9

Google and the *PageRank* Algorithm

The first three sections of this chapter make use of linear algebra (diagonalization, eigenvalues, and eigenvectors) and elementary probability theory (independence of events and conditional probability). These sections provide the basics and can be covered in about three hours. Combined, they give a good idea of how the PageRank algorithm works. Section 9.4 is more advanced, requiring a familiarity with real analysis (accumulation points and convergence of sequences); this section may be covered in one or two hours.

9.1 Search Engines

In the digital world, new problems are generally quickly solved by new algorithms or new hardware. Those who have used the world wide web for more than a few years, say since 1998, will no doubt remember the search engines provided by *AltaVista* and *Yahoo*. More than likely, these same people now use *Google*'s search engine. Surprisingly, among all the general-purpose search engines, *Google* rose to its current supremacy in a matter of months. It did so thanks to its algorithm for ranking search results: the *PageRank* algorithm. The goal of this chapter is to describe this algorithm and the mathematical foundations on which it is built: Markov chains.

Using a search engine is fairly simple. It starts with somebody sitting at a computer connected to the Internet, and a desire to learn about a particular subject. Suppose, for example, that he wants to learn about the annual snowfall in Montreal. He decides to query *Google*[1] with the keywords *precipitation*, *snow*, *Montreal*, and *century*. (Of these, the last word may seem a little strange. However, the user has chosen this word to indicate his desire for long-term statistics.) The search engine responds with a brief list of what it deems to be the best sources of information on the topic (see Figure 9.1). The horizontal bar at the top of the page indicates that the search was performed in

[1] *Google* can be found at `http://www.google.com`

C. Rousseau and Y. Saint-Aubin, *Mathematics and Technology*,
DOI: 10.1007/978-0-387-69216-6_9,

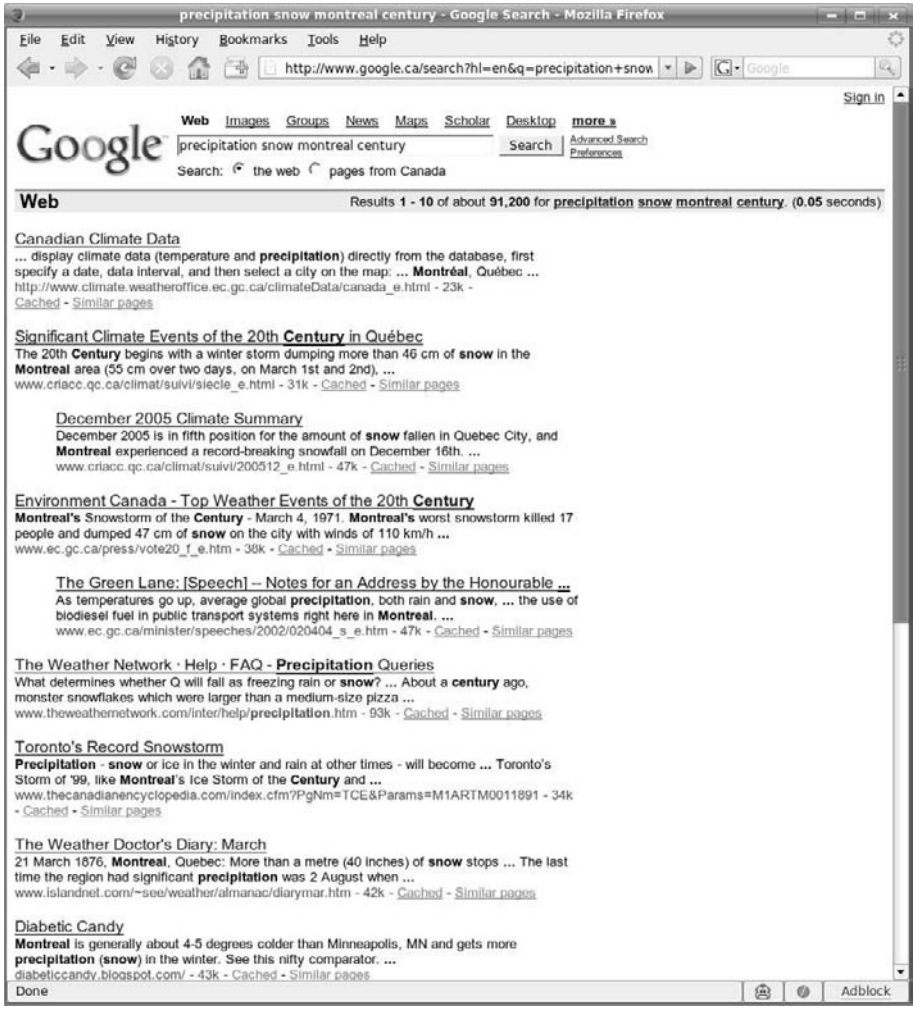

Fig. 9.1. A *Google* search on the keywords *precipitation*, *snow*, *montreal* and *century*.

