CHAPTER 14

Link Analysis and Web Search

14.1 Searching the Web: The Problem of Ranking

When you go to Google and type the word “Cornell,” the first result it shows you is www.cornell.edu, the home page of Cornell University. It's certainly hard to argue with this as a first choice, but how did Google “know” that this was the best answer? Search engines determine how to rank pages using automated methods that look at the Web itself, not some external source of knowledge, so the conclusion is that there must be enough information **intrinsic** to the Web and its structure to figure this out.

Before discussing some of the ideas behind the ranking of pages, let's begin by considering a few basic reasons why it's a hard problem. First, search is a hard problem for computers to solve in any setting, not just on the Web. Indeed, the field of **information retrieval** [36, 360] dealt with this problem for decades before the creation of the Web: automated information retrieval systems starting in the 1960s were designed to search repositories of newspaper articles, scientific papers, patents, legal abstracts, and other document collections in response to keyword queries. Information retrieval systems have always had to deal with the problem that keywords are a very limited way to express a complex information need. In addition to the fact that a list of keywords is short and inexpressive, it suffers from the problems of **synonymy** (multiple ways to say the same thing, so that your search for recipes involving scallions fails because the recipe you wanted called them “green onions”) and **polysemy** (multiple meanings for the same term, so that your search for information about the animal called a jaguar instead produces results primarily about automobiles, football players, and an operating system for the Apple Macintosh.)

For a long time, up through the 1980s, information retrieval was the province of reference librarians, patent attorneys, and other people whose jobs consisted of searching collections of documents. Such people were trained in how to formulate effective queries, and the documents they were searching tended to be written by professionals, using a controlled style and vocabulary. With the arrival of the Web, where everyone is an author and everyone is a searcher, the problems surrounding information retrieval exploded in scale and complexity.
To begin with, the diversity in authoring styles makes it much harder to rank documents according to a common criterion: on a single topic, one can easily find pages written by experts, novices, children, or conspiracy theorists, and not necessarily be able to tell which is which. Once upon a time, the fact that someone had the money and resources to produce a professional-looking, typeset, bound document meant that they were very likely (even if not always) someone who could be taken seriously. Today, anyone can create a Web page with high production values.

There is a correspondingly rich diversity in the set of people issuing queries, and the problem of multiple meanings becomes particularly severe. For example, when someone issues the single-word query “Cornell,” a search engine doesn’t have very much to go on. Did the searcher want information about the university? The university’s hockey team? The Lab of Ornithology run by the university? Cornell College in Iowa? The Nobel Prize–winning physicist Eric Cornell? The same ranking of search results can’t be right for everyone.

These represent problems that were already present in traditional information retrieval systems, just taken to new extremes. But the Web also introduces new kinds of problems. One is the dynamic and constantly changing nature of Web content. On September 11, 2001, many people ran to Google and typed “World Trade Center.” But there was a mismatch between what people thought they could get from Google and what they really got: at the time, Google was built on a model in which it periodically collected Web pages and indexed them, so the results were all based on pages that were gathered days or weeks earlier, and the top results were all descriptive pages about the building itself, not about what had occurred that morning. In response to such events, Google and the other main search engines built specialized “News Search” features, which collect articles more or less continuously from a relatively fixed number of news sources, so as to be able to answer queries about news stories minutes after they appear. Even today, such news search features are only partly integrated into the core parts of the search engine interface, and emerging Web sites such as Twitter continue to fill in the spaces that exist between static content and real-time awareness.

More fundamental still, and at the heart of many of these issues, is the fact that the Web has shifted much of the information retrieval question from a problem of scarcity to a problem of abundance. The prototypical applications of information retrieval in the pre-Web era had a “needle-in-a-haystack” flavor. For example, an intellectual-property attorney might express the information need, “find me any patents that have dealt with the design of elevator speed regulators based on fuzzy-logic controllers.” Such issues still arise today, but the hard part for most Web searches carried out by the general public is in a sense the opposite: to filter, from among an enormous number of relevant documents, the few that are most important. In other words, a search engine has no problem finding and indexing literally millions of documents that are genuinely relevant to the one-word query “Cornell”; the problem is that the human being performing the search is going to want to look at only a few of these. Which few should the search engine recommend?

An understanding of the network structure of Web pages is crucial for addressing these questions, as we now discuss.
14.2 Link Analysis Using Hubs and Authorities

So we're back to our question from the beginning of the chapter: in response to the one-word query "Cornell," what are the clues that suggest Cornell's home page, www.cornell.edu, is a good answer?

