7.2 PageRank and HITS

Two algorithms for ranking Web pages based on links, PageRank and HITS (hyperlink induced topic search), were developed around the fall of 1996 at Stanford University by Larry Page[1] and Sergey Brin, and at IBM Almaden by Jon Kleinberg. Both sought to remedy the "abundance problem" inherent in broad queries, supplementing precision with notions related to prestige in social network analysis.

In PageRank, each page on the Web has a measure of *prestige* that is independent of any information need or query. Roughly speaking, the prestige of a page is proportional to the sum of the prestige scores of pages linking to it. In HITS, a query is used to select a subgraph from the Web. From this subgraph, two kinds of nodes are identified: *authoritative* pages to which many pages link, and *hub* pages that consist of comprehensive collections of links to valuable pages on the subject.

Although there are technical differences, all three measures are defined recursively: prestige of a node depends on the prestige of other nodes, and the measure of being a good hub depends on how good neighboring nodes are as authorities (and vice versa). Both procedures involve computing eigenvectors for the adjacency matrix, or a matrix derived thereof, of the Web or a suitably relevant subgraph of the Web. In this section we will study these algorithms and take a careful look at their strengths and weaknesses.

7.2.1 PageRank

Assume for the moment that the Web graph is strongly connected—that is, from any node u there is a directed path to node v. (It is not; we come back to this issue a little later.) Consider a Web surfer clicking on hyperlinks forever, picking a link uniformly at random on each page to move on to the next page. Suppose the surfer starts from a random node in accordance with a distribution $\vec{p}_0$, with probability $p_0[u]$ of starting from node u, where $\sum_u p_0[u] = 1$. Let the adjacency matrix of the Web be E, where $E[u, v] = 1$ if there is a hyperlink $(u, v) \in E$, and zero otherwise. We overload E to denote both the edge set and its corresponding matrix.

After clicking once, what is the probability $p_1[v]$ that the surfer is on page v? To get to v, the surfer must have been at some node u with a link to v in the previous step, and then clicked on the specific link that took her from u to v.

1. PageRank is named after Larry Page, a founder of Google.

Given E, the out-degree of node u is given simply by

$$N_u = \sum_v E[u, v] \tag{7.3}$$

or the sum of the uth row of E. Assuming parallel edges (multiple links from u to v) are disallowed, the probability of the latter event given the former (i.e., being at u) is just $1/N_u$. Combining,

$$p_1[v] = \sum_{(u,v)\in E} \frac{p_0[u]}{N_u} \tag{7.4}$$

Let us derive a matrix L from E by normalizing all row-sums to one, that is,

$$L[u, v] = \frac{E[u, v]}{\sum_\beta E[u, \beta]} = \frac{E[u, v]}{N_u} \tag{7.5}$$

With L defined as above, Equation (7.4) can be recast as

$$p_1[v] = \sum_u L[u, v] p_0[u] \tag{7.6}$$

or

$$\mathbf{p}_1 = L^T \mathbf{p}_0 \tag{7.7}$$

The form of Equation (7.7) is identical to that of Equation (7.1) except for the edge weights used to normalize the degree. After the ith step, we will get

$$\mathbf{p}_{i+1} = L^T \mathbf{p}_i \tag{7.8}$$

We will initially assume that nodes with no outlinks have been removed a priori. If E and therefore L are *irreducible* (i.e., there is a directed path from every node to every other node) and *aperiodic* (i.e., for all u, v, there are paths with all possible number of links on them, except for a finite set of path lengths that may be missing), the sequence $(\mathbf{p}_i)$, $i = 0, 1, 2, \ldots$ will converge to the principal eigenvector of L^T, that is, a solution to the matrix equation $\mathbf{p} = L^T \mathbf{p}$, also called the *stationary distribution* of L. The prestige of node u, denoted $p[u]$, is also called its PageRank. Note that the stationary distribution is independent of $\mathbf{p}_0$.

For an infinitely long trip made by the surfer, the converged value of $\mathbf{p}$ is simply the relative rate at that the surfer hits each page. There is a close correspondence to the result of the "aimless surfer" model above and the notion of prestige in bibliometry: a page v has high prestige if the visit rate is high, which happens if there are many neighbors u with high visit rates leading to v.

The simple surfing model above does not quite suffice, because the Web graph is not strongly connected and aperiodic. An analysis of a significant portion of the Web graph (a few hundred million nodes) in 2000 showed that it is not strongly connected as a whole [28]. Only a fourth of the graph is strongly connected. Obviously, there are many pages without any outlinks, as well as directed paths leading into a cycle, where the walk could get trapped.