less than a tenth of a second, and that around 91,200 potentially relevant pages were identified. The first is a link to an online database of Canadian climate data, provided by Environment Canada, which runs the Canadian weather office. (From here we can learn that the most snow seen since accurate record-keeping began was 384.3 cm in 1954! Thankfully, we also learn that the 30-year average is a little more reasonable, at 217.5 cm.) The first search result returned by *Google* often has quite a good chance of answering the user's question. How about the others? As we descend through the list,

the focus of the results tends to wander, with many documents concerning the Montreal Protocol on climate change. These later documents are of very little interest to the user, since they do not speak at all about snow in Montreal. But they are related in some sense, for they effectively all contain at least three of the four search terms.

This anecdote brings up an important point:[2] the pages that *Google* returns first are often exactly those that satisfy the user's needs. The search would definitely be hopeless if the user had to go through the 91,200 pages. The exact keywords entered by the user will obviously have an impact on the pages returned, but how in general can *Google* use a computer to guess the desires of the user?

Automated search tools have been around for a few decades. We can immediately think of several domains with large bodies of knowledge that need to be efficiently navigated: library catalogs, government registries (births, deaths, taxes) and professional databases (legal, dental, medical, parts catalogs). These bodies of information all have a few points in common. First off, they all contain data that lies within a single clearly defined scope. For example, all the books in a library contain a title, one or more authors, a publisher, etc. The *uniformity of the data* to be organized thus makes the database more easily categorized and more easily searched. The *quality* of the information is also very high. For example, books are normally entered into a library's catalog by professionals, and the error rate is thus very low. If and when an error occurs, the simplicity of the database makes it easy for corrections to be made. The *uniformity of the user's needs* is also an advantage in these systems. The goal of a library catalog is above all to maintain a concise listing of exactly what books are on hand. Even though specialized terms may exist (for example in medical or legal databases), the users are typically professionals in the field and will all be familiar with them. Thus, these databases may be searched with relative ease by their users. These databases all evolve relatively slowly. In a library, very few books leave the collection in a year, and a year that sees 10% growth in the catalog would be rare. Add to this the fact that the information already in a library catalog is always accurate, and never changes! The *growth rate* is therefore relatively slow, and such databases are easily maintained by humans. Finally, it is easy to achieve a *consensus* rating on the quality of the items in the database. In most university faculties, committees guide the purchase of new books for the library. Moreover, professors guide students directly toward the best books for their courses.

None of these characteristics exists on the web. The pages on the web have an immense diversity: technical, professional, promotional, commercial, entertainment, etc. The quality to be found is also very inconsistent: we can expect to find many spelling and grammar errors, as well as misinformation (whether these errors are accidental or otherwise). The users of the web are also as numbered and varied as the pages on the web, and their familiarity with search engines is extremely variable. The speed at which

[2] If the user were to repeat this search again today, chances are the results would be vastly different and in all probability there would be many more returned pages. This is due to the constantly changing and expanding nature of the world wide web.

the web evolves is staggering: as of the end of 2005 (when they stopped publishing the size of their database on their front page), *Google* was indexing well over 9 *billion* pages, with others appearing and disappearing daily. Finally, it seems illusory to establish a consensus on the relative quality of web pages given their number, their diversity, and the equally varying interests of the hundreds of millions of users worldwide. It seems that web pages have nothing in common!

In fact, this is a bit of a lie, since most pages on the web *do* have something in common. They are nearly all written in HTML (HyperText Markup Language) or in some related dialect. And the method in which they are related to each other is uniform: links between pages are all encoded in the same manner. These links consist of a few fixed characters preceding the address of the page, otherwise known as its URL (Uniform Resource Locator). These are precisely the links that a human user may follow in surfing the web, and which a computer can differentiate from the text, images, and other elements of a web page. In January 1998, four researchers from Stanford University, L. Page, S. Brin, R. Motwani, and T. Winograd, proposed an algorithm [3] for ranking pages on the web. This algorithm, *PageRank*, does not use the textual or visual content of the page, but rather the structure of the links between them.[3]

9.2 The Web and Markov Chains

The web is composed of billions of individual pages, and even more links between them.[4] As such, the web can be modeled as a directed graph, where pages are nodes, and links are directed edges between them. For example, Figure 9.2 represents a (small) web containing five pages (A, B, C, D, and E). The directed edges between the nodes indicate that

- the only link from page A leads to page B,
- page B links to pages A and C,
- page C links to pages A, B, and E,
- the only link from page D leads to page A, and
- page E links to pages B, C, and D.