Voting by In-Links. In fact, there is a natural way to address this question, provided we start from the right perspective. This perspective is to note that there is not really any way to use features purely internal to the page www.cornell.edu to solve this problem: it does not use the word "Cornell" more frequently or more prominently than thousands of other pages, and so there is nothing on the page itself that makes it stand out. Rather, it stands out because of features on other Web pages: when a page is relevant to the query "Cornell," very often www.cornell.edu is among the pages it links to.

This is the first part of the argument that links are essential to ranking: that we can use them to assess the authority of a page on a topic, through the implicit endorsements that other pages on the topic confer through their links to it. Of course, each individual link may have many possible meanings: it may be off-topic; it may convey criticism rather than endorsement; it may be a paid advertisement. It is difficult for search engines to automatically assess the intent of each link. But we hope that, in aggregate, if a page receives many links from other relevant pages, then it is receiving a kind of collective endorsement.

In the case of the query "Cornell," we could operationalize this by first collecting a large sample of pages that are relevant to the query, as determined by a classical, text-only, information retrieval approach. We could then let pages in this sample "vote" through their links: which page on the Web receives the greatest number of in-links from pages that are relevant to Cornell? Even this simple measure of link-counting works quite well for queries such as "Cornell," where, ultimately, there is a single page that most people agree should be ranked first.

A List-Finding Technique. It's possible to make deeper use of the network structure than just counting in-links, and this brings us to the second part of the argument that links are essential. Consider, as a typical example, the one-word query "newspapers." Unlike the query "Cornell," there is not necessarily a single, intuitively "best" answer here; there are a number of prominent newspapers on the Web, and an ideal answer would consist of a list of the most prominent among them. With the query "Cornell," we discussed collecting a sample of pages relevant to the query and then letting them vote using their links. What happens if we try this for the query "newspapers"?

What you will typically observe, if you try this experiment, is that you get high scores for a mix of prominent newspapers (i.e., the results you'd want) along with pages that are going to receive a lot of in-links no matter what the query is - pages like Yahoo!, Facebook, Amazon, and others. In other words, to make up a very simple hyperlink structure for purposes of this example, we'd see something like Figure 14.1: the unlabeled circles represent our sample of pages relevant to the query "newspapers," and among the four pages receiving the most votes from them, two are newspapers (New York Times and USA Today) and two are not (Yahoo! and Amazon). This example
is designed to be small enough to try by hand; in a real setting, of course, there would be many plausible newspaper pages and many more off-topic pages.

But votes are only a very simple kind of measure that we can get from the link structure; there is much more to be discovered if we look more closely. To try getting more, we ask a different question. In addition to the newspapers themselves, there is another kind of useful answer to our query: pages that compile lists of resources relevant to the topic. Such pages exist for most broad enough queries: for “newspapers,” they would correspond to lists of links to online newspapers; for “Cornell,” one can find many alumni who maintain pages with links to the University, its hockey team, its Medical School, its Art Museum, and so forth. If we could find good list pages for newspapers, we would have another approach to the problem of finding the newspapers themselves.

In fact, the example in Figure 14.1 suggests a useful technique for finding good lists. Among the pages casting votes, we notice that a few of them in fact voted for many of the pages that received a lot of votes. It would be natural, therefore, to suspect that these pages have some sense of where the good answers are, and to score them highly as lists. Concretely, we could say that a page’s value as a list is equal to the sum of the votes received by all pages for which it voted. Figure 14.2 shows the result of applying this rule to the pages casting votes in our example.
Figure 14.2. Finding good lists for the query "newspapers": each page's value as a list is written as a number inside it.

The Principle of Repeated Improvement. If we believe that pages scoring well as lists actually have a better sense for where the good results are, then we should weight their votes more heavily. So, in particular, we could tabulate the votes again, but this time giving each page's vote a weight equal to its value as a list. Figure 14.3 shows what happens when we do this on our example: now the other newspapers have surpassed the initially high-scoring Yahoo! and Amazon, because these other newspapers were endorsed by pages that were estimated to be good lists.

In fact, you can recognize the intuition behind this reweighting of votes in the way we evaluate endorsements in our everyday lives. Suppose you move to a new town and hear restaurant recommendations from a lot of people. After discovering that certain restaurants get mentioned by a lot of people, you realize that certain people in fact had mentioned most of these highly recommended restaurants when you asked them. These people play the role of the high-value lists on the Web, and it's only natural to go back and take more seriously the more obscure restaurants that they recommended, because you now particularly trust their judgment. This last step is exactly what we are doing in reweighting the votes for Web pages.

The final part of the argument for link analysis is then the following: Why stop here? If we have better votes on the right-hand side of the figure, we can use these to get still more refined values for the quality of the lists on the left-hand side of the figure. And
Figure 14.3. Reweighting votes for the query "newspapers": each labeled page's new score is equal to the sum of the values of all lists that point to it.

with more refined estimates for the high-value lists, we can reweight the votes that we apply to the right-hand side once again. The process can go back and forth forever: it can be viewed as a principle of repeated improvement, in which each refinement to one side of the figure enables a further refinement to the other.

