hub weights of the pages that point to this page and the hub weight is the sum of the authorities'

wrights of the pages that are pointed to by this page. HITS starts with a focused hyperlink graph of the www for a topic, using results from some existing search engines. A good hub page should contain a set of good links related to the query topic, whereas a good authority page pointed by good hub.

The HITS is iterative algorithm for computing the weight of web page. A good hub links to a number of good authorities and a good authority is one, which is pointed to by good hubs.

1.4 Limitations of Previous Information Retrieval Systems

Previous Information Retrieval systems work well with controlled, finite collection of documents. Documents in such collections are generally self contained units and they are truthful about their contents. Relevance of a document to the user query can be evaluated easily for such collections. Quality of results can be evaluated in terms of precision. Also, users are not willing to go beyond the top few results.
Similarly, precision also cannot be considered as an important measure. Most of the users give short queries and there will be thousands of documents containing that query. Further, many documents are not truthful about their contents. How can one decide the most relevant top 10-20 pages for a query related thousands of candidate pages?

1.5 Hyperlink Graph of Web Pages

The hyperlink graph of web pages is a directed graph. Pages are represented as nodes and hyperlinks between them as directed edges. The WWW contains billions of such nodes and edges. In short it can be called a Hyperlink *Grap*h. A network of these links is a rich source of latent information. There is a central assumption in most link based analysis algorithms, that hyperlink confers authority. But not all the links carry the same weight. Many Search Engine Optimization (SEO) companies purposefully create artificial hyperlinked communities so as to improve their rank. Some times

sites are mirrored at many places.

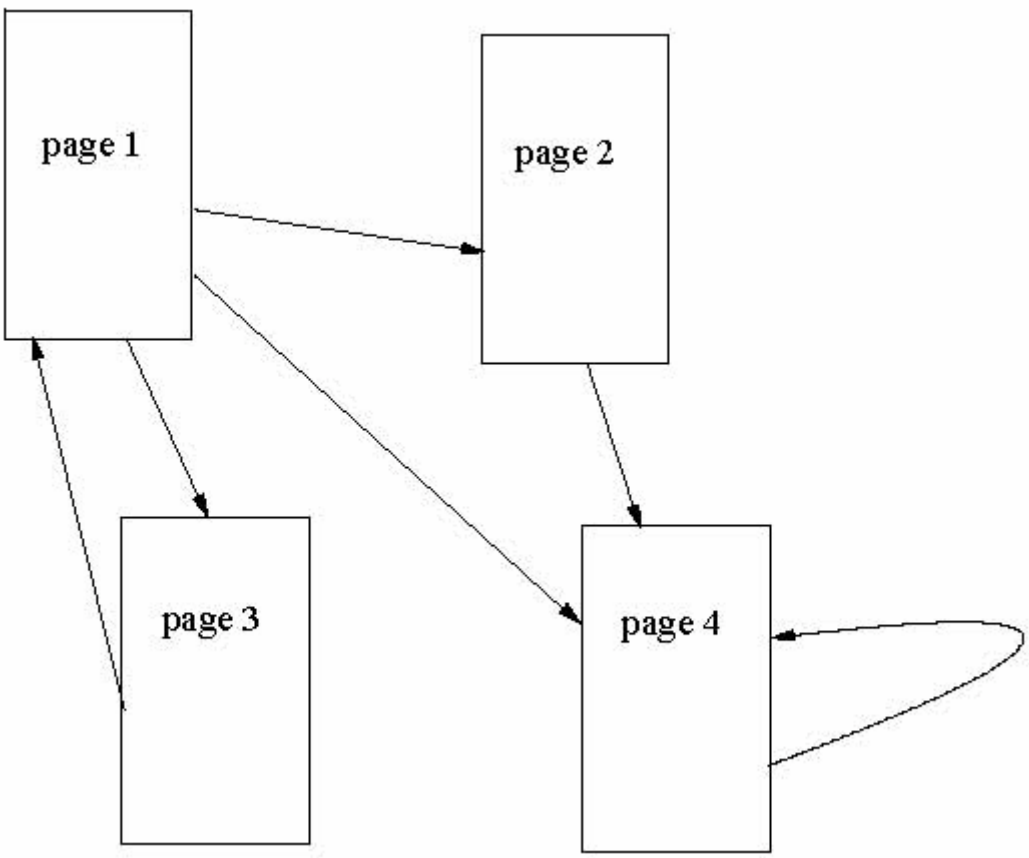

Figure 1.2: Hyper Linked Pages

One possible solution is, to have edge weights in the graph. But then, deciding perfect weights of edges and updating them with time is also a challenging problem. Also graph models capture only static aspects of the WWW [9].

We can say that this hyperlink Graph is efficient for retrieve the information which every user want. But hyper link based analysis can provide valuable information which can be used along with other methods. Successful search engines like Google depend heavily on link analysis and it also uses other information such as text, anchor text, etc., along with a link based ranking scheme.

1.6 Measure the Quality of Web Searching

Quality of Web search results is definitely depending upon matter. Web Page Importance may be varying from user to user based on link structure of WWW how can we define a good page? Will it be query dependent? Or can it be query independent? Current Web search systems respond to user queries within a fraction of a second. Users will not mind having a Web search system that responds within a few seconds, provided it returns considerably better results. We explain few factors for measure the importance of web Page.

1.6.1 Popularity

Popularity of a page can be calculated with number inlinks it has. The Popular page has higher degree of inlinks there. Here we assume that, if many pages point to a page then it should be a popular page.

1.6.2 Centrality

Distance from node u to v can be defined as minimum number of links via which we can reach v from *u*. Radius of a node is its maximum distance from any node in the graph. Center of the graph is the node with the smallest radius.

1.6.3 Quality

Quality of a page can be recursively defined as the sum of the quality of pages pointing to it. Here we consider not just number of inlinks but also quality of those inlinks. This is the motivation behind PageRank.

1.6.4 Informativeness

A node is informative if it points to several nodes that contain useful information. Here we consider not just number of outlinks, but also quality of nodes pointed.

1.6.5 Authority

Authority of a node is similar to the quality of the node with the difference that

authority is measured with respect to some focused tiny sub graph on a particular topic.

1.7 Existing Web Page Ranking Algorithms

In this section we review some of the most important Page Ranking algorithms for Web search. The main algorithms considered are In Degree, PageRank, HITS, and SALSA. PageRank was proposed in 1998 and HITS was proposed in 1999 where as SALSA was published in 2000. The feature common to all of them is that they are based on eigenvector computation.

1.7.1 The In Degree Algorithm

A Simple heuristic that can be viewed as the predecessor of link analysis ranking is to rank the page according to their popularity. The popularity of the page is measured by the number of pages that link to this page. We refer to this algorithm is called in degree algorithm, since it ranks pages according to their in-degree in the graph G. That is for every node i

$$A(i) = \frac{B(i)}{E} \tag{1.1}$$

The In Degree algorithm is not efficient for measure the authority of webpage [10]. B(i) determines the number of backward links of page and E represent the total number of edges in hyperlink graph.

1.7.2 The PageRank Algorithm

The PageRank computations require several passes, called "iterations", through the collection to adjust approximate PageRank values to more closely reflect the theoretical true value [7]. A probability is expressed as a numeric value between 0 and 1. A 0.5 probability is commonly expressed as a "50% chance" of something happening. Hence, a PageRank of 0.5 means there is a 50% chance that a person clicking on a random link will be directed to the document with the 0.5 PageRank.The In degree algorithm is that a good authority is a page that is pointed to by many nodes in the graph G. May be all the pages not carry the same weight. It is not only

important how many pages is point to a page, but also quality of pages. The Page Rank Algorithm determines the random selection on the graph G that simulates the random surfer. Normally there are many nodes with no outgoing links, so remove these nodes from the hyperlink graph and run the PageRank algorithm on the resulting graph. Page Ranking is classified into two categories:

a) Content Based Page Ranking
b) Link Connectivity based Page Ranking

The Page Rank Algorithem does not rank web sites as a whole but is determined for each page individually according to their authoritativeness

The probability, at any step, that the person will continue is a damping factor *d*. but it is generally assumed that the damping factor will be set around 0.85. The damping factor is subtracted from 1 (and in some variations of the algorithm, the result is divided by the number of documents in the collection) and this term is then added to the product of the damping factor and the sum of the incoming PageRank scores.

$$PR(A) = (1-d) + d\left(\left(\frac{PR(ti)}{L(ti)}\right) + \ldots\ldots\ldots + \left(\frac{PR(tn)}{L(tn)}\right)\right) \qquad (1.2)$$

Here d is the damping factor and L is represent the outbound link for node tn. The Page Rank Algorithm [7] performs a random walk on the graph G that simulates the behavior of a "Random Surfer". The surfer starts from some node chosen according to their distribution D. At each step the surfer proceeds with probability 1-e an outgoing links is picked uniformly as random and surfer move to a new page.