In order to determine the ranking to be accorded to each of these five pages, we consider a simple version of the *PageRank* algorithm. Suppose that an impartial web surfer navigates through this web by randomly choosing links to follow. When he has only one choice (for example, if he finds himself on page D), then he will follow that link (leading to page A in this example). If he finds himself on page C, he will follow the link to page A one-third of the time and similarly for the links to pages B and E. In other

[3]The first four letters of *PageRank* refer to the first author's last name, and not to pages of the web.

[4]When Page et al. published their algorithm in 1998, they estimated the size of the web as roughly 150 million pages with 1.7 billion links between them. In early 2006, the web was estimated as containing around 12 billion pages.

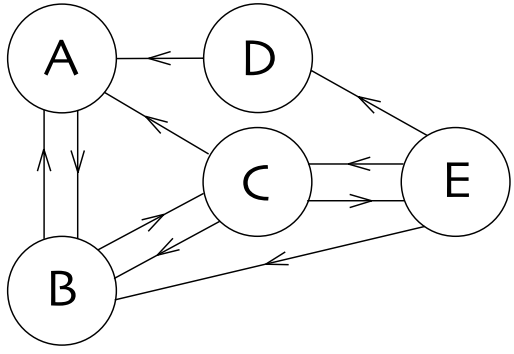

Fig. 9.2. A web of five pages and its links.

words, when he finds himself on a given page, he will randomly choose from among the outbound links, according each an equal probability. If such a web surfer were left to crawl the web in such a manner following one link per minute, where would he find himself in an hour, in two days, or after some large number of jumps? More precisely, given that his path is determined probabilistically, with what probability would he find himself on a given page after a given amount of time?

Figure 9.3 answers this question for the first two steps of an impartial web surfer starting at page C. This page contains three outbound links; thus the web surfer can end up only on one of the pages A, B, E. Thus, after the first step he would find himself on page A with probability $\frac{1}{3}$, on page B with probability $\frac{1}{3}$, and on page E with probability $\frac{1}{3}$. This is indicated in the middle column of Figure 9.3 by the three relations

$$p(A) = \frac{1}{3}, \qquad p(B) = \frac{1}{3}, \qquad p(E) = \frac{1}{3}.$$

Similarly,

$$p(C) = 0 \quad \text{and} \quad p(D) = 0$$

indicate that after one step the web surfer could not possibly be on page C or D, since no links from his previous page can lead him there. Each of the three possible paths is indicated by its probability of being taken. Furthermore, given that he must stay within the web, they satisfy

$$p(A) + p(B) + p(C) + p(D) + p(E) = 1.$$

The results after the first step are rather simple and predictable. However, even after only two steps, things begin to get complicated. The third column of Figure 9.3 gives the possible trajectories after a second step. If the web surfer was on A after the first step, he would be guaranteed to be on B after a second step. Since he had been on A with probability $\frac{1}{3}$, this path contributes $\frac{1}{3}$ to the probability of being on B after

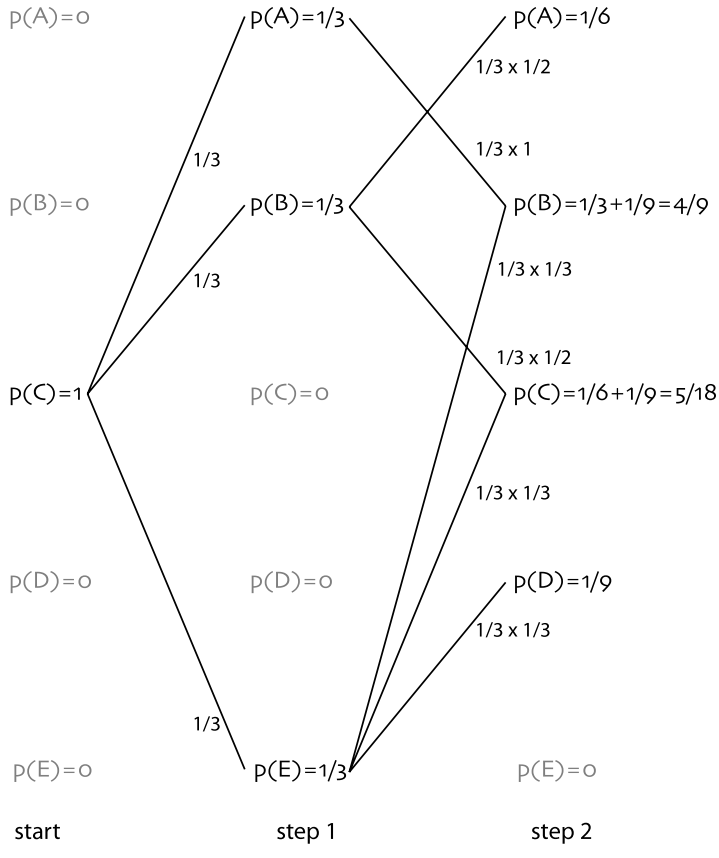

Fig. 9.3. The first two steps of an impartial web surfer starting at page C.

