

## CHAPTER 1

### CONTENT AND LINK STRUCTURE ANALYSIS FOR SEARCHING THE WEB

Kemal Efe, Vijay Raghavan, and Arun Lakhotia  
*Center for Advanced Computer Studies*  
*University of Louisiana, Lafayette LA 70504*

Finding relevant pages in response to a user query is a challenging task. Automated search engines that rely on keyword matching usually return too many low quality matches. Link analysis methods can substantially improve the search quality when they are combined with content analysis. This chapter surveys the mainstream work in this area.

#### 1. Introduction

Automated search engines continuously discover, index, and store information about web pages. When a user issues a query, this repository is searched to find a result set of most relevant pages. An ideal search scheme must satisfy two basic requirements: high recall, and high precision. Recall measures the ability of an algorithm to find as many relevant pages as possible. Precision measures the ability of an algorithm to reject as many nonrelevant pages as possible. An ideal search algorithm should find all of the relevant pages, rank them by relevance to the user query, and present a rank-ordered result to the user.

The earlier generations of search engines relied solely on keyword matching to perform the search. Unfortunately this approach didn't work very well. Too many nonrelevant pages were returned along with relevant ones, and their rankings rarely agreed with users' interests. Since user queries are short, usually consist of 2-3 words,<sup>25</sup> the problems associated with synonymy and polysemy make it particularly difficult to evaluate which pages will be of interest to a user.

The user is more likely to be interested in a page if it contains authoritative information on its subject and it is relevant to the user query. Authoritative pages are usually cited by others frequently, and the link

structure around these pages constitute certain special graph patterns. In modern search schemes a keyword matching algorithm initially identifies “potentially” relevant pages based on content analysis. Link analysis (often combined with further content analysis) is then applied to improve the search precision by focusing the search within the graph neighborhoods of these pages. This chapter provides a survey of such approaches. Other related tutorials can be found in.<sup>7,29,24,21,32,33,40</sup>

## 2. Intuitive Basis for Link Structure Analysis

A link on a web page provides valuable and readily available information. The person who created that link must think, or even recommend, that the cited page is related to the citing page. The term “collective intelligence” refers to an unorchestrated outcome from independent web page creators citing one another. Collective intelligence must surely play an important role in the formation of collective preferences which would manifest itself in the form of special graph patterns (or *signatures*) around authoritative sources in the web graph. By searching for (or computing) these patterns we could try to identify the authoritative pages.

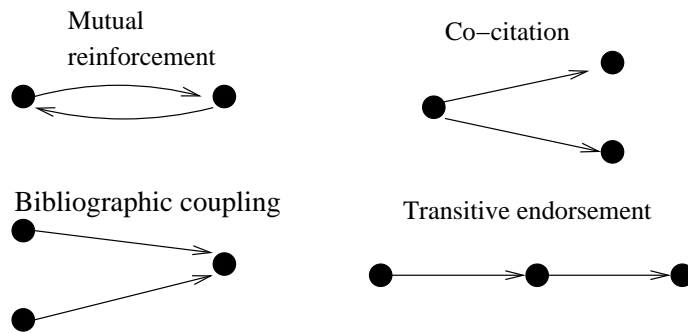


Fig. 1. Basic patterns formed by two directed edges.

To build an intuitive understanding of link structure analysis, consider Figure 1 which shows all possible connected graph patterns containing exactly two links. Each of these patterns has a corresponding interpretation: *Mutual reinforcement* occurs when two pages cite each other, reinforcing our intuition that the two pages are related to each other. *Co-citation* occurs when a page cites two other pages. In bibliometric studies<sup>52</sup> it has been observed that related papers are often cited together. Conversely, papers

that are cited together are likely to be related. *Bibliographic coupling* is the situation where two independent documents cite the same page. From this pattern we infer that the two pages are related to each other since they cite the same document. Finally *transitive endorsement* occurs when page  $p_1$  links to  $p_2$  which in turn links to  $p_3$ . Transitivity  $p_1$  may be considered to endorse  $p_3$ . However this is a weak endorsement and is rarely a sign of true relation between pages (generally the notion of “related to” is not transitive). We included it in Figure 1 only to cover all possible patterns involving exactly two links.

Statistical evidence observed in recent research validates these intuitive assertions.<sup>10,14,43</sup> However there is a significant percentage of cases when these assertions are violated. This is because human judgement applied to web citation is generally subjective and noisy. Also, if topic of discussion changes on a page, citations at different regions of a page may link to pages not related to each other. Because of these reasons we consider the above assertions as weak assertions. After all, for a graph containing only two links, it is hard to talk about collective intelligence.

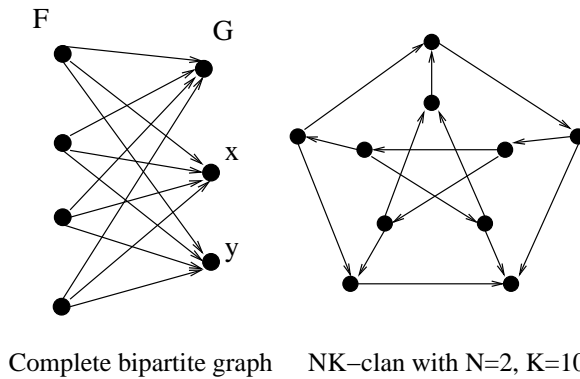


Fig. 2. Complex patterns that are indicative of related pages.