1.7.3 The HITS (Hyperlink Induced Topic Search) Algorithm

Kleinberg's hypertext-induced topic selection (HITS) algorithm is also developed for ranking documents based on the link information among a set of documents [6]. The algorithm produces two types of pages:

- *Authority:* pages that provide an important, trustworthy information on a given topic .
- *Hub:* pages that contain links to authorities.

Authorities and hubs exhibit a mutually reinforcing relationship: a better hub points to many good authorities, and a better authority is pointed to by many good hubs.

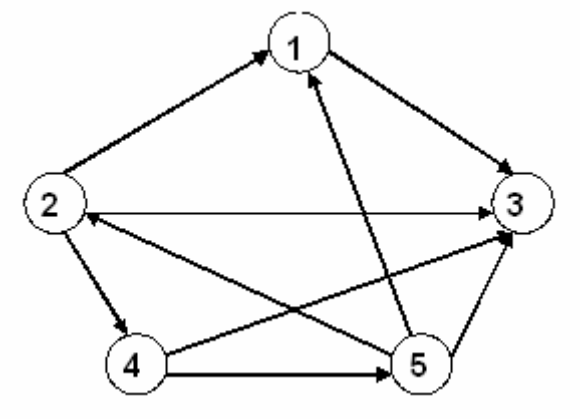

Figure 1.3: HITS Hyperlink Graph

Hubs point to lots of authorities. Authorities are pointed to by lots of hubs and together they form a bipartite graph. The idea behind the **HITS** (Hyperlink Induced Topic Distillation) algorithm is that the authorities and hubs mutually reinforce each other. Authority weight of a page is calculated as a sum of hub weights pointing to it, and weight of a hub as a sum of weights of authorities pointed to by it.

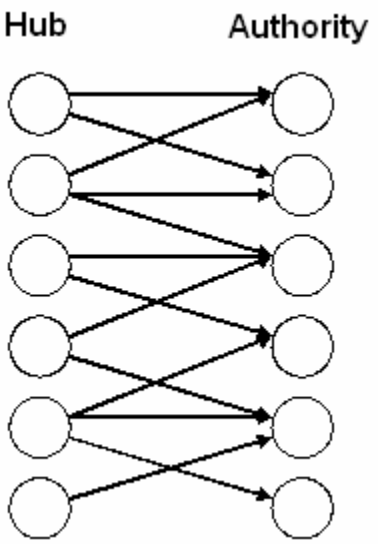

Figure 1.4: HUB and Authority Links

In other words a hub is as good as the authorities linked by it, and vice versa [6]. The notation of the algorithm is as follows. Let S be a set of pages for which hub and

authority weights are being calculated, n number of pages in the set. Then H is a subset of S containing pages acting as hubs, and A is a subset of S containing authorities. Since each page can be an authority and a hub, A and H overlap. For every page i in its hub role F(i) is the number of outgoing links. For every page i in its authority role B(i) is the number of incoming links. The n-dimensional vector of authority weights is denoted as a, and vector of hub weight – as h. Then hub and authority weights are calculated by the following formula:

$$a_i = \sum_{j \in B(i)} h_j \quad \text{and} \quad h_j = \sum_{i \in F(j)} a_i \tag{1.3}$$

The process is iterative. First all the weights receive value of 1. Then hubs and authority weights are calculated and the vectors are normalized. This stage is repeated until vectors a and h converge. These are the following step for applying HITS algorithm.

- Determines a *base* set *S*
- let set of documents returned by a standard search engine be called the *root* set *R*
- Initialize *S* to *R*
- Add to *S* all pages pointed to by any page in *R*.
- Add to *S* all pages that point to any page in *R*
- Maintain for each page *p* in *S:*
- Authority score**:** a_p (vector a)
- Hub score*:* h_p (vector ***h***)
- For each node initiliaze the a_p and h_p to 1/n
- In each iteration calculate the authority weight for each node in S

$$a_p = \sum_{q:q \to p} h_q$$

- In each iteration calculate the hub weight for each node in S

$$h_p = \sum_{q:p \to q} a_q$$

The hub weights are computed from the current authority weights, which were computed from the previous hub weights. After new weights are computed for all nodes, the weights are normalized**:**

$$\sum_{p \in S} (a_p)^2 = 1 \qquad \text{and} \qquad \sum_{p \in S} (h_p)^2 = 1 \qquad (1.4)$$

HITS emphasizes mutual reinforcement between authority and hub webpages, while PageRank does not attempt to capture the distinction between hubs and authorities. It ranks pages just by authority. HITS is applied to the local neighborhood of pages surrounding the results of a query whereas PageRank is applied to the entire web. HITS is query dependent but PageRank is query-independent [4]. Both HITS and PageRank correspond to matrix computations. Both can be unstable: changing a few links can lead to quite different rankings. PageRank doesn't handle pages with no outedges very well, because they decrease the PageRank overall. HITS is a general algorithm used for calculating the authority and hubs in order to rank the retrieved data. The basic aim of that algorithm is to induce the Web graph by finding set of pages with a search on a given topic (query). Results demonstrates that it is good in calculating the authority nodes and hubness.

1.7.4 The SALSA Algorithm

"The Stochastic Approach for Link-Structure Analysis (SALSA) by Ronnie Lempel and Shlomo Moran [11], which is published on the Web at the website for the Ninth International World Wide Web Conference, held in Amsterdam, The Netherlands, from May 15-19, 2000. The SALSA method examines random walks on graphs derived from the link structure among pages in a search result. While preserving the theme that Web sites pertaining to a given topic should be split into hubs and authorities, it replaces Kleinberg's Mutual Reinforcement method by a stochastic method, in which the coupling between hubs and authorities is less tight. The method is based on considering a bipartite graph G, whose two parts correspond to hubs and authorities, where an edge between hub r and authority s means that there is an informative link from r to s. Then, authorities and hubs pertaining to the dominant topic of the sites in G should be highly visible (reachable) from many sites in G.

These sites are identified by examining certain random walks in G, under the provision that such random walks will tend to visit these highly visible sites more frequently than other, less connected sites. The SALSA approach is based upon the theory of Markov chains, and relies on the stochastic properties of random walks performed on a collection of sites. It differs from Kleinberg's Mutual Reinforcement approach in the manner in which the association matrices are defined. The SALSA approach also initially assumes uniform probability over all pages, and relies on the random walk process to determine the likelihood that a particular page will be visited.

An alternative algorithm SALSA was proposed by Lempel and Moran that combines the ideas from both HITS and Page Rank Algorithms. Stochastic Algorithm for Link Structure Analysis (SALSA) is a combination of PageRank and HITS. It calculates hub and authority values per query like HITS. We had considered row normalized adjacency matrix E_r and Column normalized adjacency matrix E_c. SALSA is less susceptible rather than HITS. As in the case of HITS, visualize the graph G as a bipartite graph, where hubs point to authorities [5]. The SALSA algorithm performs a random walk on the bipartite hubs and authority graph, alternating between the hub and authority nodes. The SALSA algorithm can be though of as a variation of the HITS Algorithm.