A simple fix is to insert fake, low-probability transitions all over the place. In the new graph, the surfer first makes a two-way choice at each node:

1. With probability d, the surfer jumps to a random page on the Web.
2. With probability $1 - d$, the surfer decides to choose, uniformly at random, an out-neighbor of the current node as before.

d is a tuned constant, usually chosen between 0.1 and 0.2. Because of the random jump, Equation(7.7) changes to

$$\mathbf{p}_{i+1} = (1-d)L^T\mathbf{p}_i + d\begin{pmatrix} 1/N & \cdots & 1/N \\ \vdots & \ddots & \vdots \\ 1/N & \cdots & 1/N \end{pmatrix}\mathbf{p}_i$$

$$= \left((1-d)L^T + \frac{d}{N}\boxed{\mathbf{1}_N}\right)\mathbf{p}_i$$

simplifying notation,

$$= (1-d)L^T\mathbf{p}_i + \frac{d}{N}(1,\ldots,1)^T \tag{7.9}$$

where N is the number of nodes in the graph. $p[u]$ is the PageRank of node u. Given the large number of edges in E, direct solution of the eigen system is usually not feasible. A common approach is to use *power iterations* [91], which involves picking an arbitrary nonzero $\mathbf{p}_0$ (often with all components set to $1/N$), repeated multiplication by $(1-d)L^T + \frac{d}{N}\boxed{\mathbf{1}_N}$, and intermittent scaling $|\mathbf{p}_i|$ to one. Since notions of popularity and prestige are at best noisy, numeric convergence is usually not necessary in practice, and the iterations can be terminated as soon as there is relative stability in the ordering of the set of prestige scores.

There are two ways to handle nodes with no outlink. You can jump with probability one in such cases, or you can first preprocess the graph, iteratively removing all nodes with an out-degree of zero (removing some nodes may lead to the removal of more nodes), computing the PageRanks of surviving nodes, and propagating the scores to the nodes eliminated during the preprocessing step.

In this application, the exact values of $\mathbf{p}_i$ are not as important as the ranking they induce on the pages. This means that we can stop the iterations fairly quickly. Page et al. [169] report acceptable convergence ranks in 52 iterations for a crawl with 322 million links.

In Google, the crawled graph is first used to precompute and store the PageRank of each page. Note that the PageRank is independent of any query or textual content. When a query is submitted, a text index is used to first make a selection of possible response pages. Then an undisclosed ranking scheme that combines PageRank with textual match is used to produce a final ordering of response URLs. All this makes Google comparable in speed, at query time, to conventional text-based search engines.

PageRank is an important ranking mechanism at the heart of Google, but it is not the only one: keywords, phrase matches, and match proximity are also taken into account, as is anchor text on pages linking to a given page. Search Engine Watch (*www.searchenginewatch.com/*) reports that during some weeks in 1999, Google's top hit to the query "more evil than Satan" returned *www.microsoft.com/*, probably because of anchor text spamming. This embarrassment was fixed within a few weeks. The next incident occurred around November 2000, when Google's top response to a rather offensive query was *www.georgewbushstore.com/*. This was traced to *www.hugedisk.com/*, which hosted a page that had the offensive query words as anchor text for a hyperlink to *www.georgewbushstore.com/*.

Although the details of Google's combined ranking strategy are unpublished, such anecdotes suggest that the combined ranking strategy is tuned using many empirical parameters and checked for problems using human effort and regression testing. The strongest criticism of PageRank is that it defines prestige via a single random walk uninfluenced by a specific query. A related criticism is of the artificial decoupling between relevance and quality, and the ad hoc manner in which the two are brought together at query time, for the sake of efficiency.

7.2.2 HITS

In hyperlink induced topic search (HITS), proposed by Kleinberg [122], a query-dependent graph is chosen for analysis, in contrast to PageRank. Specifically, the query q is sent to a standard IR system to collect what is called a *root set* R of nodes in the Web graph. For reasons to be explained shortly, any node u that neighbors any $r \in R$ via an inbound or outbound edge—that is, $(u, r) \in E$ or $(r, u) \in E$—is included as well (E is the edge set for the Web). The additional

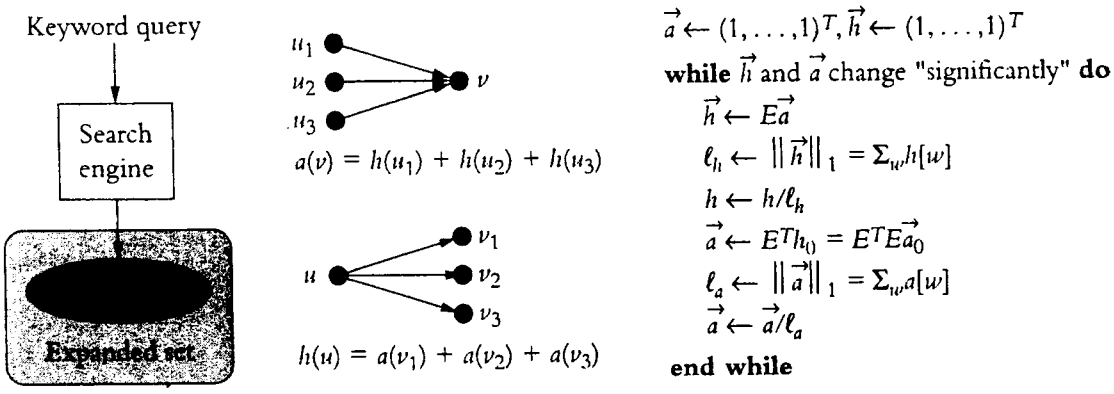

FIGURE 7.2 The HITS algorithm. ℓ_h and ℓ_a are L_1 vector norms.

nodes constitute the *expanded set* and, together with the root set, form the *base set* V_q. Edges that connect nodes from the same host are now eliminated because they are considered "navigational" or "nepotistic" (also see Section 7.3.1). Let us call the remaining edges E_q. We thus construct the query-specific graph $G_q = (V_q, E_q)$ (see Figure 7.2). (I will drop the subscript q where clear from context.)