**Hubs and Authorities.** This suggests a ranking procedure that we can try to make precise, as follows [247]. First, we'll call the kinds of pages we were originally seeking—the prominent, highly endorsed answers to the queries—the authorities for the query. We'll call the high-value lists the hubs for the query. Now, for each page \( p \), we're trying to estimate its value as a potential authority and as a potential hub, and so we assign it two numerical scores: \( auth(p) \) and \( hub(p) \). Each of these starts out with a value equal to 1, indicating that we're initially agnostic as to which is the best in either of these categories.

Now, voting—in which we use the quality of the hubs to refine our estimates for the quality of the authorities—is simply the following:

*Authority Update Rule:* For each page \( p \), update \( auth(p) \) to be the sum of the hub scores of all pages that point to it.

On the other hand, the list-finding technique—in which we use the quality of the authorities to refine our estimates for the quality of the hubs—is the following:
Hub Update Rule: For each page $p$, update $hub(p)$ to be the sum of the authority scores of all pages that it points to.

Notice how a single application of the Authority Update Rule (starting from a setting in which all scores are initially 1) is simply the original casting of votes by in-links. A single application of the Authority Update Rule followed by a single application of the Hub Update Rule produces the results of the original list-finding technique. In general, the principle of repeated improvement says that, to obtain better estimates, we should simply apply these rules in alternating fashion, as follows:

- We start with all hub scores and all authority scores equal to 1.
- We choose a number of steps, $k$.
- We then perform a sequence of $k$ hub-authority updates. Each update works as follows:
  - First apply the Authority Update Rule to the current set of scores.
  - Then apply the Hub Update Rule to the resulting set of scores.
- At the end, the hub and authority scores may involve numbers that are very large. However, we only care about their relative sizes, so we can normalize to make them smaller: we divide down each authority score by the sum of all authority scores, and divide down each hub score by the sum of all hub scores. (For example, Figure 14.4 shows the result of normalizing the authority scores that we determined in Figure 14.3.)

![Diagram of web page links and scores]

**Figure 14.4.** Reweighting votes after normalizing for the query “newspapers.”
Figure 14.5. Limiting hub and authority values for the query “newspapers.”

What happens if we do this for larger and larger values of $k$? It turns out that the normalized values actually converge to limits as $k$ goes to infinity; in other words, the results stabilize so that continued improvement leads to smaller and smaller changes in the values we observe. We won’t prove this right now, but we provide a proof in Section 14.6 at the end of this chapter. Moreover, the analysis in that section shows that something even deeper is going on: except for a few rare cases (characterized by a certain kind of degenerate property of the link structure), we reach the same limiting values no matter what we choose as the initial hub and authority values, provided only that all of them are positive. In other words, the limiting hub and authority values are purely a property of the link structure, not of the initial estimates we use to start the process of computing them. (For the record, the limiting values for our “newspapers” example are shown, to three decimal places, in Figure 14.5.)

Ultimately, what these limiting values correspond to is a kind of equilibrium: their relative sizes remain unchanged if we apply the Authority Update Rule or the Hub Update Rule. As such, they reflect the balance between hubs and authorities that provided the initial intuition for them: your authority score is proportional to the hub scores of the pages that point to you, and your hub score is proportional to the authority scores of the pages you point to.

14.3 PageRank

The intuition behind hubs and authorities is based on the idea that pages play multiple roles in the network, and in particular that pages can play a powerful endorsement role.
without themselves being heavily endorsed. For queries with a commercial aspect – such as our query for newspapers in the previous section, or searches for particular products to purchase, or more generally searches that are designed to yield corporate pages of any type – there is a natural basis for this intuition. Competing firms do not link to each other, except in unusual circumstances, and so they can’t be viewed as directly endorsing each other; rather, the only way to conceptually pull them together is through a set of hub pages that link to all of them at once.

In other settings on the Web, however, endorsement is best viewed as passing directly from one prominent page to another – a page is important if it is cited by other important pages. This is often the dominant mode of endorsement, for example, among academic or governmental pages, among bloggers, or among personal pages more generally. It is also the dominant mode in the scientific literature. And it is this mode of endorsement that forms the basis for the PageRank measure of importance [79].

As with hubs and authorities, the intuition behind PageRank starts with simple voting based on in-links, and refines it using the principle of repeated improvement. In particular, the principle is applied here by having nodes repeatedly pass endorsements across their outgoing links, with the weight of a node’s endorsement based on the current estimate of its PageRank: nodes that are currently viewed as more important get to make stronger endorsements.