In the web graph, these basic structures can blend together to form more complex patterns of multiple links that further reinforce the implicated relationships among a set of web pages. For example consider the complete bipartite graph in Figure 2. In this graph the nodes are divided into two subsets  $F$  and  $G$  such that each node in  $F$  links to every node in  $G$ . Above we have seen that co-cited pages are likely to be related to each other. In the case of directed complete bipartite graph any two pages  $x, y$  in  $G$  are

co-cited by all the pages in  $F$ . That is, creators of all of the pages in  $F$  independently thought that  $x$  and  $y$  were related to each other. Similarly, by also considering the concept of bibliographic coupling, the aggregation of links in a complete bipartite graph constitute a strong evidence for the associated pages to be related to each other. Similar arguments can be applied to the NK-clan graph in Figure 2 also.

A number of researchers reported successful results from searching for various pre-defined, special patterns in the web graph by graph-theoretical methods. These included methods that search for directed complete bipartite graphs,<sup>30,47</sup> NK-Clan graphs,<sup>51</sup> and sets of pages that have more links to members than to non-members.<sup>19,5</sup> These approaches work well when searching for a cluster of related pages. Searching for well chosen patterns often achieve a high precision in the set of pages returned. However, these methods suffer from poor recall. From a graph-theoretical viewpoint, the problem of subgraph isomorphism is NP-complete, and there is no guarantee that all occurrences of the specified patterns will be found. Also, there may be high quality pages in other patterns that resemble but not necessarily identical to the specified pattern being searched. As a result, many highly authoritative pages may be missed. More flexible techniques are needed that are general enough to find clusters with known patterns even if the pattern lacks a few links, as well as detecting clusters with unknown patterns.

### 3. Link Structure Analysis

The more successful approaches for determining authoritative pages are based on computing, rather than searching for graph patterns. These include authority flow models and random walk models.

#### 3.1. Authority flow models

In this approach, we consider edge creation as a way of creating a channel through which authority can *flow* from the citing page to the cited page. The larger the number of citations received, the greater the authority flowing into a page. We can compute the authority ranks of pages iteratively as a function of the amount of authority flow they receive. Consider the graph in Figure 3 and its adjacency matrix  $A$ . Let  $r$  be the rank vector that represents authority ranks of all pages. The amount of authority flown into each page can be computed by

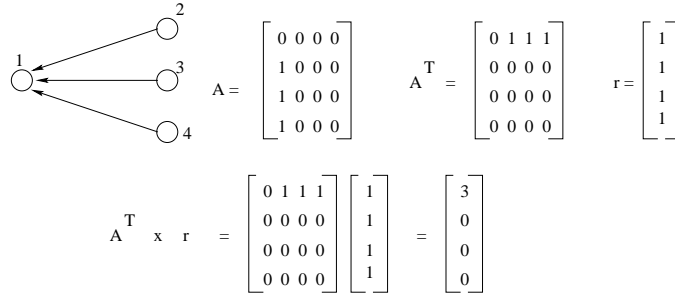


Fig. 3. Flow of authority into a page.

$$r = A^T \times r.$$

In this equation the amount of authority pumped out of a page depends on its rank. Since the rank changes after this computation we are interested in the final rank values after several iterations, provided the iterative computation

$$r(i+1) = A^T \times r(i) \quad (1)$$

converges as the iteration count  $i$  tends to infinity.

This computation assumes that a page  $q$  with authority rank  $r_q(i)$  at iteration  $i$  is able to pump all of its current authority weight at each of its outgoing link. We can modify this computation so that a page divides its authority equally between its outgoing links. Let  $x_q$  be the number of outgoing links on page  $q$ . Let  $W$  be the matrix obtained by dividing row  $q$  of  $A$  by  $x_q$  for all rows. The above equation becomes

$$r(i+1) = W^T \times r(i)$$

or, equivalently

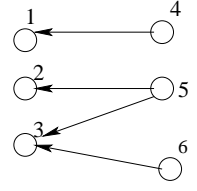
$$r_p(i+1) = \sum_{\forall q: q \rightarrow p} r_q(i)/x_q. \quad (2)$$

When this computation converges, the total authority pumped out of a page equals the total authority it receives. The final authority value is used as the rank of a page. A more elaborate version of this computation is used in Google search engine,<sup>4</sup> as we will see in section 5.1.

3.1.1. *Correlated Citations*

Equations 1 and 2 (or their real-life versions discussed in Section 5) don't have any built in mechanism to tell if an authoritative page belongs to a cluster of pages. An authoritative page on a subject is likely to be co-cited with other authoritative pages on the same subject, making it part of an authoritative group. Therefore it is reasonable to augment the authority rank of a page based on the degree that it is co-cited with other authorities.

To better explain this notion consider a directed graph  $G$  and its adjacency matrix  $A$  as shown in Figure 4. The matrix product  $A^T A$ , called the *co-citation matrix*,<sup>50</sup> has been known in bibliometric studies for a long time. Observe the following properties of the co-citation matrix.



$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$A^T \qquad \qquad A \qquad \qquad A^T A$

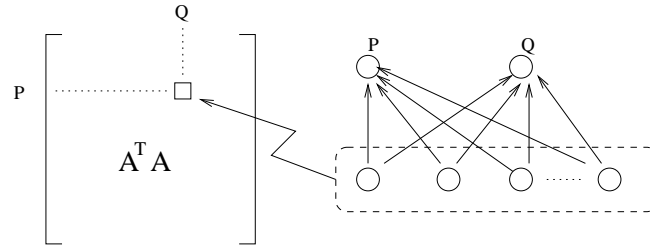


Fig. 4. Co-citation matrix and its properties.

- An entry  $(p,q)$  in  $A^T A$  represents the number of joint cocitations received by pages  $p$  and  $q$ ; i.e. among the pages that cite  $p$  the number that cite  $q$  also.
- The diagonal term in row  $p$  of  $A^T A$  is equal to the in-degree of page  $p$ ; i.e. the special case when  $q = p$ .
- Excluding the diagonal term, the sum of values in row  $p$  is the total number of times the page  $p$  is co-cited with other pages.