a second step. However, $p(B)$ does not equal $\frac{1}{3}$ after the second step, since there is another independent path that could lead him there: $C \to E \to B$. If the web surfer found himself on page E after the first step, he could choose (with equal probability) from the three links leading to pages B, C, and D. Each of these paths contributes $\frac{1}{3} \times \frac{1}{3} = \frac{1}{9}$ to the probabilities $p(B)$, $p(C)$, and $p(D)$ after the second step. Although there are more possibilities and the attached probabilities are more complicated, the end result is relatively simple. After two steps, the web surfer finds himself on a given page with the following probabilities:

$$p(A) = \frac{1}{6}, \qquad p(B) = \frac{4}{9}, \qquad p(C) = \frac{5}{18}, \qquad p(D) = \frac{1}{9}, \qquad p(E) = 0.$$

Again, we see that these probabilities satisfy

$$\begin{aligned} p(A)+p(B)+p(C)+p(D)+p(E) &= \frac{1}{6}+\frac{4}{9}+\frac{5}{18}+\frac{1}{9}+0 \\ &= \frac{3+8+5+2+0}{18}=1. \end{aligned}$$

At this point, the method should seem clear, and we could continue to calculate the probabilities after a few more steps. However, it is useful to formalize this impartial walk through the web. The tool best suited to this job is the theory of Markov chains.

A *random process* $\{X_n, n = 0, 1, 2, 3, \dots\}$ is a family of random variables parameterized by the integer n. We assume that each of these random variables X_n takes its values from a finite set T. In the example of the impartial web surfer, T is the set of pages in the web: $T = \{A, B, C, D, E\}$. For each step $n \in \{0, 1, 2, \dots\}$, the position of the web surfer is X_n. Sticking to the language of random processes, we determined earlier the probabilities of the possible outcomes for X_1 and X_2 assuming that the walk started from C. This can be rephrased as a conditional probability $P(I|J)$, which gives the probability that event I occurs given that event J has already occurred. For example, $P(X_1 = A|X_0 = C)$ gives the probability of the web surfer finding himself on page A at step 1 after having been on page C at the beginning (step 0). Thus

$$p(X_1 = A|X_0 = C) = \frac{1}{3}, \quad p(X_1 = B|X_0 = C) = \frac{1}{3}, \quad p(X_1 = C|X_0 = C) = 0,$$
$$p(X_1 = D|X_0 = C) = 0, \quad p(X_1 = E|X_0 = C) = \frac{1}{3},$$

and

$$p(X_2 = A|X_0 = C) = \frac{1}{6}, \quad p(X_2 = B|X_0 = C) = \frac{4}{9}, \quad p(X_2 = C|X_0 = C) = \frac{5}{18},$$
$$p(X_2 = D|X_0 = C) = \frac{1}{9}, \quad p(X_2 = E|X_0 = C) = 0.$$

The random walk followed by the impartial web surfer possesses the defining property of Markov chains. First off, we will define Markov chains.

Definition 9.1 *Let $\{X_n, n = 0, 1, 2, 3, \dots\}$ be a random process taking its values from the set $T = \{A, B, C, \dots\}$. We say that $\{X_n\}$ is a* Markov chain *if the probability $P(X_n = i)$, $i \in T$, depends only on the value of the process at the previous step, X_{n-1}, and not on any of the preceding steps, $X_{n-2}, X_{n-3}, \dots$. We define $N < \infty$ as the number of elements in T.*

In the example of the impartial web surfer, the random variables are the positions X_n after n steps. In thinking back to our earlier calculations we notice that in calculating the probabilities after the first step, $P(X_1)$, we used only the starting point. Similarly, in calculating the probabilities after the second step, $P(X_2)$, we used only the probabilities from the first step. This property of being able to calculate $P(X_n)$ using only the information from $P(X_{n-1})$ is the defining property of Markov chains. Are all random

processes Markov chains? Certainly not. It takes only a slight change to the rules of our impartial web surfer in order to lose the Markov property. Suppose that we want to prevent the web surfer from ever returning immediately to the page where he came from. For example, after the first step, our web surfer found himself on pages A, B, and E with equal probability. He cannot return to page C from page A, but he could possibly do so from pages B and E. Thus, we could prevent the web surfer from following the links to page C from pages B and E. Under these new rules, the web surfer would have only a single choice when arriving at page B from page C (he would have to go to page A), and he would be reduced to two choices at page E (either page B or page D). In prohibiting the web surfer from following links to its previous page we have lost the Markov property: the process has *memory*. In fact, in order to determine the probabilities $P(X_2)$ we need to know not only the probabilities at step 1, but also the page (or pages) where the web surfer was at the start (step zero). The rules that we originally defined are thus rather special in a mathematical sense: Markov chains have no memory of past states, and the future state is completely determined by the current state.