Kleinberg observed that as in academic literature, where some publications (typically in conferences) initiate new ideas and others consolidate and survey significant research (typically in journals or books), the Web includes two flavors of prominent or popular pages: *authorities,* which contain definitive high-quality information, and *hubs,* which are comprehensive lists of links to authorities. Every page is, to an extent, both a hub and an authority, but these properties are graded. Thus, every page u has two distinct measures of merit, its hub score $h[u]$ and its authority score $a[u]$. Collectively, the scores over all the nodes in G_q are written as vectors $\vec{a}$ and $\vec{h}$, with the uth vector component giving the score for node u.

As in the case of PageRank, the quantitative definitions of hub and authority scores are recursive. The authority score of a page is proportional to the sum of hub scores of pages linking to it, and conversely, its hub score is proportional to the authority scores of the pages to which it links. In matrix notation, this translates to the following pair of equations:

$$\vec{a} = E^T\vec{h} \tag{7.10}$$

$$\vec{h} = E\vec{a} \tag{7.11}$$

Again, power iterations may be used to solve this system of equations iteratively, as shown in the pseudocode in Figure 7.2. When $\vec{a}$ attains convergence, it will be

the principal eigenvector of $E^T E$. $\vec{h}$ will converge to the principal eigenvector of EE^T. Typically, runs with several thousand nodes and links "converge" in 20 to 30 iterations, in the sense that the rankings of hubs and authorities stabilize.

Summarizing, the main steps in HITS are

1. Send query to a text-based IR system and obtain the root set.
2. Expand the root set by radius one to obtain an expanded graph.
3. Run power iterations on the hub and authority scores together.
4. Report top-ranking authorities and hubs.

The entire process is generically called *topic distillation*. User studies [40] have shown that reporting hubs is useful over and above reporting authorities, because they provide useful annotations and starting points for users to start exploring a topic.

HITS cannot precompute hub and authority scores because the graph G_q can only be computed after query "q" is known. This is both a strength and a weakness. The model of conferring authority through linkage clearly makes more sense when restricted to a subgraph of the Web that is relevant to a query, and therefore we expect HITS to need fewer ranking tweaks than PageRank once the scores are computed. Haveliwala [98] has proposed to precompute a few topic-specific PageRanks to address this limitation. The flip side is that HITS has to undertake an eigenvector computation per query.

Bipartite subgraphs are key to the reinforcement process in HITS. Consider Figure 7.2. If in some transfer step node v_1 collects a large authority score, in the next reverse transfer, the hub u will collect a large hub score, which will then diffuse to siblings v_2 and v_3 of node v_1. Many times, such diffusion is crucial to the success of HITS, but it can be overdone. Some causes and remedies are discussed in Sections 7.3 and 7.4.

The key distinction of HITS from PageRank is the modeling of hubs. PageRank has no notion of a hub, but (Google) users seem not to regard this as a major handicap to searching, probably because on the Web, great hubs soon accumulate inlinks and thereby high prestige, thus becoming good authorities as well.

Higher-order eigenvectors and clustering

If the query is ambiguous (e.g., "Java" or "jaguar") or polarized (e.g., "abortion" or "cold fusion"), the expanded set will contain a few, almost disconnected, link communities. In each community there may be dense bipartite subgraphs. In

```
1: while X does not converge do
2:   X ← MX
3:   for i = 1, 2, ... do
4:     for j = 1, 2, ..., i − 1 do
5:       X(i) ← X(i) − (X(i) · X(j))X(i)
         {orthogonalize X(i) with regard to column X(j)}
6:     end for
7:     normalize X(i) to unit L2 norm
8:   end for
9: end while
```

FIGURE 7.3 Finding higher-order eigenvectors in HITS using power iterations.

such cases, the highest-order eigenvectors found by HITS will reveal hubs and authorities in the largest near-bipartite component. One can tease out the structure and ranking within smaller components by calculating not only the principal eigenvector but also a few more. The iterations expressed in Equation (7.10) find the *principal eigenvectors* of EE^T and E^TE. Other eigenvectors can also be found using the iterative method. Given an $n \times n$ matrix M ($= E^TE$, say) for which we wish to find k eigenvectors, we initialize an $n \times k$ matrix X (generalizing the $n \times 1$ vector before) with positive entries. Let $X(i)$ be the ith column of X. The iterations are generalized to the steps shown in Figure 7.3 [91].

Similar to Larson's study (Figure 7.1), higher-order eigenvectors can reveal clusters in the graph structure. In the a or h vector, each graph node had only one number as a representation. Thanks to using X, each node now has k hub scores and k authority scores. These should not be interpreted as just more scores for ranking but as a multidimensional geometric embedding of the nodes. For example, if $k = 2$, one can plot each node as a point in the plane using its authority (or hub) score row-vector. For a polarized issue like "abortion," there are two densely linked communities on the Web, with sparse connections in between, mostly set up via eclectic hubs. A low-dimensional embedding and visualization may bring out community clustering graphically in case a query matches multiple link communities.