For the directed complete bipartite graph of Figure 2, all of the non-zero terms in a row of the co-citation matrix would be equal to the diagonal term. This is because any page that cites a page  $x$  in  $G$  also cites all of the other pages in  $G$ . Consequently, number of citations to a page  $x$  equals to its number of co-citations with  $y$  for each  $x, y$  in  $G$ . Now define a new iterative equation for the computation of authority ranks as follows:

$$a[i + 1] = (A^T A) \times a[i] \quad (3)$$

Due to the diagonal term in the co-citation matrix a page  $p$  receiving a large number of citations receives a large amount of authority inflow. Due to the non-diagonal terms, this authority inflow is strengthened by the degree that page  $p$  is co-cited with other pages. In fact, as the reader can easily verify, co-citations of a page can help improve its authority weight much more than the mere number of its citations.

### 3.1.2. Hubs Versus Authorities

If the concept of authority can be measured by in-degrees of pages, is there a symmetric case for out-degrees? Imagine for the sake of argument that surfers always follow the links in the backward direction. This is not possible physically, because web pages don't have reverse links to the pages citing them. But if it were possible to go in the reverse direction of links, which pages would be visited by the most number of surfers?

It turns out that this is a meaningful question with practical implications. While the reader may find it amusing to write the reverse equations paralleling those of 1-3 above, we will only consider the case for equation 3. In this case we have the matrix product  $AA^T$  which is called the *bibliographic coupling matrix*.<sup>26</sup> As illustrated in Figure 5 the bibliographic coupling matrix has the following properties:

- An entry  $(p,q)$  in  $AA^T$  represents the degree of bibliographic coupling of pages  $p$  and  $q$ ; i.e. the number of pages jointly cited by  $p$

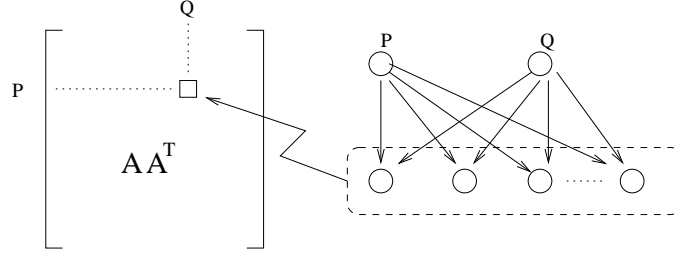


Fig. 5. Properties of the bibliographic coupling matrix.

and  $q$ .

- The diagonal term in row  $p$  of  $AA^T$  is equal to the out-degree of page  $p$ ; i.e. the special case when  $q = p$ .
- Excluding the diagonal term, the sum of values in row  $p$  gives the total number of times that pages cited by  $p$  are also cited by other pages.

These properties imply that if we define a new equation such as

$$h[i + 1] = (AA^T) \times h[i] \quad (4)$$

we compute the ability of a page to cite good sources. This ability has two components: Due to the diagonal term in  $AA^T$ , equation 4 gives higher weights to pages with larger out-degrees. Due to the non-diagonal terms the weight of a page is increased proportional to its ability to cite well-cited pages. This is precisely the ability needed in knowledgeable pages that are aware of good sources on the web.

In his paper<sup>27</sup> Kleinberg called these pages as “hubs.” Internet users are likely to be interested in both authority pages and hub pages. While a good authority page may provide valuable content a good hub page may lead the user to a variety of good authority pages to select from.

### 3.2. Random Walk Models

In a random walk model, the surfer can be seen as walking on the web graph, making random decisions about where to go next while at a web page. Some of the equations in the previous sections admit random walk interpretations while others don't. For example equation 1 does not admit a random walk interpretation since a page pumps out an amount of authority equal to its own out of every link it has. It would imply that a random surfer



splits itself into as many copies as the number of outgoing links at each new page so that each copy of the surfer takes a different path.

Equation 2 avoids this problem by dividing the authority weight of a page equally between the outgoing links. In this interpretation a surfer is required to choose one of the outgoing links of the current page to click. Consequently, the probability of leaving a page, which is equal to one, is the sum of probabilities for following different outgoing links. It follows, therefore, that the rank of a page represents the probability of reaching a page by following the links of the web graph.

To better explain, consider a user who clicks on the links at random. While at page  $q$  assume that the user clicks on the outgoing links with equal probability. If page  $q$  has  $x_q$  outgoing links, the probability that a user will click on any of the outgoing links is  $1/x_q$ . Then the probability that a page  $p$  is reached by following the links is just the summation term in equation 2. A modified version of this computation has been used in Google's PageRank algorithm.<sup>4</sup> In pageRank a surfer has two options: either click on one of the outgoing links or jump to an unrelated page. We discuss PageRank algorithm in more detail in Section 5.1.

For equations 3 and 4, a random walk model is not applicable. The situation here is similar to that of Equation 1 where a surfer would have to split itself into several copies at each new page. Population explosion of surfers makes this computation less stable than equation 2. Kleinberg's HITS algorithm, which uses equations similar to those of equations 3 and 4, normalizes the weight vectors  $a$  and  $h$  at each iteration to force the convergence.