Markov chains are unique in that their behavior may be entirely characterized by their initial state ($p(C) = 1$ in the example of Figure 9.3) and a *transition matrix* given by

$$p(X_n = i \mid X_{n-1} = j) = p_{ij}. \tag{9.1}$$

A matrix P is a Markov chain transition matrix if and only if

$$p_{ij} \in [0,1] \quad \text{for all } i,j \in T \qquad \text{and} \qquad \sum_{i \in T} p_{ij} = 1 \quad \text{for all } j \in T. \tag{9.2}$$

For our impartial web surfer, the elements p_{ij} of the transition matrix P represent the probabilities of finding himself at page $i \in T$ when he is coming from page $j \in T$. However, our rules force the surfer to choose with equal probability from among the available links. Thus, if page j offers m links, then column j of P will contain $\frac{1}{m}$ in the rows corresponding to the m linked pages, and 0 in the remaining rows. The transition matrix for the simple web in Figure 9.2 is thus given by

$$P = \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{3} & 1 & 0 \\ 1 & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & \frac{1}{3} \\ 0 & 0 & \frac{1}{3} & 0 & 0 \end{pmatrix} \end{array} \begin{array}{c} A \\ B \\ C \\ D \\ E \end{array} \tag{9.3}$$

The columns of P indicate possible destinations: from page E the web surfer may proceed to pages B, C, and D. Similarly, the nonzero entries in rows indicate possible origins: the single nonzero entry in the fourth row indicates that we may arrive at page D only from page E.

What exactly does the second constraint of (9.2) mean? To clarify, we rewrite it with the help of the transition matrix defined in (9.1):

$$\sum_{i\in T} p_{ij} = \sum_{i\in T} p(X_n = i \mid X_{n-1} = j) = 1,$$

which may be read as follows: if at step $n-1$ the system is in state j (at page $j \in T$), then the probability of being in *any* possible state at step n is 1. Stated even more simply, this means that a web surfer on a given page at step $n-1$ must certainly find himself still in the web at step n. Thus, the constraint is actually rather simple.

This formalization has several advantages. The operation of matrix multiplication suffices to reproduce the multitude of tedious calculations performed as we followed the web surfer through his first two steps. As before, we assume that the web crawler starts at page C. Thus

$$p^0 = \begin{pmatrix} p(X_0 = A) \\ p(X_0 = B) \\ p(X_0 = C) \\ p(X_0 = D) \\ p(X_0 = E) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

The probability vector p^1 after the first step is given by $p^1 = Pp^0$, and therefore

$$p^1 = \begin{pmatrix} p(X_1 = A) \\ p(X_1 = B) \\ p(X_1 = C) \\ p(X_1 = D) \\ p(X_1 = E) \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{3} & 1 & 0 \\ 1 & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & \frac{1}{3} \\ 0 & 0 & \frac{1}{3} & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{3} \\ \frac{1}{3} \\ 0 \\ 0 \\ \frac{1}{3} \end{pmatrix},$$

the same as we calculated before. In the same manner, applying the transformation matrix again yields $p^2 = Pp^1$; the probability vector after the second step is therefore

$$p^2 = \begin{pmatrix} p(X_2 = A) \\ p(X_2 = B) \\ p(X_2 = C) \\ p(X_2 = D) \\ p(X_2 = E) \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{3} & 1 & 0 \\ 1 & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & \frac{1}{3} \\ 0 & 0 & \frac{1}{3} & 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{3} \\ \frac{1}{3} \\ 0 \\ 0 \\ \frac{1}{3} \end{pmatrix} = \begin{pmatrix} \frac{1}{6} \\ \frac{4}{9} \\ \frac{5}{18} \\ \frac{1}{9} \\ 0 \end{pmatrix}.$$

The same method may be followed to calculate the probability vector after any number of steps: $p^n = Pp^{n-1}$, or alternatively,

$$p^n = Pp^{n-1} = P(Pp^{n-2}) = \cdots = \underbrace{PP\cdots P}_{n \text{ times}} p^0 = P^n p^0.$$

The constraints of (9.2) on the transition matrix P result in several properties of Markov chains that are very important for the *PageRank* algorithm.