1.8 Structure of the Thesis

In Chapter 2 we discuss the Literature Survey of existing Algorithms for Page Ranking.
In Chapter 3 We discuss the Proposed Solution for Problem Statement
In Chapter 4 We perform the Experiments and Results.
Conclusion is outlined in Chapter 5.

CHAPTER 2

LITERATURE SURVEY

2.1 PAGERANK Algorithm

The World Wide Web contains an enormous amount of information, but it can be exceedingly difficult for users to locate resources that are both high in quality and relevant to their information needs [7]. There are a number of fundamental reasons for this. The Web is a hypertext corpus of enormous size approximately three hundred million Web pages as on this writing and it continues to grow at a phenomenal rate.

Traditional Information retrieval system is not efficient for web search. Because size of the web is more than a billion pages, and many of those pages contain incorrect information. The Ranking algorithm is used to distinguish high-quality pages from low-quality pages. The web Information retrieval users enters very short queries, the search tool generally views only the top few pages returned by the search. Ideally, a search engine should be ranked by query relevance. Determining a page's relevance to query terms is a complex problem for an Information retrieval system, but the inherent hyperlink structure of the web may be used to generate an approximation. This idea is implemented in the PageRank algorithm [7], first devised by Sergey Brin and Larry Page at Stanford University in 1998, and implemented by Google, a highly successful web search engine. Textual search of the pages in the web where query terms are matched to the documents and a relevancy set is created involved in traditional IR System. The relevancy set is the group of pages most closely related to the query terms.

PageRank is a family of algorithms for assigning numerical weightings to hyperlinked documents (or web pages) indexed by a search engine. The popular search engine Google, to help determine a page's relevance or importance uses the PageRank system. Google's founders Larry Page and Sergey Brin developed it, while at

Stanford University in 1998 [7]. The PageRank algorithm evaluates webpage reputations based on the hyperlinks that connect them. The PageRank of a page is defined recursively and depends on the number and PageRank metric of all pages that link to it ("incoming links"). A page that is linked by many pages with high rank receives a high rank itself. If there are no links to a web page there is no support of this specific page. The Google Toolbar PageRank goes from 0 to 10. It seems to be a logarithmic scale. The exact details of this scale are unknown. PageRank is a numeric value that represents how important a page is on the web. Google figures that when one page links to another page, it is effectively casting a vote for the other page. The more votes that are cast for a page, the more important the page must be. Also, the importance of the page that is casting the vote determines how important the vote itself is. Google calculates a page's importance from the votes cast for it.

PageRank relies on the uniquely democratic nature of the web by using its vast link structure as an indicator of an individual page's value. Google interprets a link from page A to page B as a vote, by page A, for page B. But, Google looks at more than the sheer volume of votes, or links a page receives; it also analyzes the page that casts the vote. Votes cast by pages that are themselves "important" weight more heavily and help to make other pages "important." Once a relevancy set is created containing documents relevant to the query terms, the Google system merges relevancy set rankings with PageRank scores and displays the results. The PageRank values are pre-calculated and stored for all pages known to the IR system. This means every page in the web has a PageRank score that is completely independent of query terms. A search that returns PageRank scores is reporting the importance hierarchy of pages containing the query terms.

PAGERANK Pseudo code:

```
1 start
    PR(i) = 1 for all i;
2  For each P in Page
   PR(P) = (1-d)/N + d * ((PR(t1)/C(t1) + PR(t2)/C(t2) ...)
```

for all ti that point to P

(C(ti) is the outlink of ti)

3 calculate linear polynomial with linear statistical method.

2.1.1 Hyperlink Structure of the Web

A set of pages in the web may be modeled as nodes in a directed graph. The edges between nodes represent links between pages. A graph of a simple 8-page web is depicted in Figure 2.1 below. The directed edge from node two to node three signifies that page two links to page three. However, page three does not link to page two, so there is no edge from node three to node two.

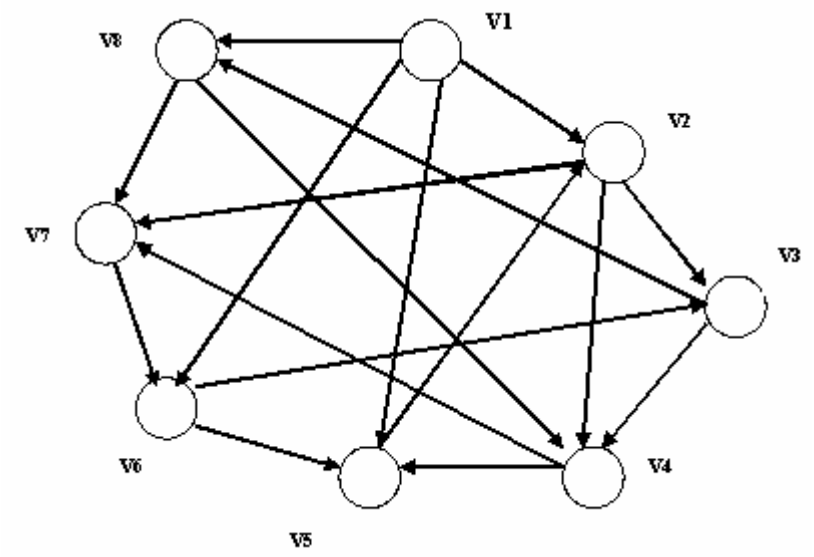

Figure 2.1: Hyperlink Graph Model of 8-Nodes

The Page Rank constructs page importance hierarchies based upon the link structure of the web. The inlinks from a page may be seen as a recommendation. Generally, more important pages will have more inlinks. Inlinks from important pages will also have a greater effect on PageRank for a particular page. The calculation of PageRank is recursive, building the rank for a particular page based on the ranks of the pages that link to it.

Most search engines have underlying rankings of web pages that they then filter based on a user's query in order to determine the web pages most likely to be of use to the user. A significant component of the rank of a particular page is based on how other

web pages rank the webpage (in the simplest case, we can say a webpage ranks another highly if it links to it). Typically sites classified as good raters will be given more weight in determining the rank of a page than those classified as bad raters. Normally rating ability and ranking are treated as the same thing, so there is no distinguishing between being a good source of information and being a good judge of other web pages.

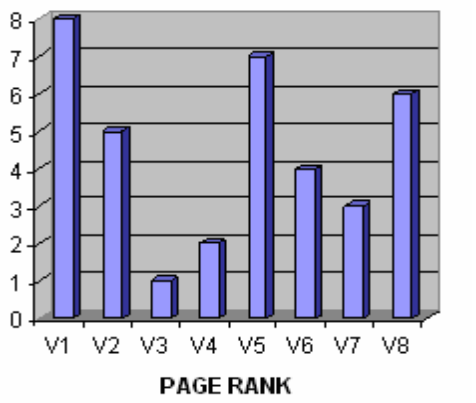

Figure 2.2: Page Rank Implementation Graph

2.2 Hypertext Induced Topic Search Algorithm

HITS algorithm is proposed by Kleinberg in 1999 [4]. Kleinberg proposed the hypertext induced topic search algorithm for topic search on the WWW. Brin and Page proposed a more improvement for the importance of Web Pages. HITS algorithm is also developed for ranking documents based on the link information among a set of documents. Brin and Page proposed two level schemes for importance of web pages hub identity and authority identity. The HITS algorithm starts with a focused hyperlink graph for a WWW for a query. It then does iterative, eigenvector based computation to identify good hub pages and authority pages Independent of Brin and Page, Kleinberg [6] proposed a more refined notion for the importance of Web pages. He proposed a two-level weight propagation scheme where endorsement is conferred on authorities through hubs, rather than directly between authorities. In his framework, every page can be thought of as having two identities. The hub identity captures the quality of the page as a pointer to useful resources, and the

authority identity captures the quality of the page as a resource itself. A good authority is a source of useful information, while a good hub is a page that contains a useful collection of links. If we make two copies of each page, we can visualize graph G as a bipartite graph, where hubs point to authorities. There is a mutual reinforcing relationship between the two. A good hub is a page that points to good authorities, while a good authority is a page pointed to by good hubs. In order to quantify the quality of a page as a hub and an authority, Kleinberg associated every page with a hub and an authority weight [6]. Following the mutual reinforcing relationship between hubs and authorities, Kleinberg defined the hub weight to be the sum of the authority weights of the nodes that are pointed to by the hub, and the authority weight to be the sum of the hub weights that point to this authority. Let h denote the n-dimensional vector of the hub weights, where hi, the i-*th* coordinate of vector h, is the hub weight of node i. We have that

$$a_i = \sum_{j \in B(i)} h_j \quad \text{and} \quad h_j = \sum_{i \in F(j)} a_i \tag{2.1}$$