The reader will notice that the computations of hubs and authorities in equations 3 and 4 are derived from equation 1. It is possible to derive these computations from equation 2 instead, which is the probabilistic version of equation 1. In particular, let  $W$  be the matrix obtained from the adjacency matrix  $A$  by dividing each non-zero term in a row by the number of non-zero terms in that row. Then the corresponding equations

$$a = W^T W a$$

$$h = W W^T h$$

represent a random surfer who is allowed a zig-zag walk going forward and backward on the links of the web graph.<sup>37</sup> Considering the directed bipartite graph of Figure 2, the equation  $a = W^T W a$  computes the probability that surfers reach an authority page from other authority pages after a two-step

zig-zag walk. The diagonal terms in  $W^T W$  measure the in-degrees of pages while non-diagonal terms measure the frequency that a page is co-cited with other pages. These co-citation edges serve to bring a surfer from one authority page to another. Similarly, the equation  $h = W W^T h$  represents corresponding calculations for hub pages. If the graph is not bipartite, the computed weights will be dominated by the in/out-degrees of pages. For pages that are in a bipartite component, the computed weights may be dominated by zig-zag walkers.

A more general model is obtained by defining different weights for diagonal terms and non-diagonal terms in the above computations. If the weights of non-diagonal terms are set as zero, then the authority ranks of pages depend on their in-degrees alone. Conversely, if the weights of diagonal terms are set as zero, the computed ranks depend on co-citation frequencies alone. Similar statements can be made for hub weights also. Without the influence of diagonal terms, these equations can be used for clustering web pages based on their membership status in bipartite subgraphs of the web graph. By adjusting these weights, the model can be made to behave more like the PageRank algorithm or more like the HITS algorithm.<sup>17</sup>

#### 4. Content Analysis Based Retrieval

Link analysis schemes can identify important pages in the web graph but they cannot tell if a page is relevant to the user query. This task is performed by content analysis. As we discuss in Section 5, most search schemes start with content analysis to determine a candidate subset of relevant pages, and apply link analysis in the graph neighborhood of these pages.

The most basic tool used in various content analysis tasks in information retrieval is a measure of similarity between two documents. In the case of web search the user query replaces one of the documents and the other document is a web page. There are crude but computationally efficient measures based on vector space models. These measures (see for example,<sup>49</sup> page 318) are all based on computing the inner product of term-frequency vectors  $x, y$  derived from two documents. A popular method is the Cosine similarity given by

$$S = \frac{\sum_{i=1}^t x_i \times y_i}{(\sum_{i=1}^t x_i^2 \times \sum_{i=1}^t y_i^2)^{1/2}}$$

where  $t$  is the length of the vectors  $x$  and  $y$ . This equation can be used for clustering documents on similar topics.

In this equation every term has equal weight, meaning that every word is assumed to have the same descriptive power in determining the topic of a document. When the user query consists only of a few words, as in a typical internet search query, inverse document frequency is a more informative measure of a term's value as a discriminator. Terms that do not occur with high frequency are highly useful for distinguishing documents in which they occur from those they do not occur. Let  $tf_j$  be the total number of occurrences of term  $T_j$  in  $N$  documents. Then the inverse document frequency  $idf_j$ , defined as  $idf_j = \log \frac{N}{df_j}$ , is an indicator of  $T_j$  as a document discriminator. Let  $tf_{i,j}$  denote the frequency of term  $j$  in document  $i$ . The product  $w_{i,j} = tf_{i,j} \log \frac{N}{df_j}$  can be used as a weight for term  $j$  in the above computation so that seldom used words are given higher weights if they are found on a given page.

Other variations of the above similarity measure have been defined. For example the Okapi measure<sup>23</sup> takes into account the length of a document in comparison to the average document length. Three Level Scoring (TLS)<sup>39</sup> is another variation where different weights computed for subqueries of different lengths are combined together. Cover Density Ranking (CRD)<sup>11</sup> is a method where a hit for the whole query has higher weight than a hit for any subset of query terms regardless of frequency of occurrence. In a recent comparison,<sup>38</sup> these four methods showed no significant performance difference when they were combined with an improved version of the HITS algorithm. However, these variations show improvements over the simple cosine similarity.<sup>49</sup>

Other sources of difficulties in relevance measuring of documents are synonymy and polysemy; many words can have similar meanings while a word can have several meanings. Synonymy causes many related pages to be missed while polysemy causes many unrelated pages to be declared as being authority on a subject. Latent Semantic Analysis (LSA)<sup>31,16</sup> and the Generalized Vector Space Model (GVSM)<sup>53</sup> are the two approaches that are frequently credited for successfully addressing these problems in information retrieval. In these models words are not treated as being independent from one another; their usage patterns are taken into account as well by computing an orthogonal vector of terms across documents. In a comparative study<sup>54</sup> these schemes were found to facilitate better document classifications, document search, and relevance ranking. It is also noted that the GVSM model is more efficient and more stable across various parameter values than the LSA model. A recent review of related indexing methods in information retrieval has been given in.<sup>48</sup>

Another related issue is *where* on the page to search for the user query terms. A web document has a title and a body, both of which contain potential sources of information. Research articles also contain an abstract as an identifiable section but beyond these most web pages lack any semantic structure to guide the search algorithms. Brin<sup>4</sup> notes that searching in the title alone returns astonishingly good matches. In addition the body text of the page can be mined for more detailed information about its relevance such as term frequencies for the query as well as its subqueries. Other useful information includes distance between subquery terms, the fonts, any highlighting used such as boldface or italic, and others.

Besides these, the anchor text associated with a link that points to a page provides quite accurate information about the content of a page. McBryan<sup>41</sup> was the first to observe that the anchor text often describes the content of a cited page better than the page itself. In the opinion of the person who created that link, the best source for the query in the anchor text is the cited page. Thus if the user query matches the anchor text, the pointed page must be an authoritative source for the user query. For some pages (e.g. the ones that mainly contain images, programs, databases) there may be no text in the page itself. In such cases, we are limited to the information in the title of a page and the anchor text associated with the links pointing to it.