This first property we will examine can be seen by taking several powers of the transition matrix P. The powers P^4, P^8, P^{16}, and P^{32}, rounded to three decimal places, are given by

$$P^4 = \begin{pmatrix} 0.333 & 0.296 & 0.204 & 0.167 & 0.420 \\ 0.222 & 0.463 & 0.531 & 0.667 & 0.160 \\ 0.389 & 0.111 & 0.160 & 0.000 & 0.370 \\ 0.056 & 0.000 & 0.031 & 0.000 & 0.019 \\ 0.000 & 0.130 & 0.074 & 0.167 & 0.031 \end{pmatrix}, \quad P^8 = \begin{pmatrix} 0.265 & 0.313 & 0.294 & 0.323 & 0.279 \\ 0.420 & 0.360 & 0.409 & 0.372 & 0.381 \\ 0.217 & 0.233 & 0.191 & 0.201 & 0.252 \\ 0.031 & 0.022 & 0.018 & 0.012 & 0.035 \\ 0.067 & 0.072 & 0.088 & 0.092 & 0.052 \end{pmatrix},$$

$$P^{16} = \begin{pmatrix} 0.294 & 0.291 & 0.293 & 0.291 & 0.294 \\ 0.388 & 0.392 & 0.389 & 0.391 & 0.391 \\ 0.220 & 0.219 & 0.221 & 0.221 & 0.218 \\ 0.024 & 0.025 & 0.025 & 0.025 & 0.024 \\ 0.074 & 0.073 & 0.072 & 0.072 & 0.074 \end{pmatrix}, \quad P^{32} = \begin{pmatrix} 0.293 & 0.293 & 0.293 & 0.293 & 0.293 \\ 0.390 & 0.390 & 0.390 & 0.390 & 0.390 \\ 0.220 & 0.220 & 0.220 & 0.220 & 0.220 \\ 0.024 & 0.024 & 0.024 & 0.024 & 0.024 \\ 0.073 & 0.073 & 0.073 & 0.073 & 0.073 \end{pmatrix}.$$

We observe that P^m seems to converge to a constant matrix as m increases. As it turns out, this is not just by luck, but rather it is a property of most Markov chain transition matrices.

Property 9.2 *The transition matrix P of a Markov chain has at least one eigenvalue equal to* 1.

Proof: Recall that the eigenvalues of a matrix are always equal to the eigenvalues of its transpose. This is a result of the fact that both matrices share the same characteristic polynomial:

$$\Delta_{P^t}(\lambda) = \det(\lambda I - P^t) = \det(\lambda I - P)^t = \det(\lambda I - P) = \Delta_P(\lambda),$$

which itself follows from the fact that the determinant of a matrix is equal to that of its transpose. It is simple to find an eigenvector of P^t. Let $u = (1, 1, \ldots, 1)^t$. Then $P^t u = u$. In fact, expanding the matrix multiplication directly, we see that

$$\begin{aligned} (P^t u)_i &= \sum_{j=1}^{n} [P^t]_{ij} u_j = \sum_{j=1}^{n} p_{ji} \cdot 1, \qquad \text{since all } u_j \text{ are } 1, \\ &= 1, \end{aligned}$$

by (9.2). □

Property 9.3 *If λ is an eigenvalue of an $n \times n$ transition matrix P, then $|\lambda| \leq 1$. Furthermore, there exists an eigenvector associated to the eigenvalue $\lambda = 1$ with all nonnegative entries.*

This property is a direct result of a theorem attributed to Frobenius. Although the proof relies only on elementary linear algebra and analysis, it is far from simple. We will explore this proof in Section 9.4.

Hypotheses Before we continue, we will state three hypotheses that we will assume from now on.

(i) First off, we will suppose that there is exactly one eigenvalue such that $|\lambda| = 1$, and therefore by Property 9.2 this eigenvalue is 1.
(ii) Next, we will suppose that this eigenvalue is not degenerate, which is to say that the associated eigensubspace has dimension 1.
(iii) Finally, we will take for granted that the transition matrix P representing the web is diagonalizable, meaning that its eigenvectors form a basis.

The first two hypotheses are not actually true for all transition matrices, and it is in fact possible to construct valid transition matrices that violate both of them (see the exercises). However, these remain reasonable hypotheses for transition matrices generated by large webs. The third hypothesis is there to simplify the following result.