**The Basic Definition of PageRank.** Intuitively, we can think of PageRank as a kind of “fluid” that circulates through the network, passing from node to node across edges and pooling at the nodes that are the most important. Specifically, PageRank is computed as follows.

- In a network with $n$ nodes, we assign all nodes the same initial PageRank, $1/n$.
- We choose a number of steps, $k$.
- We then perform a sequence of $k$ updates to the PageRank values, using the following rule for each update:

  **Basic PageRank Update Rule:** Each page divides its current PageRank equally across its outgoing links and passes these equal shares to the pages it points to. (If a page has no outgoing links, it passes all its current PageRank to itself.) Each page updates its new PageRank to be the sum of the shares it receives.

Notice that the total PageRank in the network will remain constant as we apply these steps: since each page takes its PageRank, divides it up, and passes it along links, PageRank is neither created nor destroyed, just moved around from one node to another. As a result, we don’t need to do any normalizing of the numbers to prevent them from growing, the way we had to with hub and authority scores.

As an example, let’s consider how this computation works on the collection of 8 Web pages in Figure 14.6. All pages start out with a PageRank of $\frac{1}{8}$, and their PageRank values after the first two updates are given by the following table:

<table>
<thead>
<tr>
<th>Step</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{16}$</td>
<td>$\frac{1}{16}$</td>
<td>$\frac{1}{16}$</td>
<td>$\frac{1}{16}$</td>
<td>$\frac{1}{16}$</td>
<td>$\frac{1}{16}$</td>
<td>$\frac{1}{16}$</td>
</tr>
<tr>
<td>2</td>
<td>$\frac{1}{16}$</td>
<td>$\frac{3}{16}$</td>
<td>$\frac{1}{4}$</td>
<td>$\frac{1}{32}$</td>
<td>$\frac{1}{32}$</td>
<td>$\frac{1}{32}$</td>
<td>$\frac{1}{32}$</td>
<td>$\frac{1}{16}$</td>
</tr>
</tbody>
</table>
Figure 14.6. A collection of eight pages: page A has the largest PageRank, followed by pages B and C (which collect endorsements from A).

For example, A gets a PageRank of $\frac{1}{2}$ after the first update because it gets all of F's, G's, and H's PageRank, and half each of D's and E's. On the other hand, B and C each get half of A's PageRank, so they only get $\frac{1}{18}$ each in the first step. But once A acquires a lot of PageRank, B and C benefit in the next step. This is in keeping with the principle of repeated improvement: after the first update causes us to estimate that A is an important page, we weigh its endorsement more highly in the next update.

**Equilibrium Values of PageRank.** As with hub–authority computations, one can prove that, except in certain degenerate special cases, the PageRank values of all nodes converge to limiting values as the number of update steps, $k$, goes to infinity.

Because PageRank is conserved throughout the computation – with the total PageRank in the network equal to 1 – the limit of the process has a simple interpretation. We can think of the limiting PageRank values, one value for each node, as exhibiting the following kind of equilibrium: if we take the limiting PageRank values and apply one step of the Basic PageRank Update Rule, then the values at every node remain the same. In other words, the limiting PageRank values regenerate themselves exactly when they are updated. This description gives a simple way to check whether an assignment of numbers to a set of Web pages forms such an equilibrium set of PageRank values: we check that they sum to 1, and we check that when we apply the Basic PageRank Update Rule, we get the same values back.

For example, on the network of Web pages from Figure 14.6, we can check that the values shown in Figure 14.7 have the desired equilibrium property: assigning a PageRank of $\frac{4}{13}$ to page A, $\frac{2}{13}$ to each of B and C, and $\frac{1}{13}$ to the five other pages achieves this equilibrium.

Now, depending on the network structure, the set of limiting values may not be the only values that exhibit this kind of equilibrium. However, one can show that if the network is strongly connected – that is, each node can reach each other node by a directed path, following the definition from Chapter 13 – then there is a unique set of equilibrium values, and so whenever limiting PageRank values exist, they are the only values that satisfy this equilibrium.

**Scaling the Definition of PageRank.** There is a difficulty with the basic definition of PageRank, however: in many networks, the “wrong” nodes can end up with all the
PageRank. Fortunately, there is a simple and natural way to fix this problem, yielding the actual definition of PageRank that is used in practice. Let’s first describe the problem and then its solution.

To trigger the problem, suppose we take the network in Figure 14.6 and make a small change, so that F and G now point to each other rather than pointing to A. The result is shown in Figure 14.8. Clearly this ought to weaken A somewhat, but in fact a much more extreme thing happens: PageRank that flows from C to F and G can never circulate back into the rest of the network, and so the links out of C function as a kind of “slow leak” that eventually causes all the PageRank to end up at F and G. We can indeed check that by repeatedly running the Basic PageRank Update Rule, we converge to PageRank values of 1/2 for each of F and G, and 0 for all other nodes.