In matrix-vector terms

$$a = W^{\mathrm{T}} h \quad \text{and} \quad h = Wa \tag{2.2}$$

Building upon the mutual reinforcing relationship between hubs and authorities, Kleinberg proposed the following iterative algorithm for computing the hub and authority weights. Initially all authority and hub weights are set to 1. At each iteration, the operations *O* ("out") and *I* ("in") are performed. The *O* operation updates the authority weights, and the *I* operation updates the hub weights, both using the equation 2.2. A normalization step is then applied, so that the vectors *a* and *h* become unit vectors in some norm. The algorithm iterates until the vectors converge. Let a^t denote the authority vector after the t-th iteration. Given a constant $\in$, we say the vector a^t has converged, if where $\left\| a^t - a^{t-1} \right\| \leq \in$, $\|.\|$ is the normalization norm. This idea was later implemented as the HITS (Hyperlink Induced Topic Distillation) algorithm. The algorithm is summarized in figure 2.3.

Initialize all weights to 1

Repeat until the weights converge

For every hub $i \in H$

$$h_i = \sum_{j \in F(i)} a_j$$

For every authority $i \in A$,

$$a_i = \sum_{j \in B(i)} h_j$$

Normalize

Figure 2.3: The HITS Algorithm

Kleinberg proves that the algorithm computes the principal left and right singular vectors of the adjacency matrix *W*. That is, the vectors *a* and *h* converge to the principal right eigenvectors of the matrices $M_H = W^T W$ and $M_H^T = WW^T$, respectively. The convergence of HITS to the singular vectors of matrix W is subject to the condition that the initial authority and hub vectors are not orthogonal to the principal eigenvectors of matrices M_H and M_H^T respectively. Since these eigenvectors have non-negative values, it suffices to initialize all weights to positive values, greater than zero.

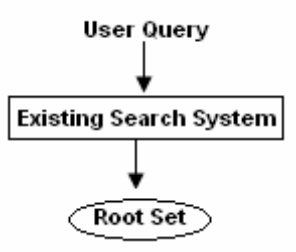

Figure 2.4: Root set generation

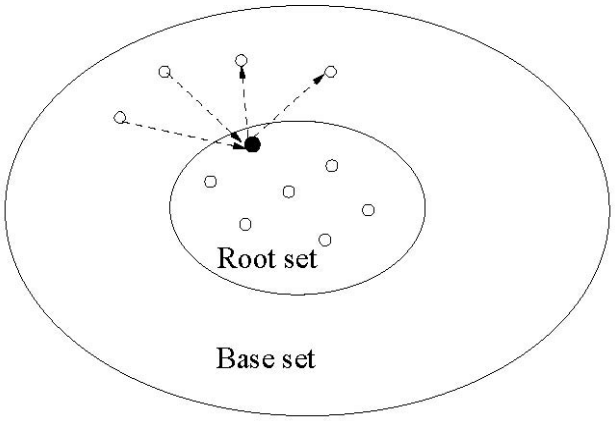

Figure 2.5: Generating the Base Set from Root

HITS Pseudo code:

1) Start
2) Initial for all N ∈ Graph (G): $a_{N=}h_{N=1}$
3) for i=1 to k
   a. for all N ∈ Graph (G): $a_{N=\sum} h_N$
   b. for all N ∈ Graph (G): $h_{N=\sum} a_N$
   c. for all N ∈ Graph (G): $a_N = a_{N/C} C: \sum (a_{N/C})^2=1$
   d. for all N ∈ Graph (G): $h_N = h_{N/C} C: \sum (h_{N/C})^2=1$
4) Exit

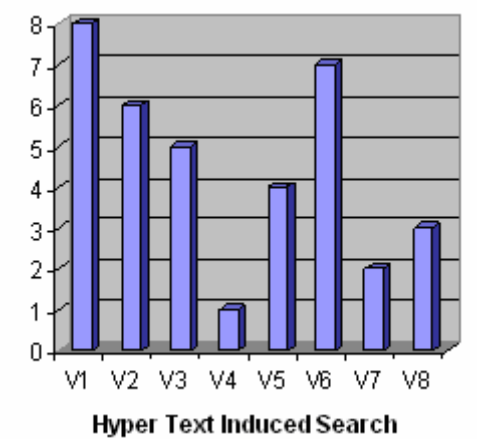

Figure 2.6: HITS Implementation Graph

HITS algorithm computes lists of hubs and authorities for WWW search topics. Beginning with a search topic, specified by one or more query terms, the HITS algorithm applies two main steps: a sampling component, which constructs a focused collection of several thousand Web pages likely to be rich in relevant authorities; and a weight-propagation component, which determines numerical estimates of hub and authority weights by an iterative procedure. The pages with the highest weights are returned as hubs and authorities for the search topic. We view the Web as a directed graph, consisting of a set of nodes with directed edges between certain pairs of the nodes. Given any subset S of nodes, they induce a subgraph containing all edges that connect two nodes in S. The first step of the HITS algorithm constructs the subgraph in which we will search for hubs and authorities [5]. Our goal is to have a subgraph that is rich in relevant, authoritative pages, we construct such a subgraph as follows. We first use the query terms to collect a root set of pages from an index based search engine of the type described in the introduction. We do not expect that this set necessarily contains authoritative pages; however, since many of these pages are presumably relevant to the search topic, we expect at least some of them to have links to most of the prominent authorities. We therefore expand the root set into a base set by including all pages that are linked to by pages in the root set, and all pages that link to a page in the root set (up to a designated size cut-of). We restrict our attention to this base set for the remainder of the algorithm; we find that this set typically contains roughly 1000-5000 pages, and that (hidden) among these are a large number of pages that one would subjectively view as authoritative for the search topic.

We work with the subgraph induced by the base set, with one modification. We find that links between two pages with the same WWW domain very often serve a purely navigational function, and thus do not correspond to our notion of links as conferring authority. By WWW domain" here, we mean simply here the first level in the URL string associated with a page. We therefore delete all links between pages with the same domain from the subgraph induced by the base set, and apply the remainder of the algorithm to this modified subgraph.

2.2.1 Features of HITS Algorithm

One of the fundamental problems in Information Retrieval is the ranking of search results. In the context of web search, where the corpus is massive and queries rarely contain more than three terms, most searches produce hundreds of results. Given that the majority of search engine users examine only the first page of results, effective ranking algorithms are key to satisfying user's needs. Leading search engines rely on many features in their ranking algorithms. Sources of evidence can include textual similarity between query and documents (or query and anchor texts of hyperlinks pointing to documents), the popularity of documents with users (measured for instance via browser toolbars or by clicks on links in search result pages), and finally hyper linkage between web pages, which is viewed as a form of peer endorsement among content providers.