## 5. Retrieval Techniques Combining Content and Link Structure Analysis

Google's PageRank algorithm and Kleinberg's HITS (Hyperlink Induced Topic Search) are two of the best known algorithms for topic search. Here we consider these algorithms and several of their variations proposed in the literature.

### 5.1. PageRank Algorithm

Google's web crawlers continuously search the web to collect new pages and update the old ones. These pages are stored in a data repository. The link structure of these pages are stored separately from other information to represent the web graph. This graph is used for computing page ranks by using the PageRank Equation off-line.

Google's PageRank algorithm considers a random surfer who has two options: either click on a forward link or jump to an unrelated page. Let  $d$  represent the probability that while at page  $q$  a surfer chooses to click on

one of the outgoing links instead of jumping to another page. Then  $(1 - d)$  represents the probability of jumping while at page  $q$ . If the surfers select the jump destinations with equal probability for all pages, then  $(1 - d)$  also represents the aggregate probability that a surfer reaches a given page  $p$  by jumping from any of the other pages. Accordingly the probability that Google's random surfer reaches a page  $p$  is given by the equation:

$$r_p = (1 - d) + d \sum_{\forall q: q \rightarrow p} r_q / x_q \quad (5)$$

where initial page ranks are chosen such that their sum equals to unity. By appropriately choosing  $d$  in the range  $0 < d < 1$  the above computation is guaranteed to converge because the parameter  $d$  dampens the authority inflow to keep it from growing indefinitely (other modifications to this computation for eliminating the effects of short loops are discussed in.<sup>45</sup>)

When a user issues a query, Google initially uses a keyword matching scheme to find a set of candidate pages. These pages are then ordered by their ranks before presenting to the user. This is not a simple case of sorting the pages by their ranks from equation 5. Rather, the rank of a page is a complex combination of weights and scores defined on various parameters, one of them being the static rank obtained from equation 5. The keyword frequency, position of keywords on the page, fonts, capitalization, the distance between component words of a multi-word query are examples of factors that contribute to the rank of a page.<sup>4</sup>

Google stores the anchor text associated with a link together with the cited page. During keyword search on a page, these pieces of text are also considered, and matches found in the anchor text contribute to the rank of the cited page. A hit on a page has different weights depending on whether the keyword is found on the title of the page, in the body text, or in the anchor text of an incoming link. Google also attributes different weights for links depending on who is citing a page. Citations by reputable sources such as Yahoo's directory service are weighted more heavily than others.

## 5.2. Topic Sensitive PageRank

In the original PageRank algorithm a single authority weight is computed for each page independent of any particular search query. To yield more accurate results, Haveliwala<sup>22</sup> proposed to compute a vector of page ranks for each page, corresponding to the importance of a page for each category in a preselected set of topics.

The main difference here is the way jump probabilities are computed. In equation 5 above, the probability of jumping is assumed to be same for every possible destination. In topic sensitive PageRank, jumping probabilities are computed for each topic.

Let there be  $N$  pages in total, of which  $T_j$  pages belong to topic  $j$ . In equation 5, the probability that a surfer jumps to page  $p$  is equal to  $1/N$ . In topic sensitive PageRank this probability is computed as  $1/T_j$  if page  $p$  is in topic  $j$ . Otherwise the probability of jumping to page  $p$  is zero for category  $j$ . The rest of the ranking equation is similar to the PageRank algorithm.

By using topic-dependent jumping probabilities, different page ranks are computed for each page, one rank value for each topic. When a user issues a query, all topics represented in the query are identified. The rank of a page is computed as the sum of its category ranks for each of these topics.

### 5.3. *HITS Algorithm*

Kleinberg's HITS algorithm tries to identify hubs and authorities by using the equations:

$$h = Aa \quad (6)$$

$$a = A^T h \quad (7)$$

which are equivalent to equations 3 and 4. Hub and authority vectors are normalized before every iteration such that squares of their respective weights sum to unity. Kleinberg proved that the  $a$  vector converges to the principal eigenvector of  $A^T A$  and the  $h$  vector converges to the principal eigenvector of  $AA^T$ . At steady state, pages on a common topic and with the largest hub and authority weights are highly likely to represent pages of a graph resembling the directed bipartite graph in Figure 2.

This algorithm has two major steps: sampling and weight-propagation. The sampling step uses a keyword-based search to select around 200 pages by using one of the commercially available search engines. This set of pages is called the *root set*. This root-set is then expanded into a *base set* by adding any page on the web that has a link to/from a page in the root set. (These same steps were used earlier in WebQuery system<sup>6</sup> where authors called these sets of pages as "hit set" and "complete neighbor set." Web-Query ranks pages in the complete neighbor set in decreasing order of their connectivity, i.e. the number of incoming plus outgoing links). The base set

typically contains a few thousand pages. The pages in the base set may or may not constitute a connected graph but at least it has a large connected component.<sup>28</sup>

The weight-propagation step of HITS algorithm computes the hub weights and authority weights for the pages in the base set by using equations 6 and 7. The output of the algorithm is a short list of pages with the largest hub weights and a list of pages with the largest authority weights. The implementation typically outputs 10 from each group as the final list. Gibson et al.<sup>20</sup> reported that HITS algorithm is very effective in finding clusters of related pages.