The connection between HITS and LSI/SVD

There is a direct mapping between finding the singular value decomposition (SVD) of E, as described in Section 4.3.4, and the eigensystem of EE^T or E^TE. Let the SVD of E be $U\Sigma V^T$, where $U^TU = \mathbf{I}$ and $V^TV = \mathbf{I}$ and Σ is a diagonal

matrix $\text{diag}(\sigma_1, \ldots, \sigma_r)$ of singular values, where r is the rank of E, and $\mathbf{I}$ is an identity matrix of suitable size. Then $EE^T = U\Sigma V^T V \Sigma U^T = U\Sigma \mathbf{I} \Sigma U^T = U\Sigma^2 U^T$, which implies that $EE^T U = U\Sigma^2$. Here if E is $n \times n$ with rank r, then U is $n \times r$; Σ and Σ^2 are $r \times r$. Specifically, $\Sigma^2 = \text{diag}(\sigma_1^2, \ldots, \sigma_r^2)$. $U\Sigma^2$ is $n \times r$ as well. If $U(j)$ is the jth column of U, we can write $EE^T U(j) = \sigma_j^2 U(j)$, which means that $U(j)$ is an eigenvector of EE^T with corresponding eigenvalue σ_j^2, for $j = 1, \ldots, r$. If Σ^2 is arranged such that $\sigma_1^2 \geq \ldots \sigma_r^2$, it turns out that finding the hub scores for E is the same as finding $U(1)$, and more generally, finding multiple hubs/authorities corresponds to finding many singular values of EE^T and $E^T E$.

Thus, the HITS algorithm is equivalent to running SVD on the hyperlink relation (`source,target`) rather than the (`term,document`) relation to which SVD is usually applied. Recall that SVD finds us vector representations for terms and documents in "latent semantic space." As a consequence of the equivalence shown above, a HITS procedure that finds multiple hub and authority vectors also finds a multidimensional representation for nodes in a hypertext graph. We can either present the SVD representation visually to aid clustering, or use one of the many clustering algorithms discussed in Chapter 4 on this representation of documents.

7.2.3 Stochastic HITS and Other Variants

Several subsequent studies have provided deeper analysis and comparison of HITS and PageRank. I provide here several observations that improve our understanding of how these algorithms work.

HITS is sensitive to local topology. The two graphs in Figure 7.4(a) differ only in the insertion of one node (5) to turn a single edge into a chain of two edges, something that frequently happens on the Web owing to a redirection or reorganization of a site. You can verify that this edge splitting upsets the scores for HITS quite significantly, whereas it leaves PageRanks relatively unaffected. More specifically, the update equations for authorities change from the system

$$a_2 \leftarrow 2a_2 + a_4 \tag{7.12}$$
$$a_4 \leftarrow a_2 + a_4 \tag{7.13}$$

to the new system

$$a_2 \leftarrow 2a_2 + a_4 \tag{7.14}$$
$$a_4 \leftarrow a_4 \tag{7.15}$$
$$a_5 \leftarrow a_2 + a_5 \tag{7.16}$$

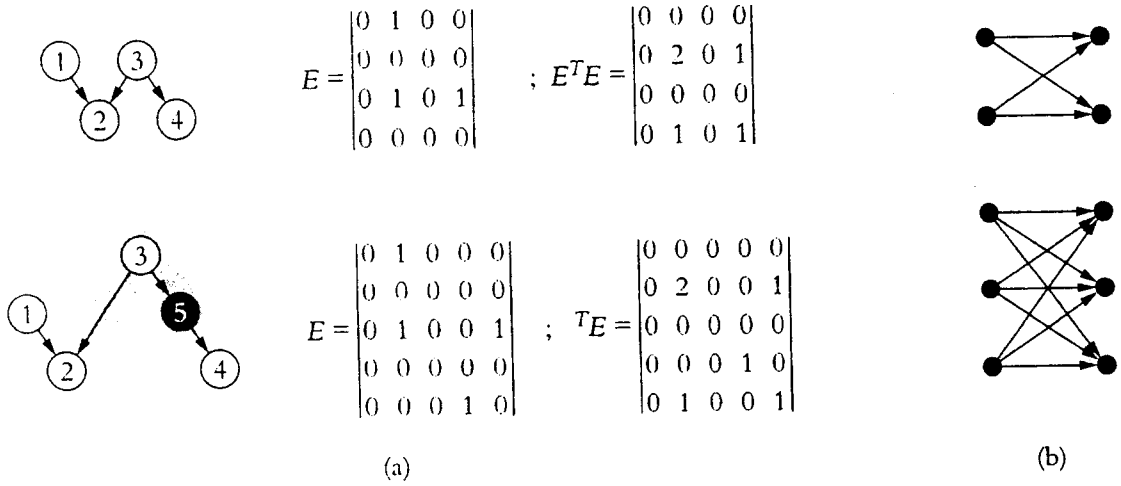

FIGURE 7.4 Minor perturbations in the graph may have dramatic effects on HITS scores (a). The principal eigenvector found by HITS favors larger bipartite cores (b).

Thus, node 5 takes the place of node 4, the mutual reinforcement between the authority scores of nodes 2 and 4 is lost, and node 4's authority score vanishes to zero compared to those of nodes 2 and 5.