This is clearly not what we wanted, but it’s an inevitable consequence of the definition. And it becomes a problem in almost any real network to which PageRank is applied: as long as there are small sets of nodes that can be reached from the rest of the graph, but have no paths back, then PageRank will accumulate there. Fortunately, there is a simple and natural way to modify the definition of PageRank to get around this problem, and it follows from the “fluid” intuition for PageRank. Specifically, if we think about the (admittedly simplistic) question of why all the water on earth doesn’t inexorably run downhill and reside exclusively at the lowest points, it’s because there’s a counterbalancing process at work: water also evaporates and gets drawn back down at higher elevations.

We can use this idea here. We pick a scaling factor \( s \) that should be strictly between 0 and 1. We then replace the Basic PageRank Update Rule with the following:

**Scaled PageRank Update Rule:** First apply the Basic PageRank Update Rule. Then scale down all PageRank values by a factor of \( s \). This means that the total PageRank in the network has shrunk from 1 to \( s \). We divide the residual \( 1 - s \) units of PageRank equally over all nodes, giving \((1 - s)/n\) to each.

If we think back to the bow-tie structure of the Web from Chapter 13, there is a way to describe the problem in these terms as well: there are many “slow leaks” out of the giant strongly connected component (SCC), and so in the limit, all nodes in the giant SCC will get PageRank values of 0; instead, all the PageRank will end up in set OUT of downstream nodes.
Figure 14.8. The same collection of eight pages, but F and G have changed their links to point to each other instead of to A. Without a smoothing effect, all the PageRank would go to F and G.

This rule also preserves the total PageRank in the network, since it is just based on redistribution according to a different "water cycle" that evaporates $1 - s$ units of PageRank in each step and rains it down uniformly across all nodes.

The Limit of the Scaled PageRank Update Rule. One can show that repeated application of the Scaled PageRank Update Rule converges to a set of limiting PageRank values as the number of updates, $k$, goes to infinity. Moreover, for any network, these limiting values form the unique equilibrium for the Scaled PageRank Update Rule: they are the unique set of values that remain unchanged under the application of this update rule. Notice, of course, that these values depend on our choice of the scaling factor $s$: in effect, there is really a different update rule for each possible value of $s$.

This is the version of PageRank that is used in practice, with a scaling factor $s$ that is usually chosen to be between 0.8 and 0.9. The use of the scaling factor also turns out to make the PageRank measure less sensitive to the addition or deletion of small numbers of nodes or links [268, 422].

Random Walks: An Equivalent Definition of PageRank. To conclude our discussion in this section, we now describe an equivalent formulation of PageRank that looks quite different on the surface, but in fact leads to exactly the same definition.

It works as follows. Consider someone who is randomly browsing a network of Web pages, such as the one in Figure 14.6. They start by choosing a page at random, picking each page with equal probability. They then follow links for a sequence of $k$ steps: in each step, they pick a random outgoing link from their current page and follow it to where it leads. (If their current page has no outgoing links, they just stay where they are.)

\footnote{As an aside about our earlier motivating example, one can check that using a value of $s$ in this range doesn't completely fix the problem with Figure 14.8: nodes F and G still get most (though no longer all) of the PageRank under the scaled update rule with such values of $s$. The problem is that an eight-node example is simply too small for the redistribution of the PageRank to truly offset the problem of a slow leak into a dead-end region of the network: on only eight nodes, a "slow leak" isn't actually so slow. However, on large networks such as are used in real applications, the redistribution of PageRank works well to give very small limiting PageRank values to most nodes outside the giant strongly connected component of the network.}
are.) Such an exploration of nodes performed by randomly following links is called a random walk on the network. We should stress that this is not meant to be an accurate model of an actual person exploring the Web; rather it is a thought experiment that leads to a particular definition.

In Section 14.6, we analyze this random walk and show the following fact.

**Claim:** The probability of being at a page $X$ after $k$ steps of this random walk is precisely the PageRank of $X$ after $k$ applications of the Basic PageRank Update Rule.

Given that the two formulations of PageRank — based on repeated improvement and random walks, respectively — are equivalent, we do not strictly speaking gain anything at a formal level by having this new definition. But the analysis in terms of random walks provides some additional intuition for PageRank as a measure of importance: the PageRank of a page $X$ is the limiting probability that a random walk across hyperlinks will end up at $X$ as we run the walk for larger and larger numbers of steps.