The work of Bharat and Henzinger<sup>2</sup> showed that a straight implementation of the HITS algorithm does not work well for topic search. More successful implementations depended on using additional heuristics to tackle the observed causes of poor performance.<sup>2,9</sup> For example Chakrabarti et al.<sup>9</sup> observed that when the topic of discussion varies on different parts of a page, the outgoing links also point to different topics. A page with a large out-degree will award the same authority weight to each page with which it links on the subject of the user query. However, these cited pages may not even be on the same topic. To solve this problem they used a page splitting heuristic. If large documents are split into several small documents, there is a smaller probability for the cited pages to be unrelated to one another. The authors reported significantly improved results with this heuristic.

Li et al.<sup>38</sup> present another improvement of the HITS algorithm where hub weights of pages are increased depending on their authority weights. A hub page with many incoming links has a higher hub weight than a hub page with fewer or no incoming links. This is intuitively appealing because a good hub is likely to be cited, i.e. it must a good authority at being a hub.

Another problem observed with the HITS algorithm is the *Tightly Knit Community* (TKC) effect. Examples include the Nebraska tourist information page being returned in response to a query for skiing in Nebraska,<sup>9</sup> and pages on “computational linguistics” dominating the returned pages when searching for authoritative pages on “linguistics.”<sup>20</sup> In both cases HITS has converged to regions of the web graph with the considerably greater density of linkage.

Other researchers<sup>12,13</sup> observed that the TKC effect of HITS algorithm is related to its convergence to the principal eigenvectors. Ideally the rank of a page in the root set should reflect the likelihood of it being cited in its community. In HITS algorithm a popular page would be deemed unimpor-

tant if it is part of a smaller community. For example the root set returned in response to the query “jaguar” may contain pages on the automobile, on the animal, on the Atari Jaguar product, or anything else that has the word “jaguar” in its name. The set of pages represented in the principal eigenvector would be dominated by one of these categories completely ignoring other pages that are rightfully popular in their respective communities.

An improvement over the HITS algorithm eliminating its TKC effect should then manifest itself in its ability to include popular pages from each community in the same base set. Cohn and Chang proposed a probabilistic model of citations called the PHITS algorithm where the rank of a page is supposed to represent the probability of its citation within its own community rather than within the entire base set. Borodin et. al.<sup>3</sup> present comparisons of several variations of the HITS algorithm. Interesting observations are reported about differences in the sets of pages returned by different variations of HITS algorithm.

In another implementation, HITS algorithm was used for finding pages related to a given web page.<sup>15</sup> Here the algorithm starts with a seed URL and finds pages that are related to it. This is similar to the “What’s Related” facility in Netscape.<sup>44</sup> In this implementation the base set required by the HITS algorithm is obtained from the seed URL by including its parents (the pages that link to it), its children (the pages that it links to), children of its parents, and parents of its children. At the end of the iterative computations the algorithm outputs 10 of the highest ranked authority pages. The authors found that instead of a full implementation of the HITS algorithm, a simpler approach performs much better: Given the seed page, find the pages that link to it, and then determine “who else” they link to. The algorithm outputs 10 of the pages that are most frequently co-cited with the seed URL.

A search engine that needs to respond to thousands of queries per second cannot be expected to run complex content analysis algorithms. For this reason, simple ideas that work are immensely valuable. One such idea first introduced by McBryan<sup>41</sup> is to perform limited content analysis in the anchor text of links in the citing page. This idea has sound intuitive basis since the anchor text complements the citation. Creator of that link says: “here is the most relevant page for the query in the anchor text.” As mentioned in section 5.1, PageRank algorithm makes use of this concept.

In the CLEVER project,<sup>8,9,10</sup> this idea was implemented by comparing the user query against the text around the link. A relevance weight is computed for each link. The weight  $w(p, q)$  is just the number of matches found



on page  $p$  around the link pointing to  $q$ . This yields a modified adjacency matrix where the entries are computed as  $x(p, q) = 1 + w(p, q)$ . Thus if target page is not related to the search topic, the anchor text should assign a small weight to the link. Small link weights work as filters that block transfer of authority toward unrelated pages. The authors report that the results of the CLEVER algorithm produced substantially improved results over the HITS algorithm. In fact, in user evaluations, pages returned by this implementation achieved higher approval than the manually compiled Yahoo directory.

Another approach<sup>1</sup> focused on controlling the influence of pages rather than the individual links in them. Since users only type a few key words, it is difficult to compute a meaningful similarity measure between the key words and web documents. Thus the researchers constructed a query document by combining together the first 1000 words from each document in the root set. Then they computed the cosine-normalized similarity of this reference page with all the pages in the base set. This computation yielded the relevance weights of different documents. These weights are used to dampen the hub weights and authority weights of pages before each iteration is started. Authority weight of a page  $p$  is computed as  $a_p = a_p \times r_p$  where  $r_p$  is the relevance weight of page  $p$ . This algorithm effectively weeds-out irrelevant pages in the base set and adjusts the weight of other pages depending on their similarity with the reference page. The result was much better than a straight implementation of HITS algorithm.

## 6. Conclusions and Future Directions

Due to space limitations, much of the ongoing works in related areas are left out of the scope of this tutorial. Here we briefly mention some of the potentially useful areas that can further improve the existing search algorithms. For example, more accurate mathematical models may be obtained by using the observed frequencies of link usages instead of treating all outgoing links of a page with equal weight as in the PageRank algorithm or in the topic sensitive PageRank. Some work in modeling a non-random surfer has been reported.<sup>46</sup> More research in this direction could focus on efficient implementation of such a non-random surfer model.