Property 9.4 *1. If the transition matrix P of a Markov chain satisfies the three hypotheses above, then there exists a unique vector π such that the entries $\pi_i = P(X_n = i), i \in T$, satisfy*

$$\pi_i \geq 0, \qquad \pi_i = \sum_{j \in T} p_{ij} \pi_j, \qquad \text{and} \qquad \sum_{i \in T} \pi_i = 1.$$

We will call the vector π the stationary regime *of the Markov chain.*
2. Regardless of the initial point $p_i^0 = P(X_0 = i)$ (where $\sum_i p_i^0 = 1$), the distribution of probabilities $P(X_n = i)$ will converge to the stationary regime π as $n \to \infty$.

Proof: The first point simply repeats the fact that P has a single eigenvector with eigenvalue 1 whose components sum to 1. In fact, the defining equation for the stationary regime is simply $\pi = P\pi$. In other words, π is the eigenvector of P associated with the nondegenerate eigenvalue 1. Property 2 tells us that π is composed of nonnegative entries. Since an eigenvector is always nonzero, the sum of its entries must be strictly positive. By renormalizing this vector we can therefore always ensure that $\sum_i \pi_i = 1$.

To show the second point we rewrite the initial state vector p^0 in terms of the basis formed by the eigenvectors of P. We index the eigenvalues of P as follows: $1 = \lambda_1 > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_N|$. Hypotheses *(i)* and *(ii)* tell us that the first inequality in this ordering is strict (that is, the absolute value of λ_1 is strictly larger than that of λ_2), while hypothesis *(iii)* assures us that the eigenvectors of P form a basis for the space of dimension N where P acts. (For this last step, the eigenvalues must be counted with their multiplicities.) Let v_i be the eigenvector associated with the eigenvalue λ_i. Furthermore, assume that v_1 has been normalized such that $v_1 = \pi$. The set $\{v_i, i \in T\}$ forms a basis, allowing us to write

$$p^0 = \sum_{i=1}^{N} a_i v_i,$$

where the a_i are the coefficients of p^0 in this basis.

We will show that the coefficient a_1 is always 1. For this, we will make use of the vector $u^t = (1, 1, \ldots, 1)$ that was introduced in the discussion of Property 1. If v_i is an eigenvector of P with eigenvalue λ_i (which is to say that $Pv_i = \lambda_i v_i$), then the matrix product $u^t P v_i$ can be simplified in two ways. The first yields

$$u^t P v_i = (u^t P) v_i = u^t v_i,$$

and the second,

$$u^t P v_i = u^t (P v_i) = \lambda_i u^t v_i.$$

These two expressions must be equal by the associativity of matrix multiplication. For $i \geq 2$, the eigenvalue λ_i is not 1, and the equality can only hold if $u^t v_i = 0$, which expands as

$$u^t v_i = \sum_{j=1}^{N} (v_i)_j = 0,$$

where $(v_i)_j$ represents the jth coordinate of the vector v_i. This condition states that the sums of the coordinates of the vectors $v_i, i \geq 2$, must all be zero. If we now sum the components of p^0, we get 1 by hypothesis ($\sum_{i=1}^{N} p_i^0 = 1$). Thus

$$\begin{aligned} 1 = \sum_{j=1}^{N} p_j^0 &= \sum_{j=1}^{N} \sum_{i=1}^{N} a_i (v_i)_j = \sum_{i=1}^{N} a_i \sum_{j=1}^{N} (v_i)_j \\ &= a_1 \sum_{j=1}^{N} (v_1)_j = a_1 \sum_{j=1}^{N} \pi_j = a_1. \end{aligned}$$

(To obtain the second inequality we used the expression p^0 written in the basis of the eigenvectors. For the fourth, we used the fact that the sums of the coefficients of the v_i are all zero-valued except for v_1.)

To obtain the behavior after m steps, repeatedly apply the transition matrix P (m times) starting from the initial state p^0:

$$P^m p^0 = \sum_{j=1}^{N} a_j P^m v_j = \sum_{j=1}^{N} a_j \lambda_j^m v_j = a_1 v_1 + \sum_{j=2}^{N} \lambda_j^m a_j v_j = \pi + \sum_{j=2}^{N} \lambda_j^m a_j v_j.$$

Thus, the distance between the state at the mth step, $P^m p^0$, and the stationary regime π is

$$\|P^m p^0 - \pi\|^2 = \left\| \sum_{j=2}^{N} \lambda_j^m (a_j v_j) \right\|^2.$$

The sum on the right-hand side is a sum over the fixed vectors $a_j v_j$ whose coefficients diminish exponentially like λ_j^m. (Recall that the $\lambda_j, j \geq 2$, all have length less than 1.) This sum is finite, and therefore converges to zero as $m \to \infty$. Thus, $p^m = P^m p^0 \to \pi$ as $m \to \infty$. □

Return to our impartial web surfer. The properties of Markov chains can be interpreted as saying that if the impartial web surfer continues to crawl through the web long enough, he will find himself on each of the pages with a probability that approaches those given by the stationary regime π, where π is the normalized eigenvector associated with eigenvalue 1.