The idea underlying the HITS algorithm can be captured in the following recursive definition of quality: "A good authority is one that is pointed to by many good hubs, and a good hub is one that points to many good authorities". Therefore, the quality of some page *p* as an authority (captured by the authority weight of page *p*) depends on the quality of the pages that point to *p* as hubs (captured in the hub weight of the pages), and vice versa. Kleinberg proposes to associate the hub and authority weights through the addition operation. The authority weight of a page *p* is defined to be the sum of the hub weights of the pages that point to *p*, and the hub weight of the page *p* is defined to be the sum of the authority weights of the pages that are pointed to by *p*. This definition has the following two implicit properties. It is *symmetric*, in the sense that both hub and authority weights are defined in the same way. If we reverse the orientation of the edges in the graph *G*, then authority and hub weights are swapped. The HITS algorithm is also *egalitarian*, in the sense that when computing the authority weight of some page *p*, the hub weights of the pages that point to page *p* are all treated equally (similarly when computing the hubs weights). However, these two properties may some times lead to non-intuitive results. Consider for example the graph. In this graph there are two components. The black component consists of a single authority pointed to by a large number of hubs. The white component consists of a single hub that point to a large number of authorities. If the number of white

authorities is larger than the number of black hubs then the HITS algorithm will allocate all authority weight to the white authorities, while giving zero weight to the black authority. The reason for this is that the white hub is deemed to be the best hub, thus causing the white authorities to receive more weight. However, intuition suggests that the black authority is better than the white authorities and should be ranked higher. In this example, the two implicit properties of the HITS algorithm combine to produce this non-intuitive result. Equality means that all authority weights of the nodes that are pointed to by a hub contribute equally to the hub weight of the node. As a result quantity becomes quality. The hub weight of the white hub increases inordinately because it points to many weak authorities. This leads us to question the definition of the hub weight, and consequently other implicit property of HITS. Symmetry assumes that hubs and authorities are qualitatively the same. However, there is a difference between the two. For example, intuition suggests that a node with high in-degree is likely to be a good authority. On the other hand, a node with high out-degree is not necessarily a good hub. If this was the case, then it would be easy to increase the hub quality of a page, simply by adding links to random pages. It seems that we should treat hubs and authorities in different manners. In this chapter we challenge both implicit properties of HITS. We present different ways for breaking the symmetry and equality principles and we study the ranking algorithms that emerge.

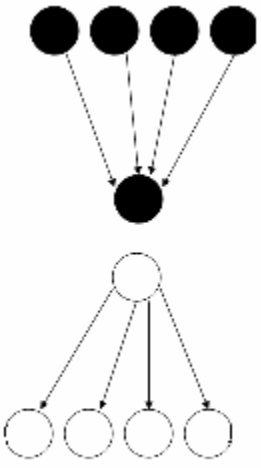

Figure 2.7: Bad Example of HITS Algorithm

2.2.2 The Hub-Averaging and Authority Averaging Algorithm

To overcome the shortcoming of the HITS algorithm of a hub getting a high weight when it points to numerous low-quality authorities, the following refinement was suggested. While using the same formula to calculate authority weights, the hub score *h* is now averaged by a number of outgoing links |*F(i)*|.

The symmetric and egalitarian nature of the HITS algorithm has the effect that the quality of the white hub is determined by the quantity of authorities it points to. Thus, the white hub is rewarded simply because it points to a large number of authorities, even though they are of low quality. We propose a modification of the HITS algorithm to help remedy the above-mentioned problem [5]. The Hub-Averaging algorithm (HubAvg) (first presented in the collaborative work with A. Borodin, G. Roberts, and J. Rosenthal) updates the authority weights like the HITS algorithm, but it sets the hub weight of some node *i* to the average authority weight of the authorities pointed to by hub *i*. Thus for some node *i* we have

$$a_i = \sum_{j \in B(i)} h_j \qquad\qquad h_i = \frac{1}{F(i)} \sum_{j \in F(i)} a_j \qquad (2.3)$$

The intuition of the HubAvg algorithm is that a good hub should point *only* (or at least mainly) to good authorities, rather than to both good and bad authorities. HubAvg assigns the same weight to both black and white hubs, and it identifies the black authority as the better authority. The HubAvg algorithm can be viewed as a "hybrid" of the HITS and SALSA algorithms. The operation of averaging the weights of the authorities pointed to by a hub is equivalent to dividing the weight of a hub among the authorities it points to.

$$h_i = \sum_{j \in F(i)} a_j \qquad\qquad a_i = \frac{1}{B(i)} \sum_{j \in B(i)} h_j \qquad (2.4)$$

Therefore, the HubAvg algorithm performs the *O* operation like the HITS algorithm (broadcasting the authority weights to the hubs), and the *I* operation like the SALSA algorithm (dividing the hub weights to the authorities). This lack of symmetry between the update of hubs and authorities is motivated by the qualitative difference between hubs and authorities previously discussed. The authority weights for the HubAvg algorithm converge to the principal right eigenvector of the matrix *MHA* = *WTWr*. It is interesting to observe what happens if we make the algorithm symmetric.

There are two ways to re-establish symmetry. We can let the authorities divide their weight among the hubs that point to them. In this case authority and hub weights are defined as follows.

$$a_i = \frac{1}{B(i)} \sum_{j \in B(i)} h_j \qquad\qquad h_i = \frac{1}{F(i)} \sum_{j \in F(i)} a_j \qquad (2.5)$$

The authority weights will then converge to the principal *left* eigenvector of the matrix WTc *Wr* and the algorithm becomes the SALSA algorithm. Alternatively, we can make the authority weight of a node be the average of the hub weights that point to that node. The authority weights will then converge to the principal *right* eigenvector of the matrix *WT c Wr*. Since this is a stochastic matrix, the principal right eigenvector is the uniform vector. Thus, the algorithm degenerates to the algorithm that assigns the same weight to each node in the graph. So in order to achieve a high weight a hub should link good authorities. Unfortunately this approach has its own flaw. Consider two hubs pointing to an equal number of equally good authorities. The two hubs are identical until one puts one more link to a low quality authority. The average sum of the authorities it points to sinks, and it gets penalized in weight. This is quite illogical but can be fixed by using so-called Authority Threshold Algorithm.

2.2.3 Norm (*p*) Family of Algorithms

The Authority Threshold algorithm operates on the principle of *preferential treatment* of the authority weights. That is, higher authority weights should be more important in the computation of the hub weight. This principle is enforced by applying a threshold operator. A smoother approach is to *scale* the weights, so that lower authority weights contribute less to the hub weight. An obvious question is how to select the scaling factors. A natural solution is to use the weights themselves to determine the scaling factors. This idea is implemented in the Norm (*p*) family of algorithms. In this case we set the hub weight of node *i* to be the *p*-norm of the vector of the authority weights of the nodes pointed to by node *i*. Finally, we broke down our query set by query specificity, and found that SALSA is most effective for general queries.

Norm (p)
Initialize authority weights to 1
Repeat until the weights converge:
For every hub $i \in h$

$$h_i = \left(\sum_{J \in F(I)} a_j^p \right)^{1/p}$$

For every authority $i \in a$

$$a_i = \sum_{j \in B(i)} h_j$$

Normalize

Figure 2.8: Norm (p) Algorithm

Again it is interesting to examine the behavior of the algorithm in the extreme cases of the value p. For p=1 the Norm (1) algorithm is the HITS algorithm. For p=∞ the p-norm reduce the max operator.

2.3 SALSA (Stochastic Approach for Link Structure Analysis) Algorithm

An alternative algorithm, SALSA, was proposed by Lempel and Moran[11] that combines ideas from both HITS and PageRank. As in the case of HITS, visualize the hyperlink graph *G* as a bipartite graph, where hubs point to authorities. The SALSA algorithm performs a random walk on the bipartite hubs and authorities graph, alternating between the hub and authority sides. The random walk starts from some authority node selected uniformly at random. The random walk then proceeds by alternating between backward and forward steps. When at a node on the authority side of the bipartite graph, the algorithm selects one of the incoming links uniformly at random and moves to a hub node on the hub side. When at node on the hub side the algorithm selects one of the outgoing links uniformly at random and moves to an authority. The authority weights are defined to be the stationary distribution of this random walk.