HITS needs *bipartite cores* in the score reinforcement process. Consider the graph in Figure 7.4(b): it has two connected components, each of which is a complete bipartite graph, with 2×2 and 2×3 nodes. Let us assign all hub scores to 1 and start HITS iterations. After the first iteration, each authority score in the smaller component will be 2 and each authority score in the larger component will be 3. The scores will progress as follows:

Iteration	h_{small}	a_{small}	h_{large}	a_{large}
0	1	0	1	0
1a	1	2	1	3
1h	4	2	9	3
2a	4	8	9	27
2h	16	8	81	27

Here I ignore score scaling, because the relative magnitude of the scores illustrates the point. In general, after $i > 0$ full iterations, we can show that $a_{\text{small}} = 2^{2i-1}$ and $a_{\text{large}} = 3^{2i-1}$. Thus their ratio is $a_{\text{large}}/a_{\text{small}} = (3/2)^{2i-1}$, which grows without bound as i increases. Thus, in the principal eigenvector, the smaller component finds absolutely no representation. In contrast, it can be verified that PageRank

will not be so drastic; the random jump will ensure some positive scores for the prestige of all nodes.

Many researchers have sought to improve HITS by removing some of these anomalies. Lempel and Moran [135] proposed *SALSA, a stochastic algorithm for link structure analysis*. The goal of SALSA was to cast bipartite reinforcement in the random surfer framework. They proposed and analyzed the following random surfer specification while maintaining the essential bipartite nature of HITS:

1. At a node v, the random surfer chooses an inlink (that is, an incoming edge (u, v)) uniformly at random and moves to u.
2. Then, from u, the surfer takes a random forward link (u, w) uniformly at random.

Thus, the transition probability from v to w is

$$p(v, w) = \frac{1}{\text{InDegree}(v)} \sum_{(u,v),(u,w)\in E} \frac{1}{\text{OutDegree}(u)} \tag{7.17}$$

This may be regarded as the authority-to-authority transition; a symmetric formulation (follow an outlink and then an inlink) handles hub-to-hub transitions.

SALSA does not succumb to tightly knit communities to the same extent as HITS. In fact, the steady-state node probabilities of the authority-to-authority transition (assuming it is irreducible and ergodic) have a very simple form:

$$\pi_v \propto \text{InDegree}(v) \tag{7.18}$$

That is, the SALSA authority score is proportional to the in-degree. Although the sum in Equation (7.17) suggests a kind of sibling link reinforcement, the probabilities are chosen such that the steady-state node probabilities do not reflect any nonlocal prestige diffusion. It might be argued that a total absence of long-range diffusion is at the opposite extreme from HITS, and an intermediate level of reinforcement is better than either extreme.

A recent study by Ng et al. [162] shows that HITS's long-range reinforcement is bad for *stability*: random erasure of a small fraction (say, 10%) of nodes or edges can seriously alter the ranks of hubs and authorities. It turns out that PageRank is much more stable to such perturbations, essentially because of its random jump step. Ng et al. propose to recast HITS as a bidirectional random walk by a "random surfer" similar to PageRank: Every timestep, with probability d, the surfer jumps

to a node in the base set uniformly at random. With the remaining probability $1 - d$:

- If it is an odd timestep, the surfer takes a random outlink from the current node.
- If it is an even timestep, the surfer goes backward on a random inlink leading to the current node.

Ng et al. showed that this variant of HITS with random jumps has much better stability in the face of small changes in the hyperlink graph, and that the stability improves as d is increased. (They also showed this to be the case with PageRank.) Obviously, $d = 1$ would be most stable but useless for ranking: scores would diffuse all over. There is no recipe known for setting d based on the graph structure alone. It is clear that, at some stage, page content must be reconciled into graph models of the Web to complete the design of Web IR systems [98].

7.3 Shortcomings of the Coarse-Grained Graph Model

Both HITS and PageRank use a coarse-grained model of the Web, where each page is a node in a graph with a few scores associated with it. The model takes no notice of either the text or the markup structure on each page. (HITS leaves the selection of the base set to an external IR algorithm.)

In real life, Web pages are more complex than the coarse-grained model suggests. An HTML page sports a tag-tree structure, which is rendered by browsers as roughly rectangular regions with embedded text and hyperlinks. Unlike HITS or PageRank, human readers do not pay equal attention to all the links on a page. They use the position of text and links (and their interpretation of the text, of course) to carefully judge where to click to continue on their (hardly random) surfing.

Algorithms that do not model the behavior of human information foragers may fall prey to many artifacts of Web authorship, which I illustrate in this section. In the next section, I will describe several enhancements to the model and algorithms that avoid such pitfalls.

7.3.1 Artifacts of Web Authorship

The central assumption in PageRank and HITS is that a hyperlink confers authority. Obviously, this holds only if the hyperlink was created as a result of editorial judgment based on the contents of the source and target pages, as is

largely the case with social networks in academic publications. Unfortunately, that central assumption is increasingly being violated on the Web.

Much has changed about authoring Web pages ever since those algorithms were proposed. HTML is increasingly generated by programs, not typed in by hand. Pages are often generated from templates and/or dynamically from relational and semistructured databases (e.g., Zope; *zope.org/*). There are sites designed by companies whose mission is to increase the number of search engine hits for their customers. Their common strategies include stuffing irrelevant words in pages and linking up their customers in densely connected cliques, even if those customers have nothing in common. The creation and dissemination of hypertext happens at an unprecedented scale today and is inexorably coupled with commerce and advertising. I will describe three related ways in that these authoring idioms manifest themselves.

Nepotistic links

Kleinberg summarily discarded links connecting pages on the same host, because these links, largely authored by the same person, did not confer authority in the same sense as an academic citation, and could therefore be regarded as "nepotistic."[2]

Soon after HITS was published, Bharat and Henzinger [18] found that the threat of nepotism was not necessarily limited to same-site links. Two-site nepotism (a pair of Web sites endorsing each other) was on the rise. In many trials with HITS, they found two distinct sites h_1 and h_2, where h_1 hosted a number of pages u linking to a page v on h_2, driving up $a(v)$ beyond what may be considered fair.