We are now ready to make the connection between the vector π and the *PageRank* ordering of pages.

Definition 9.5 *(1) The score given to page i in the (simplified) PageRank algorithm is the corresponding coefficient π_i from the vector π.*
(2) We sort the pages based on their PageRank scores, with the largest coming first.

The initial example with the web of five pages (Figure 9.2) allows us to obtain an understanding of this score. The norms $|\lambda_i|$ of the eigenvalues of the associated matrix P are 1 with multiplicity 1, and 0.70228 and 0.33563 each with multiplicity 2. Only the eigenvalue 1 is a real number. The eigenvector associated with the eigenvalue 1 is $(12, 16, 9, 1, 3)$, which, when normalized, yields

$$\pi = \frac{1}{41} \begin{pmatrix} 12 \\ 16 \\ 9 \\ 1 \\ 3 \end{pmatrix}.$$

This tells us that given a sufficiently long walk, the impartial web surfer would visit page B the most often, with 16 out of 41 steps leading to it. Similarly, he would nearly completely ignore page D, visiting it once per 41 steps on average.

What is the final order given to the pages? Page B is ranked number 1, which means that it is the most important page. Page A is ranked second, followed by pages C, E, and finally, the least important, page D.

There is an another way in which *PageRank* scores may be interpreted: each page gives its *PageRank* score to all of the pages it links to. Return to the vector $\pi = (\frac{12}{41}, \frac{16}{41}, \frac{9}{41}, \frac{1}{41}, \frac{3}{41})$. Page D is linked to only once, from page E. Since E has a score of $\frac{3}{41}$ and three outbound links that must share this value, D receives a final score of one-third that of E, $\frac{1}{41}$. Three pages point to page B: pages A, C, and E. The three pages have respective scores of $\frac{12}{41}$, $\frac{9}{41}$, and $\frac{3}{41}$. Page A has only one outgoing link, while pages C and E have three each. Thus, the score of page B is

$$\text{score } (B) = 1 \cdot \frac{12}{41} + \frac{1}{3} \cdot \frac{9}{41} + \frac{1}{3} \cdot \frac{3}{41} = \frac{16}{41}.$$

Why does the order implied by the *PageRank* scores give a reasonable ordering of the pages on the web? Mostly because it entrusts the users of the web itself to make the decisions as to which pages are better than others. Similarly, it ignores completely what the creator thinks of the importance of his own page. Moreover, the effect is cumulative. An important page that links to a few other pages can "transmit" its importance to these other pages. Thus, users display their confidence by linking to certain pages, and by doing so they transmit part of their score to these pages in the *PageRank* algorithm. This phenomenon has been named "*collaborative trust*" by the *PageRank* inventors.

9.3 An Improved *PageRank*

The algorithm described in the last section is not quite useable as is. There are two rather evident difficulties that must first be overcome.

The first is the existence of pages that have no outgoing links. The absence of links may come from the fact that *Google*'s web-spider has not yet indexed the destinations of the links, or that the page simply does not have any links. Thus, the impartial web crawler that arrives at this page would be forever caught there. One way of avoiding this problem is simply to ignore such pages, and remove them (and all the links leading to them) from the web. The stationary regime may then be calculated. After this is done, it is possible to assign scores to these pages by "transmitting" importance from all of the pages that link to them, as discussed at the end of the previous section:

$$\sum_{i=1}^{n} \frac{1}{l_i} r_i,$$

where l_i is the number of links issued by the ith page leading to the dead-end page, and r_i is the calculated importance of the ith page. The next problem shows that this somewhat crude approach offers only a partial solution.

The second difficulty resembles the first, but it is not quite so easy to fix. An example is depicted in the web of Figure 9.4. The web consists of the five pages from our original example, plus two others that are connected to the original web by a single link from page D. We saw in the last section that the impartial web surfer did not spend much time on page D. However, all the same, he did occasionally visit it, spending $\frac{1}{41}$ of his time there. What happens in this new modified web? Each time the web surfer visits page D he will choose to go to page A half of the time, while the other half of the time he will choose page F. If he chooses the latter option, then he can never return to the original pages A, B, C, D, or E. It is not surprising then that the stationary regime π of this new web is $\pi = (0, 0, 0, 0, 0, \frac{1}{2}, \frac{1}{2})^t$. In other words, the pages F and G "absorb" all of the importance that should have been divided up among the other pages! (Watch out! In this example, (-1) is also an eigenvalue of P, which means that P^n no longer approaches the matrix with columns π as $n \to \infty$.) Can we solve this problem as before, by simply removing the offending pages from the web? This is