Furthermore, even when the graph *Ga* is not connected, if the starting point of the random walk is selected with probability proportional to the "popularity" (in-degree) of the node in the graph *G*, then the algorithm again reduces to the InDegree algorithm. This algorithm was referred to as pSALSA (popularity SALSA) by Borodin et al. The SALSA algorithm can be thought of as a variation of the HITS algorithm [4]. In the operation of the HITS algorithm the hubs *broadcast* their weights to the authorities, and the authorities sum up the weight of the hubs that point to them. The SALSA algorithm [11] modifies the I operation as follows. Instead of broadcasting, each hub *divides* its weight equally among the authorities to which it points. There fore,

$$a_i = \sum_{j:j\in B(i)} \frac{1}{\left|F(j)\right|} h_j \tag{2.6}$$

Similarly, the SALSA algorithm modifies the O Operations so that each authority divided the weight equally among the hubs that points to it. Therefore

$$h_i = \sum_{j:j\in F(i)} \frac{1}{\left|B(j)\right|} a_j \tag{2.7}$$

CHAPTER 3

PROBLEM STATEMENT AND PROPOSED SOLUTION

3.1 Problem Definition

In the previous chapters we provided a literature survey of existing Web Page Ranking Algorithms for the Web IR. We will concentrate on link analysis algorithm Hypertext Induced Topic Search and SALSA algorithm. In the present chapter, we will have a close look at various aspects of SALSA, problems encountered. Web search can simply be considered as the process of the user enters the query and the search system returning a set of most relevant URLs. But results returned by HITS, PAGERANK are not mostly relevant to user query and ranking of these algorithms is not efficient according to user requirement. But not all user queries are same. Kleinberg divides user queries into two type's specific queries and Broad type's queries. Search engines use links to determine the authority of pages in topics described by the link anchor text. The problem is that every link is considered as a positive endorsement with no regard to the real intention of the linking person. There is no effective way for a search engine to distinguish between positive and negative endorsements in links yet. The Web is a vast collection of completely uncontrolled heterogeneous documents. Due to these characteristic, the web poses an area of data mining research with the huge amount of information available online. The unstructured characteristic of the information sources on the Web makes automated discovery of Web information difficult. Traditional search engines provide some information to users but do not provide structural information and categorization, content-based relevance ranking of the search result, filtering or interpretation of the documents, etc. Recently, Web data mining methods appear to be useful in the context of these problems.

The Authority algorithm operates on the principle of alters the authority weights. That is, higher authority weights should be more important in the computation of the hub weight. This principle is enforced by applying a operator. A smoother approach is to *scale* the weights, so that lower authority weights contribute less to the hub weight. An obvious question is how to select the scaling factors. A natural solution is to use the weights themselves to determine the scaling factors.

3.2 sNorm (p) Algorithm

This idea is implemented in the Norm (*p*) family of algorithms. In this case we set the hub weight of node *i* to be the *p*-norm of the vector of the authority weights of the nodes pointed to by node *i*. The sNorm (p) algorithm weights are computed as follows:

$$a_{i=}\left(\sum_{j\in B(i)}\frac{1}{F(j)}h_j^{\,p}\right)^{1/p} \quad \text{and} \quad h_{i=}\left(\sum_{j\in F(i)}\frac{1}{B(j)}a_j^{\,p}\right)^{1/p} \tag{3.1}$$

The value of *p* is passed as a parameter to the algorithm. We assume that $p \in [1, \infty]$ as *p* increases the value of the *p*-norm is dominated by the highest weights. For example, or $p = 2$, we essentially scale every weight with itself. An almost identical algorithm was proposed by Gibson, Kleinberg and Raghavan for clustering categorical data.

Our work originates in the problem of *searching* on the www, which we could define roughly as the process of discovering pages that are relevant to a given query. The *quality* of a search method necessarily requires human evaluation, due to the subjectivity inherent in notions such as *relevance*. We begin from the observation that improving the quality of search methods on the www is, at the present time, a rich and interesting problem that is in many ways orthogonal to concerns of algorithmic deficiency and storage. In particular, consider that current search engines typically index a sizable portion of the www and respond on the order of seconds. Although

there would be considerable utility in a search tool i[th] a longer response time, provided that the results were of significantly greater value to a user, it has typically been very hard to say *what* such a search tool should be computing with this extra time. Clearly we are lacking objective functions that are both concretely defined *and* correspond to human notions of quality.

Initialize authority weights to 1

Repeat until the weights converge

For every hub $i \in H$

$$h_{i=}\left(\sum_{j \in F(i)} \frac{1}{B(j)} a_j^p \right)^{1/p}$$

For every authority $i \in A$

$$a_{i=}\left(\sum_{j \in B(i)} \frac{1}{F(j)} h_j^p \right)^{1/p}$$

Normalize

Figure 3.1: sNorm (p) Algorithm

The sNorm (p) algorithm can be made symmetric by setting the authority weight of a node to be the p-norm of the vector of the hub weights of the hubs that point to that node. The behavior of this symmetric version is particularly intriguing. First consider the limiting case p = ∞. When all initial weights are set to 1, then it is easy to see that all nodes will receive weight 1. If we initialize the authority weights to some other configuration then the algorithm assigns the authority weights as follows. Recall that the authority graph Gad is defined on the set of authorities A, and there exists an edge between two authorities if they have a hub in common. For every component in the graph Ga, all nodes in the component receive the weight of the node with the maximum initial weight in the component. The value of *p* is passed as a parameter to

the algorithm. We assume that $p \in [1, \infty]$ as p increases the value of the *p*-norm is dominated by the highest weights. For example, for $p = 2$, we essentially scale every weight with itself.

CHAPTER 4

COMPARATIVE EXPERIMENTAL RESULTS

In this chapter we describe experiments that were carried out with sNorm (p) and compare the results of sNorm (p) Algorithm with HITS and SALSA, Norm (p).

4.1 Experimental Setting

Experiments were performed on hyperlink graph of 8 vertices. Hyperlink graph features such as in-degree and Page Rank have been shown in significantly improve the performance of query retrieval algorithm on the web. The HITS and SALSA may expect more informative that other link based features because it is query independent. There are many attempts to improve the effectiveness of ranking algorithms.

4.2 Input Data Set and Measurement

Our evaluation is based on two data sets: a large web graph and query with associated results. The Web Graph is based on web crawling. We quantify the measuring performance of ranking algorithm is based on the MRR and MAP techniques as follow:

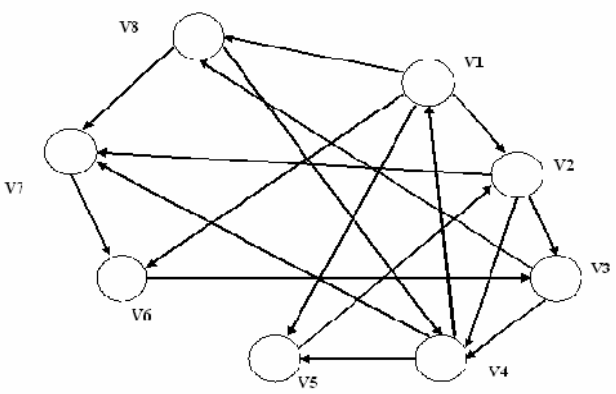

Figure 4.1: Hyperlink Graph

4.2.1 Mean Reciprocal Ranking

Reciprocal rank of the ranked result set of a query is defined to be the reciprocal value of the rank of the highest ranking relevant document in the result set. The reciprocal rank is set to be 0 if none of the highest ranking document is relevant to query [19, 20]. The mean reciprocal rank is the average reciprocal rank if all queries in the query set. Mean reciprocal rank is a statistics or evaluating any process that produces a list of possible responses to a query, ordered by probability of correctness. The reciprocal rank of a query response is the multiplicative inverse of the rank of the correct answer. The mean reciprocal rank is the average of the reciprocal ranks of a sample of queries as follow:

$$MRR = \sum_{i=1}^{n} \frac{p(i)}{\max(\ p)} \qquad (4.1)$$

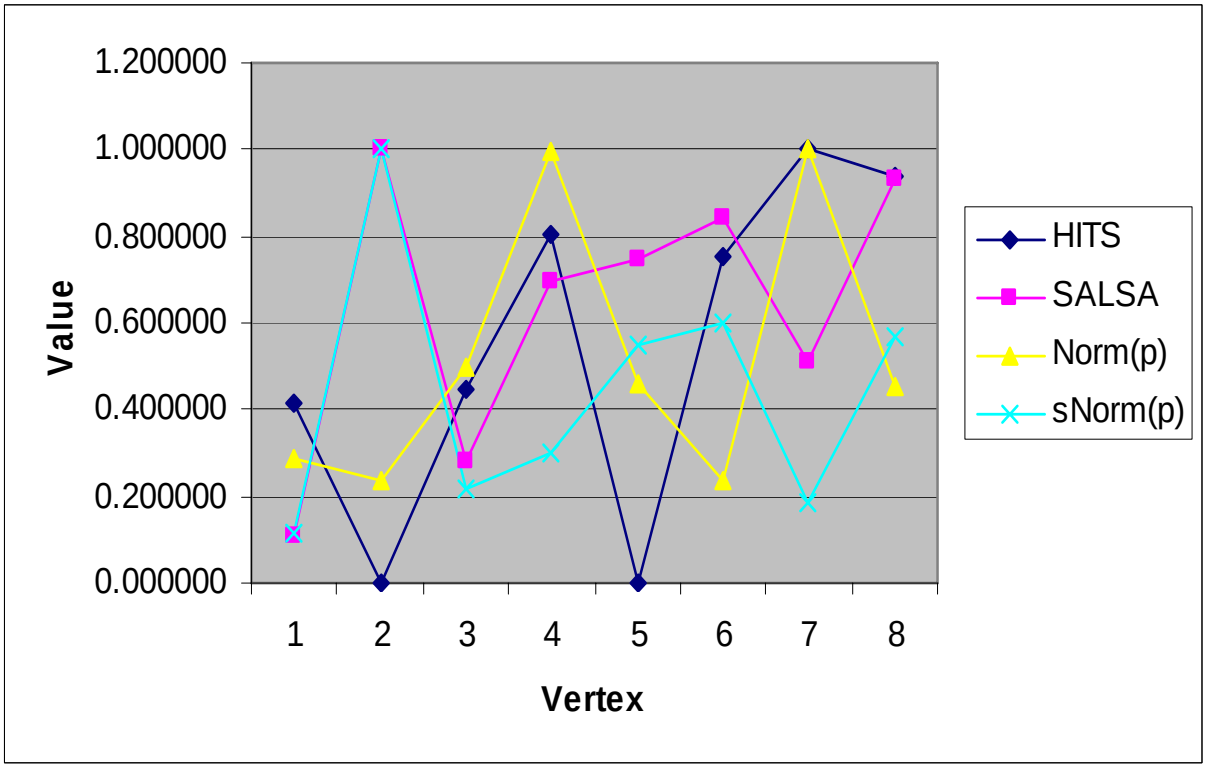

Figure 4.2: Mean Reciprocal Rank Comparison Graph

Table 4.1: Mean Reciprocal Rank

	HITS		SALSA		Norm(p)		sNorm (p)	
Vertex	**Rank Value**	**Reciprocal Rank**	**Rank Value**	**Reciprocal Rank**	**Rank Value**	**Reciprocal Rank**	**Rank Value**	**Reciprocal Rank**
V1	0.224499	0.417804	0.05535	0.110334352	0.17184	0.290171749	0.079197	0.116860753
V2	0.000010	0.000019	0.501657	1	0.138366	0.233647022	0.677704	1
V3	0.238715	0.444261	0.142388	0.28383537	0.294543	0.497369981	0.148313	0.218846281
V4	0.431851	0.803696	0.34858	0.694857243	0.590354	0.996881127	0.20397	0.300972106
V5	0.000029	0.000054	0.375372	0.748264252	0.271298	0.458118105	0.372202	0.549210275
V6	0.404010	0.751883	0.421858	0.840929161	0.138366	0.233647022	0.40794	0.601944212
V7	0.537331	1.000000	0.255737	0.509784574	0.592201	1	0.126597	0.186802793
V8	0.504154	0.938256	0.468214	0.933334928	0.269451	0.454999232	0.387128	0.571234639
	SUM	4.355972389	**SUM**	5.12133988	**SUM**	4.164834237	**SUM**	3.545871059
	MRR	0.725995398	**MRR**	0.640167485	**MRR**	0.52060428	**MRR**	0.443233882

4.2.2 Mean Average Precision

The mean average precision of a query set is the mean of the average precision of all queries in the query set. The MAP and MRR rely on the result set [19, 20]. Given a ranked set of results. The rel (i) be 1 if the result at rank is relevant and 0 otherwise. The average precision is calculated as follow:

$$MAP = \frac{\sum_{i=1}^{k} P(i)\,rel(i)}{\sum_{i=1}^{k} rel(i)} \tag{4.2}$$

The precision based on the whole list of documents returned by the system. Average precision emphasizes returning more relevant documents earlier. It is average of precisions computed after truncating the list after each of the relevant documents in turn as follow:

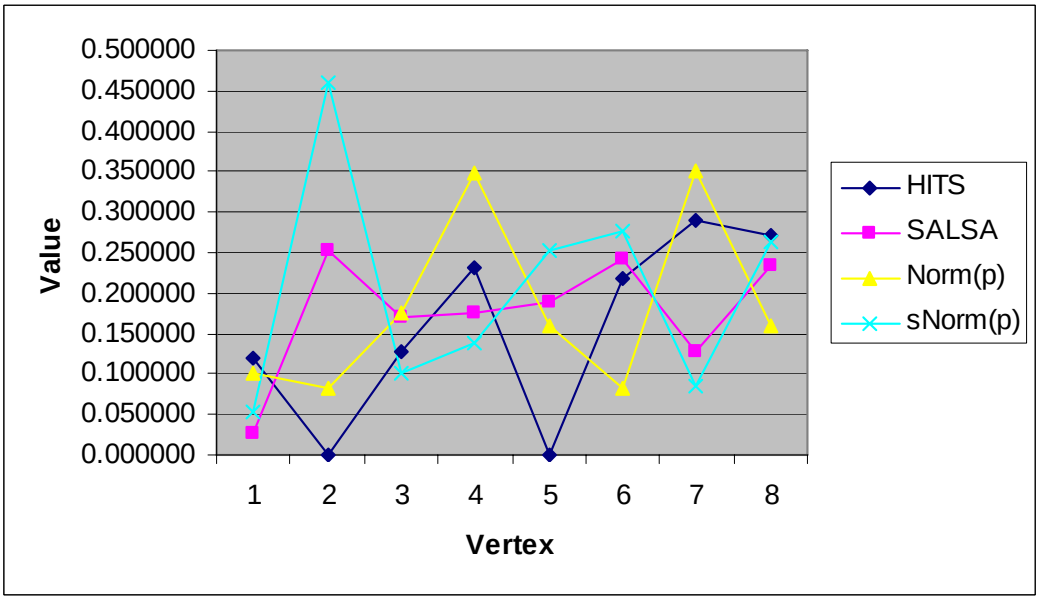

Figure 4.3: Mean Average Precision Comparison Graph

Table 4.2: Mean Average Precision

	HITS		SALSA		Norm(p)		sNorm(p)	
Vertex	Rank Value	MAP Rank	Rank Value	MAP Rank	Rank Value	MAP Rank	Rank Value	MAP Rank
V1	0.224499	0.120630	0.055361	0.027766642	0.17184	0.10176382	0.079197	0.053672124
V2	0.000010	0.000005	0.501556	0.251558421	0.138366	0.081940484	0.677704	0.459282712
V3	0.238715	0.128269	0.142505	0.171474238	0.294543	0.174428659	0.148313	0.100512313
V4	0.431851	0.232047	0.348747	0.17491615	0.590354	0.349608229	0.20397	0.138231285
V5	0.000029	0.000016	0.375327	0.188247509	0.271298	0.160662947	0.372202	0.252242784
V6	0.404010	0.217087	0.421778	0.241545287	0.138366	0.081940484	0.40794	0.27646257
V7	0.537331	0.288725	0.255883	0.128339654	0.592201	0.350702024	0.126597	0.085795293
V8	0.504154	0.270898	0.468191	0.234824005	0.269451	0.159569152	0.387128	0.262358194
SUM	**2.340599**	**1.257676401**	**2.569348**	**1.418671906**	**2.466419**	**1.460615798**	**2.403051**	**1.628557275**
MAP		**0.537331**		**0.552152494**		**0.592201**		**0.677704**