Two-host nepotism can also happen because of Web infrastructure issues, for example, in a site hosted on multiple servers such as `www.yahoo.com` and `dir12.yahoo.com`, or the use of the relative URLs with regard to a base URL specified with the `<a base=...>` HTML construct. If it is a simple case of mirroring, the algorithms in Section 3.3.2 will generally fix the problem, but deliberate nepotism also exists on the Web.

Clique attacks

Over time, two-host nepotism evolved into multihost nepotism, thanks to the culture of professional Web-hosting and "portal" development companies. It is now surprisingly common to encounter query response pages with elaborate

2. Page et al. do not discuss nepotistic links in their paper.

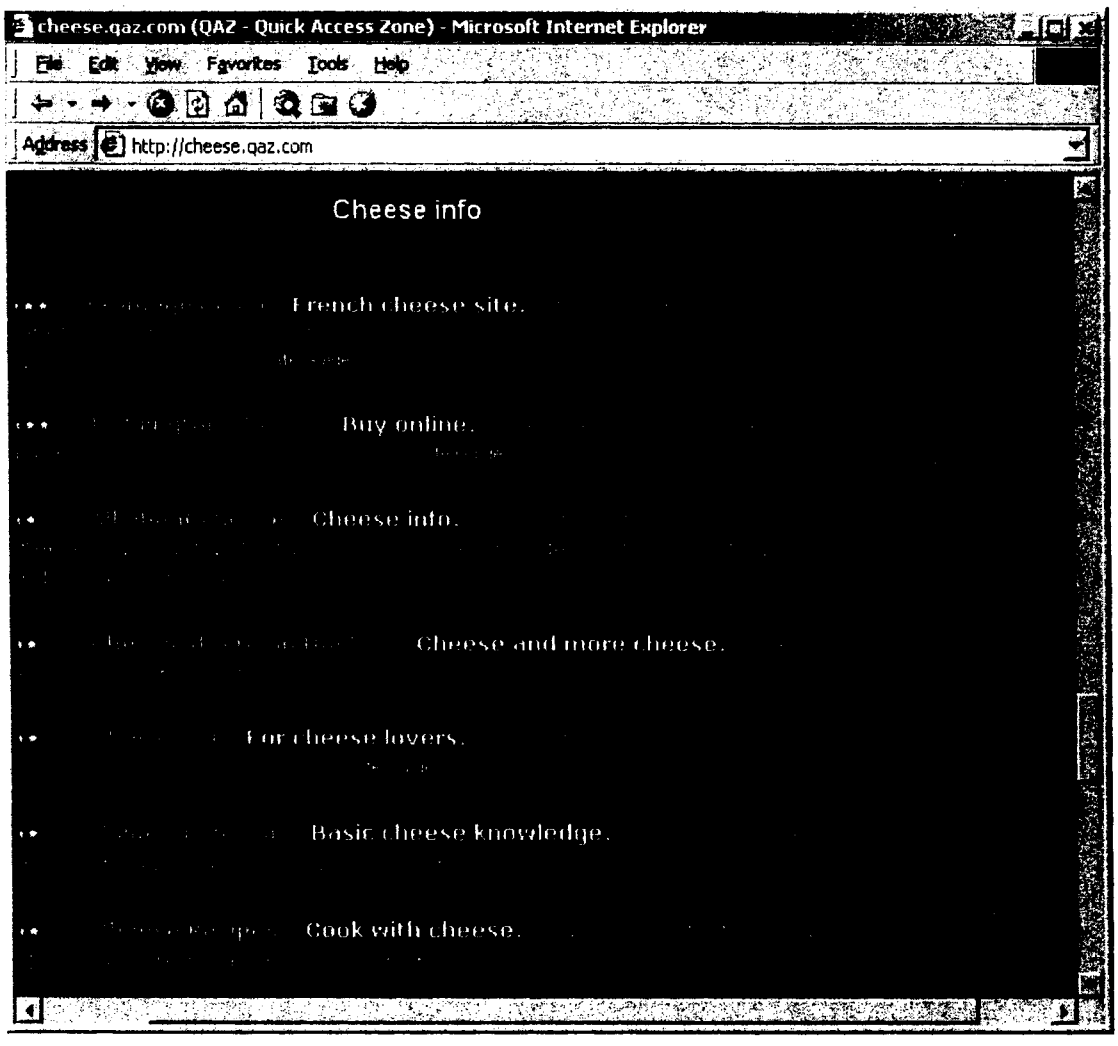

FIGURE 7.5 Hyperlinks generated from templates in navigation bars do not reflect content-based editorial judgment and often implement "clique attacks" that foil HITS-like algorithms. There are only a handful of links related to *cheese* on this page, but over 60 nepotistic links going to different hosts from *ads.qaz.com/* through *women.qaz.com/*.

navigation bars having links to other sites with *no* semantic connection, just because these sites are all hosted by a common business. I show one example in Figure 7.5, but the Web has plenty of such pages and sites.[3] These sites form a densely connected graph, sometimes even a completely connected graph, which

3. Although these sites might disappear with time, I will give some more examples: *www.411web.com/*, *www.depalma-enterprises.com/*, *www.cyprus-domains.com/*, and *www.usa.worldweb.com/*.