This equivalent definition using random walks also provides a new and sometimes useful perspective for thinking about some of the issues that came up earlier in the section. For example, the "leakage" of PageRank to nodes $F$ and $G$ in Figure 14.8 has a natural interpretation in terms of the random walk on the network. In the limit, as the walk runs for more and more steps, the probability of the walk reaching $F$ or $G$ is converging to 1, and once it reaches either $F$ or $G$, it is stuck at these two nodes forever. Thus, the limiting probabilities of being at $F$ and $G$ are converging to $\frac{1}{2}$ each, and the limiting probabilities are converging to 0 for all other nodes.

We will also show in Section 14.6 how to formulate the Scaled PageRank Update Rule in terms of random walks. Rather than simply following a random edge in each step, the walker performs a "scaled" version of the walk as follows. With probability $s$, the walker follows a random edge as before; with probability $1 - s$, the walker jumps to a random node anywhere in the network, choosing each node with equal probability.

### 14.4 Applying Link Analysis in Modern Web Search

The link analysis ideas described in Sections 14.2 and 14.3 have played an integral role in the ranking functions of the current generation of Web search engines, including Google, Yahoo!, Microsoft's search engine Bing, and Ask. In the late 1990s, it was possible to produce reasonable rankings using these link analysis methods almost directly on top of conventional search techniques, but with the growth and enormously expanding diversity of Web content since then, link analysis ideas have been extended and generalized considerably, so that they are now used in a wide range of different ways inside the ranking functions of modern search engines.

It is difficult to say anything completely concrete about the current ranking functions of the main search engines, given that they are constantly evolving in complexity, and given that the search engine companies themselves are extremely secretive about what goes into their ranking functions. (There are good reasons for this secrecy, as we will discuss later.) But we can make general observations coupled with sentiments that represent the conventional wisdom of the search community. In particular, PageRank
was one of the original and central ingredients of Google, and it has always been a core component of Google's methodology. The importance of PageRank as a feature in Google's ranking function has long been claimed to be declining over time, however. For example, in 2003 and 2004, a significant overhaul of Google's ranking function was generally believed to involve non-PageRank styles of link analysis, including a method called Hilltop developed by Krishna Bharat and George Mihaila [58] as an extension of the two-sided form of endorsement behind hubs and authorities. Around a similar time period, the search engine Ask rebuilt its ranking function around hubs and authorities, though its recent extensions have increasingly blended this with many other features as well.

Combining Links, Text, and Usage Data. While our emphasis on link analysis in this chapter is meant to motivate the ideas in a clean setting, in practice one clearly needs to closely integrate information from both network structure and textual content in order to produce the highest-quality search results. One particularly effective way to combine text and links for ranking is through the analysis of anchor text, the highlighted bits of clickable text that activate a hyperlink leading to another page [102]. Anchor text can be a highly succinct and effective description of the page residing at the other end of a link; for example, if you read "I am a student at Cornell University" on someone's Web page, it's a good guess that clicking on the highlighted link associated with the text "Cornell University" will take you to a page that is in some way about Cornell.

In fact, the link analysis methods we have been describing can be easily extended to incorporate textual features such as anchor text. In particular, the basic forms of both hubs and authorities and PageRank perform updates simply by adding up values across links. But if certain links have highly relevant anchor text while others don't, we can weight the contributions of the relevant links more heavily than the others; for example, as we pass hub or authority scores, or PageRank values, across a link, we can multiply them by a factor that indicates the quality of the anchor text on the link [57, 102].

In addition to text and links, search engines use many other features as well. For example, the way in which users choose to click or not click on a search result conveys a lot of information: if, among a search engine's ranked results for the query "Cornell," most users skip the first result and click on the second, it suggests that the first two results should potentially be reordered. There is ongoing research on methods for tuning search results based on this type of feedback [228].

A Moving Target. A final important aspect of Web search serves to illustrate a basic game-theoretic principle that we have encountered many times already—that you should always expect the world to react to what you do. As search grew into the dominant means of accessing information on the Web, it mattered to a lot of people whether they ranked highly in search engine results. For example, many small companies had business models that increasingly depended on showing up among the first screen of

---

2 Of course, not all anchor text is useful; consider the ubiquitous bit of Web page text, "For more information, click here." Such examples make you realize that creating useful anchor text is an aspect of hypertext authoring style worth paying attention to.
Google’s results for common queries ranging from “Caribbean vacations” to “vintage records.” An update to Google’s ranking function that pushed them off the first screen could spell financial ruin. Indeed, search-industry publications began naming some of Google’s more significant updates to its core ranking function in the alphabetic style usually reserved for hurricanes; the analogy was an apt one, since each of these updates was an unpredictable act of nature (in this case, Google) that inflicted millions of dollars of economic damage.