Other related research focuses on utilizing user feedback to fine-tune search parameters. Fundamental techniques for relevance feedback have been discussed in.<sup>21,48</sup> Independently, researchers at the NEC Research Institute have developed several techniques for representing and utilizing user

context to guide the search schemes.<sup>34,36</sup> These schemes are based on tailoring and augmenting the query terms to improve keyword matches. Other work involves creating metasearch engines on the fly to determine the importance of a page depending on the number of search engines containing it along with its rank in each.<sup>35,42</sup>

Another significant development is the ongoing work in XML (Extensible Markup Language) standards. A major difficulty in web search is extracting semantic structure in existing web documents. Web pages written in HTML only describe how documents should look on the computer screen. The markup tags in XML specify the meaning of each attribute in the data and facilitate searching for specific information in a document.<sup>55,56</sup> The ongoing work on XML<sup>18</sup> is aimed at providing web page designers a suite of tools to develop semantically meaningful hyperlinked text. As a whole, XML's set of tools allow creating, organizing, indexing, linking, and querying data on the web. Future work can focus on more effective content analysis algorithms in XML pages. More information about XML is available online at [www.w3.org/XML](http://www.w3.org/XML).

### Acknowledgments

This research was funded by Louisiana State's Information Technology Initiative.

### References

1. Krishna Bharat and Andrei Z. Broder. "A technique for measuring the relative size and overlap of public web search engines" in World-Wide Web'98 (WWW7), Brisbane, Australia, 1998.
2. Krishna Bharat and Monika Henzinger. "Improved Algorithms for Topic Distillation in a Hyperlinked Environment" 21st ACM SIGIR conference on Research and Development in Information Retrieval, pp. 469-477, 1998.
3. A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas, "Finding Authorities and Hubs From Link Structures on the World Wide Web," Proc. 10th International Conf. WWW, 2001.
4. Sergey Brin, and Larry page. "The Anatomy of a Large Scale Hypertextual Web Search Engine" In Proc. of WWW7, Brisbane, Australia, April 1998.
5. M. Brinkmeier, "Communities in Graphs," Online document at [www.nj.nec.com/brinkmeier02communities.html](http://www.nj.nec.com/brinkmeier02communities.html).
6. S. J. Carriere, and R. Kazman, "WebQuery: Searching and Visualizing the Web Through Connectivity," *Computer Networks and ISDN Systems*, 29: 1257-1267, 1997.
7. Soumen Chakrabarti. "Recent results in automatic Web resource discovery", ACM computing survey, 1999.

8. Soumen Chakrabarti, Byron E. Dom, Prabhakar Raghavan, Sridhar Rajagopalan, David Gibson and Jon M. Kleinberg. "Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated Text" in Proceedings of World-Wide Web'98 (WWW7), Brisbane, Australia, pp. 65-74, April 1998.
9. Soumen Chakrabarti, Byron E. Dom, David Gibson, Jon Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins. "Mining the Link Structure of the World Wide Web" IEEE Computer, Vol.32 No.8, August 1999.
10. Soumen Chakrabarti, M. M. Joshi, K. Punera, and D. Pennock, "The Structure of Broad Topics on the Web," www 2002.
11. C. L. A. Clarke, G. V. Cormack, E. A. Tudhope, "Relevance Ranking for One to Three Term Queries," *Information Processing and Management*, vol 36, 2000, pp. 291-311.
12. D. Cohn and H. Chang, "Learning to Probabilistically Identify Authoritative Documents," Proc. 17th International Conference on Machine Learning, Stanford University, 2000, pp. 167-174.
13. D. Cohn and T. Hofman, "The Missing Link - A Probabilistic Model of Document Content and Hypertext Connectivity," in T. Leen et al., eds., *Advances in Neural Information Processing Systems*, Vol 13, 2001.
14. Brian D. Davison, "Topical Locality in the Web," Proceedings of the 23rd Annual International Conference on Research and Development in Information Retrieval (SIGIR 2000), Athens, Greece, July 24-28, 2000, pp. 272-279.
15. Jeffrey Dean, and Monika R. Henzinger. "Finding related Pages in the World Wide Web" In Proc. WWW-8, 1999.
16. Scott Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman, "Indexing by Latent Semantic Analysis," *Journal of the Society of Information Science*, 41(6):391-406, 1990.
17. M. Diligenti and M. Gori and M. Maggini, "Web page scoring systems for horizontal and vertical search", In Proceedings of the 11th World Wide Web Conference (WWW11) 1-7 May 2002, Honolulu (USA), 2002.
18. "Extensible Markup Language (XML)," online document at <http://www.w3.org/XML>.
19. G. W. Flake, S. Lawrence, and C. L. Giles, "Efficient Identification of Web Communities," Proc. 6th Intn'l Conf. on Knowledge Discovery and Data Mining (ACM SIGKDD-2000), pp. 150-160.
20. David Gibson, Jon Kleinberg, Prabhakar Raghavan. "Inferring Web Communities from Link Topology" Proc. 9th ACM Conference on Hypertext and HyperMedia, 1998.
21. Venkat N. Gudivada, Vijay V. Raghavan, William I. Grosky, Rajesh Kananagottu. "Information retrieval on the world wide web," *EEE Internet Computing*, Vol. 1, No. 5, 1997, pp. 58-68.
22. T. H. Haveliwala, "Topic Sensitive PageRank," Proc. WWW 2002.
23. D. Hawking, P. Bailey, N. Craswell, "ACSys Trec-8 Experiments," Proc. TREC-8 NIST Special Publication, pp. 500-246, 1999.
24. Wen-Chen Hu, "World Wide Web Search Technologies," chapter of the book, Shi Nansi (Ed.), 'Architectural Issues of Web-Enabled Electronic Business',