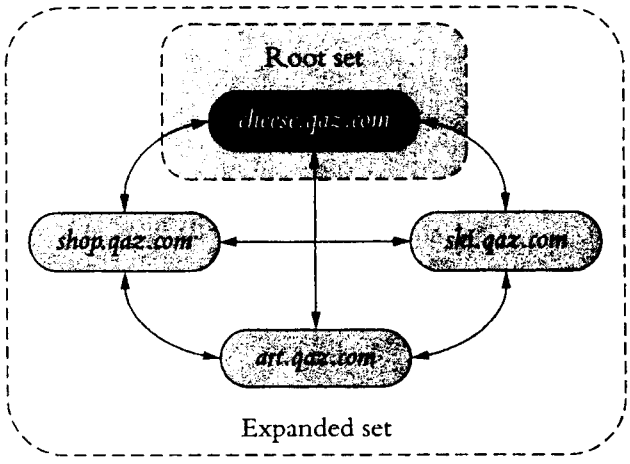

FIGURE 7.6 How a clique attack takes over link-based rankings.

led to my naming the phenomenon a "clique attack." Sometimes members of the clique have URLs sharing substrings, but they may map to different IP addresses. It is not easy to judge from the graph alone whether the clique is a bona fide, content-inspired link community or has been created deliberately. An example of a clique attack is shown in Figure 7.6. Both HITS and PageRank can fall prey to clique attacks, although by tuning d in PageRank, the effect can be reduced.

Mixed hubs

Another problem with decoupling the user's query from the link-based ranking strategy is that some hubs may be *mixed* without any attempt on the part of the hub writer to confound a search engine. Technically, this is hard to distinguish from a clique attack, but probably happens even more frequently than clique attacks. For example, a hub u containing links relevant to the query "movie awards" may also have some links to movie production companies. If a node v_1 relevant to movie awards gains authority score, the HITS algorithm (see Figure 7.2) would diffuse the score through u to a node v_2, which could be a movie production company homepage. Another example, in the form of a section of links about "Shakespeare" embedded in a page about British and Irish literary figures in general, is shown in Figure 7.7. Mixed hubs can be a problem for both HITS and PageRank, because neither algorithm discriminates between outlinks on a page. However, a system (such as Google) using PageRank may succeed at suppressing the ill effects by filtering on keywords at query time.