With this in mind, people who depended on the success of their Web sites increasingly began modifying their Web-page authoring styles to score highly in search engine rankings. For people who had conceived of Web search as a kind of classical information retrieval application, this was something novel. Back in the 1970s and 1980s, when people designed information retrieval tools for scientific papers or newspaper articles, authors were not overtly writing their papers or abstracts with these search tools in mind. From the relatively early days of the Web, however, people have written Web pages with search engines quite explicitly in mind. At first, this was often done using over-the-top tricks that aroused the ire of the search industry; as the digital librarian Cliff Lynch noted at the time, “Web search is a new kind of information retrieval application in that the documents are actively behaving badly.”

Over time, though, the use of focused techniques to improve a page’s performance in search engine rankings became regularized and accepted, and guidelines for designing these techniques emerged; a fairly large industry known as search engine optimization (SEO) came into being, consisting of search experts who advise companies on how to create pages and sites that rank highly. And so, to return to the game-theoretic view, the growth of SEO followed naturally once search became such a widespread application on the Web; it simply mattered too much to too many people that they be easily findable through search.

These developments have had several consequences. First, for search engines, the “perfect” ranking function will always be a moving target: if a search engine maintains the same method of ranking for too long, Web-page authors and their consultants become too effective at reverse-engineering the important features, and the search engine is in effect no longer in control of what ranks highly. Second, search engines are incredibly secretive about the internals of their ranking functions—not just to prevent competing search engines from finding out what they’re doing, but also to prevent designers of Web sites from finding out.

And finally, with so much money at stake, the search industry turned these developments into a very successful business model based on advertising. Rather than simply showing results computed by a ranking function, the search engine offered additional slots on the main results page through a market in which sites could pay for placement. Thus, when you look at a search results page today, you see the results computed by the ranking function alongside the paid results. We have just seen some of the ideas behind ranking functions; the paid results, as we will see in the next chapter, are allocated using the kinds of matching markets discussed in Chapter 10.

\(^4\) Of course, one can argue that, at a much less overt level, the development of standard authoring styles in these domains has been motivated by the goal of making these kinds of documents easier to classify and organize.
14.5 Applications beyond the Web

Link analysis techniques of the kind we’ve been discussing have been applied to a wide range of other settings, both before and after their use on the Web. In essentially any domain where information is connected by a network structure, it becomes natural to infer measures of authority from the patterns of links.

Citation Analysis. As we discussed in Chapters 2 and 13, the study of citations among scientific papers and journals has a long history that significantly predates the Web [145]. A standard measure in this field is Garfield’s impact factor for a scientific journal [177], defined to be the average number of citations received by a paper in the given journal over the past two years. This type of voting by in-links can thus serve as a proxy for the collective attention that the scientific community pays to papers published in the journal.

In the 1970s, Pinski and Narin [341] extended the impact factor by taking into account the idea that not all citations should be counted equally; rather, citations from journals that are themselves of high impact should be viewed as more important. This can be viewed as a use of the principle of repeated improvement, in the context of the scientific literature, just as we’ve seen it used for Web-page ranking. Pinski and Narin used this to formulate a notion of influence weights for journals [180, 341] that is defined very similarly to the notion of PageRank for Web pages.

Link Analysis of U.S. Supreme Court Citations. Recently, researchers have adapted link analysis techniques from the Web to study the network of citations among legal decisions by U.S. courts [166, 377]. Citations are crucial in legal writing – to ground a decision in precedent and to explain the relation of a new decision to what has come before. Using link analysis in this context can help in identifying cases that play especially important roles in the overall citation structure.

In one example of this style of research, Fowler and Jeon [166] applied hub and authority measures to the set of all U.S. Supreme Court decisions, a collection of documents that spans more than two centuries. They found that the set of Supreme Court decisions with high authority scores in the citation network align well with the more qualitative judgments of legal experts about the Court’s most important decisions. This set includes some cases that acquired significant authority according to numerical measures shortly after they appeared, but which took much longer to gain recognition from the legal community.

Supreme Court decisions also provide a rich setting for looking at how authority can change over long time periods. For example, Fowler and Jeon analyzed the rising and falling authority of some of the key Fifth Amendment cases from the twentieth century, as illustrated in Figure 14.9. In particular, Brown v. Mississippi – a 1936 case concerning confessions obtained under torture – began rising rapidly in authority in the early 1960s as the Warren Court forcefully took on a range of issues surrounding due process and self-incrimination. This development ultimately led to the landmark case Miranda v. Arizona in 1966 and – with this clear precedent established – the need for citations to Brown v. Mississippi quickly declined as the authority of Miranda shot upward.
Figure 14.9. The rising and falling authority of key Fifth Amendment cases from the twentieth century illustrates some of the relationships among them. (Image from [166], courtesy of Elsevier Science and Technology Journals)