4.3 Summary of Results

We evaluated our results for eight vertices graph user feedback. The users were explained the concepts of hub and authority. They were told to rate each result node on the scale of 1-8. 1 is being the best rank and 8 being the worst. Table 4.3 shows average values for hub and authority rating for all queries. In all the queries we got relevant hub and authority rank of vertices. Also, it can be observed that the rating for hub and authority for a query are close to each other as following:

Table 4.3: sNorm (p) Hub and Authority Rank Values

Authority Rank	Hub Rank
0.079197	0.857286
0.677704	0.133291
0.148313	0.220567
0.20397	0.137262
0.372202	0.36882
0.40794	0.086956
0.126597	0.14802
0.387128	0.119599

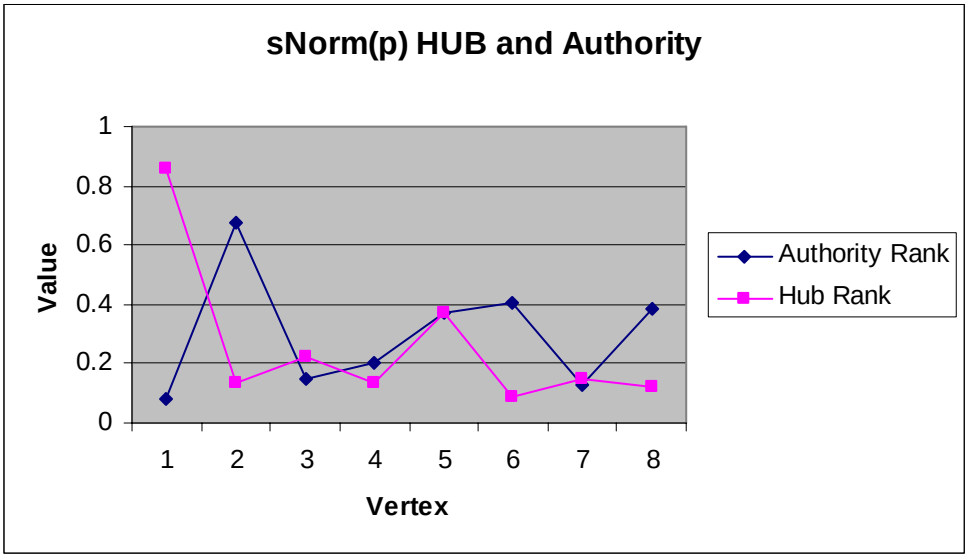

Figure 4.4 sNorm (p) Authorities and Hub Rank Graph

CHAPTER 5

CONCLUSION

The growth of the web is increase rapidly and accessibility has created the need for return the best result like query relevant pages on the top of list. Search engines are required to produce the web pages that the users are searching for within the top pages of results. So ranking of web pages becomes essential for this objective. Web Page ranking algorithm can be described as the use of hyperlink graph for the purpose of ranking web documents. Page Ranking operated under the assumption that given a collection of hyperlinked graph, that graph contains useful information about the authority of the pages, the main goal of Ranking is to extract this information and use this latent authority value and hub value to rank the web document. For the first objective we work with the hub and authority paradigm defined by Kleinberg. We proposed new ways for calculate the hub and authority weight of web pages.

HITS were implemented with different kinds of queries. It was observed that hyperlink based ranking is a powerful tool for Web based searching. We observed that root set provided good quality pages from the expanded root set topped the hub and authority. We have discussed a technique for locating high-quality information related to a broad search topic on the www, based on a structural analysis of the link graph. It is useful to highlight basic components of our approach.

1) The amount of relevant information is growing extremely rapidly, making it continually more difficult for individual users to filter the available resources for the broad query. To deal with this problem, it is for this purpose that we define a notion of authoritative sources, based on the link structure of the www.

2) At the same time, we infer global notions of structure without directly maintaining an index of the www or its link structure. We require only a basic interface to any of a number of standard www search engines, and use the techniques for producing enriched samples of www pages to determine notions of structure and quality that

make sense globally. This helps to deal with problems of scale in handling topics that have an enormous representation on the www. We began with the goal of discovering *authoritative pages*, but our approach in fact identifies a more complex pattern of social organization on the www, in which hub pages link densely to a set of thematically related authorities. This equilibrium between hubs and authorities is a phenomenon that recurs in the context of a wide variety of topics on the www. Measures of impact and influence in bibliometrics have typically lacked, and arguably not required, an analogous formulation of the role that hubs play; the www is very different from the scientific literature, and our framework seems appropriate as a model of the way in which authority is conferred in an environment such as the Web. In this we describe a large scale evaluation of the effectiveness of HITS, SALSA, sNorm(p) algorithm comparisons with other link based ranking algorithms. Evaluations are carried out with respect to a large number of human evaluated queries using major measures of effectiveness: MRR and MAP. We believe that the measurement presented in this thesis provide a solid evaluation of the best well known ranking algorithm.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley-Longman, (1999).

[2] Bernard J Jansen and Amanda Spink ,An Analysis of Web Documents Retrieved and Viewed, Pennsylvania State University ,Park PA 16802

[3] Y. Azar, A. Fiat, A. Karlin, F. McSherry, and J. Saia. Spectral analysis of data. In *Proceedings of the 33rd Symposium on Theory of Computing (STOC 2001)*, Hersonissos, Crete, Greece, (2001).

[4] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *Research and Development in Information Retrieval*, pages 104–111, (1998).

[5] Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas. Finding authorities and hubs from link structures on the World Wide Web. In *Proceedings of the 10th International World Wide Web Conference*, Hong Kong, (2001).

[6] Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of ACM (JASM)*, 46, (1999).

[7] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. In *Proceedings of the 7th International World Wide Web Conference*, Brisbane, Australia, (1998).

[8] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, (1998).

[9] Chakrabarti, S. Mining the Web: Discovering Knowledge from Hypertext Data. Morgan-Kauffman, (2002).

[10] M. Marchiori. The quest for correct information on Web: Hyper search engines. In *Proceedings of the 6th International World Wide Web Conference*, (1997).

[11] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. In *Proceedings of the 9th International World Wide Web Conference*, May 2000.

[12] Haveliwala, T. "Topic-sensitive pagerank." In proceedings of the Eleventh International World Wide Web Conference, Honolulu, Hawaii (2002).

[13] Haixuan Yang, Irwin King, "Predictive Ranking: A Novel Page Ranking Approach by Estimating the Web Structure". *WWW 2005,* May 10–14, Chiba, Japan. (2000).

[14] Henzinger, M. R. *"Web information retrieval an algorithmic perspective.*" In *Europea*n *Symposiu*m *o*n *Algorithm*s, pp. 1–8. (2000)

[15] Jaffery Dean, Monika R.HenZinge " Finding Related Pages in the World Wide Web" Compaq Western Research Laboratory (2000)

[16] Kumar, R., Raghavan, P., Rajagopalan, S., Sivakumar, D., Tomkins, A., and Upfal, E. *"The Web as a graph. In Proc. 19th ACM SIGACT-SIGMOD-AIGART Symposium.*" *Principle*s *o*f *Databas*e *Systems*, ACM Press, pp. 1–10.(2000)

[17] Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. “Crawling the Web for emerging cyber-communities”. Computer Networks (Amsterdam, Netherlands: 1999) 31, 11–16 (1999),

[18] Larson, R. Bibliometrics of the World Wide Web: An exploratory analysis of the intellectual structure of cyberspace. In *Annua*l *Meetin*g *o*f *th*e *America*n *Societ*y *fo*r *Informatio*n *Scienc*e (1996).

[19] Marc Najork, ”Comparing the Effectiveness of HITS and SALSA” *CIKM’07,* November 6–8, Lisboa, Portugal (2007)

[20] Marc Najork, Hugo Zaragoza, Michael Taylor,”HITS on the Web: How does it compare”?