- Idea Group Publishing, <http://citeseer.nj.nec.com/461532.html>
25. J. Jansen, A. Spink, J. Batesman, and T. Saracevic, "Real Life Information Retrieval: A Study of User Queries on the Web, SIGIR Forum, Vol 32, No 1, 1998, pp. 5-17.
  26. M. M. Kessler, "Bibliographic coupling between scientific papers," *American Documentation*, 14:10-25, 1963.
  27. Jon M. Kleinberg. "Authoritative sources in a hyperlinked environment" in Proceedings of ACM-SIAM Symposium on Discrete Algorithms, pp. 668-677, January 1998.
  28. Jon M. Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan and Andrew S. Tomkins. "The Web as a graph: measurements, models and methods" Proceedings of the 5th International Computing and combinatorics Conference, 1999.
  29. R. Kosala and H. Blockeel, "Web Mining Research: A Survey," SIGKDD Explorations -Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining 2 (2000), no. 1, pp. 1-15, Special Issue on "Internet Mining."
  30. Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins. "Trawling the web for emerging cyber-communities", Proc. 8th International World Wide Web Conference, WWW8, 1999.
  31. Landauer, T.K. (2002). Applications of Latent Semantic Analysis , 24th Annual Meeting of the Cognitive Science Society, August 9th 2002.
  32. S. Lawrence, and C. L. Giles, "Searching the World Wide Web," *Science*, 280:98-100, 1998.
  33. S. Lawrence, and C. L. Giles, "Searching the Web: General and Scientific Information Access," *IEEE Communications*, 37 (1):116-122, 1999.
  34. S. Lawrence and C. L. Giles, "Context and Page Analysis for Improved Web Search," *IEEE Internet Computing*, July-August 1998, pp. 38-46.
  35. S. Lawrence and G. L. Giles, "Inquirus, the NECI Meta Search Engine," 7th int'l WWW Conference, Brisbane, Australia, pp. 95-105, 1998.
  36. S. Lawrence, "Context in Web Search," *IEEE Data Engineering Bulletin*, vol. 23, No. 3, pp. 25-32, 2000.
  37. R. Lempel and S. Moran, "The Stochastic Approach for Link-Structure Analysis (SALSA) and the TKC Effect," Proc. 9th International World Wide Web Conference, May 2000.
  38. L. Li, Y. Shang, W. Zhang, "Improvement of HITS-Based Algorithms on Web Documents," WWW Conf. 2002, pp. 527-535.
  39. L. Li and Y. Shang, "A New Statistical Method for Evaluating Search Engines," Proc. IEEE 12th Intn'l Conf. Tools With Artificial Intelligence, Vancouver, British Columbia, 2000.
  40. H. Lu and L. Geng, "Integrating Database and World Wide Web Technologies," *World Wide Web*, Vol. 1, No. 2, pp. 73-86, 1998.
  41. O. A. McBryan, "GENVL and WWW" Tools for Taming the Web," Proc. 1st Int'l conf. World-Wide Web, 1994.
  42. W. Meng, C. Yu, and K. Liu, "Building Efficient and Effective Metasearch Engines," *ACM Computing Surveys*, 34(1):48-84, 2002.

43. D. Mladenic and M. Grobelnik, "Predicting Content from Hyperlinks," Proceedings of the ICML-99 Workshop on Machine Learning in Text Data Analysis, J. Stephan Institute, Ljubljana, Slovenia, 1999, pp. 19-24.
44. Netscape communications Corporation, on-line document at <http://home.netscape.com/escapes/related/faq.html#o7>
45. L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," online at [cite-seer.nj.nec.com/page98pagerank.html](http://citeseer.nj.nec.com/page98pagerank.html).
46. J. Pitkow and P. Prolli, "Mining Longest Repeating Subsequences to Predict World Wide Web Surfing," Proc. USITS'99, the 2nd USENIX Symposium on Internet Technologies and Systems, Boulder, Colorado, October 11-14, 1999.
47. P. K. Reddy and M. Kitsuregawa, "An Approach to Relate the Web Communities Through Bipartite Graphs," WISE 1: 301-310, 2001.
48. V. V. Raghavan, V. N. Gudivada, Z. Wu, and W. I. Grosky, Information Retrieval, In "The Practical Handbook of Internet Computing" (Ed. M. Singh), CRC Press (to appear).
49. Gerard Salton. "Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer" Addison Wesley Publishing Co., Reading, MA, 1989.
50. H. Small, "Co-citation in Scientific Literature: A New Measure of the Relationship Between Two Documents," *Journal of the American Society for Information Sciences*, 24:265-269, 1973.
51. Loren Terveen and Will Hill. "Finding and Visualizing Inter-site Clan Graphs" Proceedings of CHI 98: 448-455, Los Angeles, CA.
52. H. D. White, K. W. McCain. "Bibliometrics" in Annual Review of Information Science and Technology, Elsevier, pp. 119-186, 1989.
53. S. K. M. Wong, W. Ziarko, V. V. Raghavan, and P. C. N. Wong, "On Modeling of Information Retrieval Concepts in Vector Space," *ACM Transactions of Database Systems*, no. 2, 1987, pp. 299-321.
54. Y. Yang, J. G. Carbonell, R. D. Brown, and R. E. Frederkin, "Translingual Information Retrieval: Learning from Bilingual Corpora," *Artificial Intelligence Journal Special Issue: Best of IJCAI-97*, 1998, pp. 323-345.
55. J. Yoon, V. V. Raghavan, and V. Chakilam. Bitmap indexing based clustering and retrieval of XML documents. In Proceedings of ACM SIGIR Workshop on Mathematical/Formal Methods in Information Retrieval, New Orleans, LA, Sept. 2001.
56. J. Yoon, V. V. Raghavan, V. Chakilam, and L. Kerschberg. Bitcube: A three-dimensional bitmap indexing for XML documents. *J. of Intelligent Information Systems*, 17(2/3):241-254, Nov. 2001.