The analysis of Supreme Court citations also shows that significant decisions can vary widely in the rate at which they acquire authority. For example, Figure 14.10 (also from [166]) shows that Roe v. Wade – like Miranda – grew in authority very rapidly from the time it was first issued. On the other hand, the equally consequential Brown v. Board of Education only began acquiring significant authority in the citation network roughly a decade after it was issued. Fowler and Jeon argue that this trajectory aligns with legal scholars’ views of the case, writing, “Judicial specialists often point

Figure 14.10. Roe v. Wade and Brown v. Board of Education acquired authority at very different rates. (Image from [166], courtesy of Elsevier Science and Technology Journals)
towards the ruling issued in Brown as an example of a precedent that was legally weak when first issued, and was strengthened through the Civil Rights Act of 1964 and its application in subsequent civil rights cases [166].

This style of analysis thus shows how a strictly network-based analysis of a topic as intricate as legal precedent can reveal subtleties that align well with the views of the scholarly community. It also indicates some of the interesting effects that emerge when one tries to track the rising and falling pattern of authority in a complex domain—an activity that stands to provide important insights in many other settings as well.

14.6 Advanced Material: Spectral Analysis, Random Walks, and Web Search

We now discuss how to analyze the methods for computing hub, authority, and PageRank values. This analysis will require some basic familiarity with matrices and vectors. Building on this, we will show that the limiting values of these link-analysis measures can be interpreted as coordinates in eigenvectors of certain matrices derived from the underlying networks. The use of eigenvalues and eigenvectors to study the structure of networks is often referred to as the spectral analysis of graphs, and we will see that this theory forms a natural language for discussing the outcome of methods based on repeated improvement.

A. Spectral Analysis of Hubs and Authorities

Our first main goal is to show why the hub–authority computation converges to limiting values for the hub and authority scores, as claimed in Section 14.2. As a first important step in this process, we show how to write the Authority Update and Hub Update Rules from that section as matrix–vector multiplications.

Adjacency Matrices and Hub–Authority Vectors. We will view a set of n pages as a set of nodes in a directed graph. Given this set of nodes, labeled 1, 2, 3, . . . , n, let's encode the links among them in an n \times n matrix M as follows: the entry in the i\textsuperscript{th} row and j\textsuperscript{th} column of M, denoted M_{ij}, is equal to 1 if there is a link from node i to node j, and it is equal to 0 otherwise. We call this the adjacency matrix of the network. Figure 14.11 shows an example of a directed graph and its adjacency matrix. Given a large set of pages, we expect that most of them will have very few out-links relative to the total number of pages, and so most entries in the adjacency matrix will be equal to 0. As a result, the adjacency matrix is not necessarily a very efficient way to represent the network, but as we will see, it is conceptually very useful.

Now, since the hub and authority scores are lists of numbers—one associated with each of the n nodes of the network—we can represent them simply as vectors in n dimensions, where the i\textsuperscript{th} coordinate gives the hub or authority score of node i. Specifically, we write h for the vector of hub scores, where h\textsubscript{i} is equal to the hub score of node i, and we similarly write a for the vector of authority scores.
Hub and Authority Update Rules as Matrix–Vector Multiplication. Let's consider the Hub Update Rule in terms of the notation we've just defined. For a node $i$, its hub score $h_i$ is updated to be the sum of $a_j$ over all nodes $j$ to which $i$ has an edge. Note that these nodes $j$ are precisely the ones for which $M_{ij} = 1$. Thus, we can write the update rule as

$$h_i \leftarrow M_{i1}a_1 + M_{i2}a_2 + \cdots + M_{in}a_n,$$  \hspace{1cm} \text{(14.1)}$$

where we use the notation "\leftarrow" to mean that the quantity on the left-hand side is updated to become the quantity on the right-hand side. This is a correct way to write the update rule, since the values $M_{ij}$ as multipliers select out precisely the authority values that we wish to sum.

But Equation (14.1) corresponds exactly to the definition of matrix-vector multiplication, so we can write it in the following equivalent way:

$$h \leftarrow Ma.$$  

Figure 14.12 shows this for the example from Figure 14.11: the authority scores $(2, 6, 4, 3)$ lead to the hub scores $(9, 7, 2, 4)$ via the Hub Update Rule. Indeed, this is an example of a general principle: if you're updating a collection of variables according to a rule that selects out certain ones to add up, you can often write this update rule as a matrix–vector multiplication for a suitably chosen matrix and vector.

Figure 14.12. By representing the link structure using an adjacency matrix, the Hub and Authority Update Rules become matrix-vector multiplication. In this example, we show how multiplication by a vector of authority scores produces a new vector of hub scores.