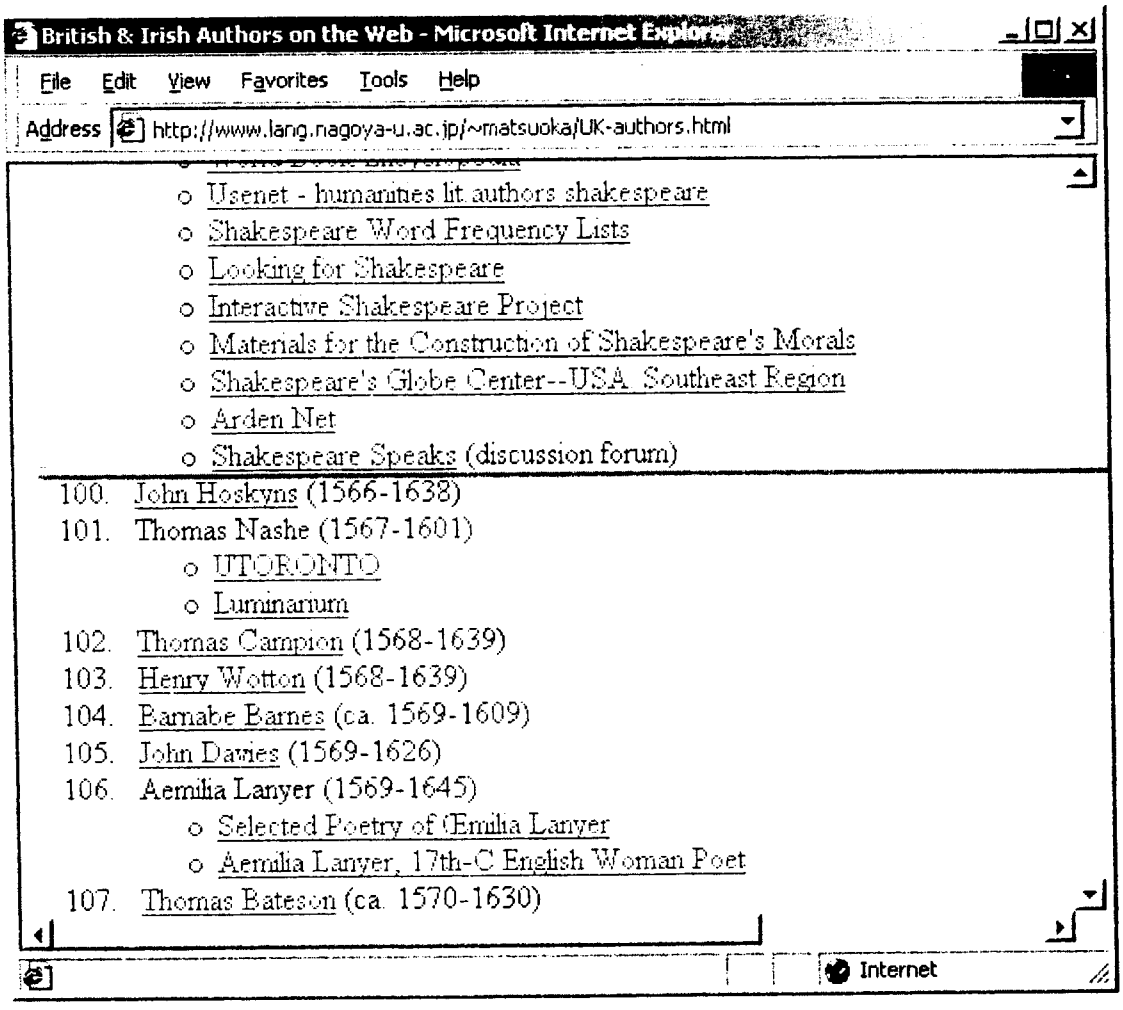

FIGURE 7.7 A mixed hub on British and Irish authors with one section dedicated to Shakespeare. (The horizontal line has been added by hand to demarcate the section.)

7.3.2 Topic Contamination and Drift

The expansion step in HITS was meant to increase recall and capture a larger graph G_q, which was subjected to eigen analysis. Why was this needed? Here is one reason. As of late 1996, the query "browser" would fail to include Netscape's Navigator and Communicator pages, as well as Microsoft's Internet Explorer page in the root set, because at that time these sites avoided a boring description like "browser" for their products. However, millions of pages included blurbs such as "this page is best viewed with a frames-capable *browser* such as . . ." and linked to these authoritative browser pages.

Conversely, sometimes good authorities would get included in the root set, but hubs linking to them might not be adequately represented in the root set for HITS to be able to estimate reliable authority scores for the former pages. The radius-1 expansion step of HITS would include nodes of both categories into the expanded graph G_q. Thus, the expansion step in HITS is primarily a recall-enhancing device. However, this boost in recall sometimes comes at the price of precision.

Consider a set of topics such as proposed by Yahoo!, and for simplicity assume that each Web page belongs to exactly one topic. Experimental evidence [45, 61] suggests that there is locality of content on the Web, that is, if a page is about cycling, following an outlink is more likely to lead to a page about cycling as well, compared to sampling a page uniformly at random from the Web. (The probability that the latter action will get us a page with a specific topic c is the fraction of pages in the Web belonging to topic c.)

This locality works in a very short radius, however. The probability of a page linking to another page of the same topic falls short of one for nontrivial topics, and the more specific the topic is, the smaller is this probability. Within a small number of links, the probability that all nodes have the same topic as the starting point vanishes rapidly.

Expansion by a single link was the maximum that could usually be tolerated by HITS; at radius two, most of the pages would be off-topic and the output of HITS would be largely unsatisfactory. (Indefinite graph expansion with HITS would make it degenerate to a PageRank-like scoring system with no connection to any specific query.) Even at radius one, severe contamination of the root set may occur, especially if pages relevant to the query are often linked to a broader, more densely linked topic. For example, at one time[4] the graph G_q corresponding to the query "movie awards" included a large number of movie company pages such as MGM and Fox, together with a number of hubs linking to them more densely than the subgraph that contained pages related to Oscar, Cannes, and so on. As a result, the hub and authority vectors have large components concentrated in nodes about movies rather than movie awards.

The above example is one of *topic generalization*. Another possible problem is that of *topic drift*. For example, pages on many topics are within a couple of

4. Both the Web and HITS have undergone significant evolution, so these specific anecdotes may be transient, although similar examples abound.

links of sites like Netscape, Internet Explorer, and Free Speech Online. Given the popularity of these sites, HITS (and PageRank) runs the danger of raising these sites to the top once they enter the expanded graph. Drift and contamination can sometimes be purposefully engineered, as in Figure 7.5. In effect, a Trojan horse page connected to a large clique can overwhelm any purely graph-based analysis (as in Figure 7.6).

An ad hoc fix is to list known *stop-sites* that would be removed from the expanded graph, but this could have undesirable effects as the notion of a "stop-site" is often context-dependent. For example, for the query "java," *www.java.sun.com/* is a highly desirable site, whereas for a narrower query like "swing," it may be considered too general.

Topic contamination may affect both HITS and PageRank. The top results from HITS may drift away from the query. The PageRank of irrelevant nodes may become unduly large because of membership or proximity to dense subgraphs. Again, a system (such as Google) using PageRank as one of many scores in ranking may be able to avoid problems by using a suitable relative weighting of scores.

7.4 Enhanced Models and Techniques

In this section we will consider hyperlink information in conjunction with text and markup information, model HTML pages at a finer level of detail, and propose enhanced prestige ranking algorithms.

The models that we have discussed thus far offer very simple and elegant representations for hypertext on the Web. Consequently, the mature fields of graph theory and matrix algebra can then be brought to bear. As we have seen in the previous section, such simple graph models break down in a variety of ways. This section offers solutions to some of the problems with the simplistic models.

7.4.1 Avoiding Two-Party Nepotism

Bharat and Henzinger [18] invented a simple and effective fix for two-site nepotism (the B&H algorithm). They observed that ascribing one unit of voting power to each page pointing to a given target may be too extreme, especially if those source pages are all on the same Web site. They proposed that a *site*, not a page, should be the unit of voting power. Therefore, if it is found that k pages on a single host link to a target page, these edges are assigned a weight of $1/k$. This is unlike HITS, where all edges have unit weight.