Chapter Four

The Mathematics of Google's PageRank

The famous and colorful mathematician Paul Erdos (1913–96) talked about The Great Book, a make-believe book in which he imagined God kept the world's most elegant and beautiful proofs. In 2002, Graham Farmelo of London's Science Museum edited and contributed to a similar book, a book of beautiful equations. *It Must Be Beautiful: Great Equations of Modern Science* [73] is a collection of 11 essays about the greatest scientific equations, equations like $E = hf$ and $E = mc^2$. The contributing authors were invited to give their answers to the tough question of what makes an equation great. One author, Frank Wilczek, included a quote by Heinrich Hertz regarding Maxwell's equation:

> One cannot escape the feeling that these mathematical formulae have an independent existence and an intelligence of their own, that they are wiser than we are, wiser even than their discoverers, that we get more out of them than was originally put into them.

While we are not suggesting that the PageRank equation presented in this chapter,

$$\boldsymbol{\pi}^T = \boldsymbol{\pi}^T(\alpha\mathbf{S} + (1-\alpha)\mathbf{E}),$$

deserves a place in Farmelo's book alongside Einstein's theory of relativity, we do find Hertz's statement apropos. One can get a lot of mileage from the simple PageRank formula above—Google certainly has. Since beauty is in the eye of the beholder, we'll let you decide whether or not the PageRank formula deserves the adjective *beautiful*. We hope the next few chapters will convince you that it just might.

In Chapter 3, we used words to present the PageRank thesis: a page is important if it is pointed to by other important pages. It is now time to translate these words into mathematical equations. This translation reveals that the PageRank importance scores are actually the stationary values of an enormous Markov chain, and consequently Markov theory explains many interesting properties of the elegantly simple PageRank model used by Google.

This is the first of the mathematical chapters. Many of the mathematical terms in each chapter are explained in the Mathematics Chapter (Chapter 15). As terms that appear in the Mathematics Chapter are introduced in the text, they are italicized to remind you that definitions and more information can be found in Chapter 15.

4.1 THE ORIGINAL SUMMATION FORMULA FOR PAGERANK

Brin and Page, the inventors of PageRank,[1] began with a simple summation equation, the roots of which actually derive from bibliometrics research, the analysis of the citation structure among academic papers. The PageRank of a page P_i, denoted $r(P_i)$, is the sum of the PageRanks of all pages pointing into P_i.

$$r(P_i) = \sum_{P_j \in B_{P_i}} \frac{r(P_j)}{|P_j|}, \tag{4.1.1}$$

where B_{P_i} is the set of pages pointing into P_i (backlinking to P_i in Brin and Page's words) and $|P_j|$ is the number of outlinks from page P_j. Notice that the PageRank of inlinking pages $r(P_j)$ in equation (4.1.1)) is tempered by the number of recommendations made by P_j, denoted $|P_j|$. The problem with equation (4.1.1) is that the $r(P_j)$ values, the PageRanks of pages inlinking to page P_i, are unknown. To sidestep this problem, Brin and Page used an iterative procedure. That is, they assumed that, in the beginning, all pages have equal PageRank (of say, $1/n$, where n is the number of pages in Google's index of the Web). Now the rule in equation (4.1.1) is followed to compute $r(P_i)$ for each page P_i in the index. The rule in equation (4.1.1) is successively applied, substituting the values of the previous iterate into $r(P_j)$. We introduce some more notation in order to define this *iterative procedure*. Let $r_{k+1}(P_i)$ be the PageRank of page P_i at iteration $k+1$. Then,

$$r_{k+1}(P_i) = \sum_{P_j \in B_{P_i}} \frac{r_k(P_j)}{|P_j|}. \tag{4.1.2}$$

This process is initiated with $r_0(P_i) = 1/n$ for all pages P_i and repeated with the hope that the PageRank scores will eventually converge to some final stable values. Applying equation (4.1.2) to the tiny web of Figure 4.1 gives the following values for the PageRanks after a few iterations.

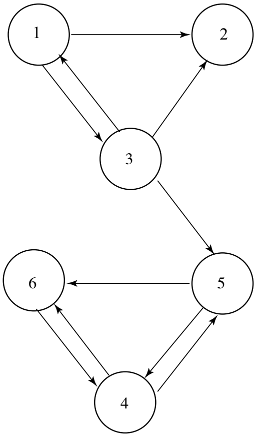

Figure 4.1 Directed graph representing web of six pages

[1] The patent for PageRank was filed in 1998 by Larry Page and granted in 2001 (US Patent #6285999), and thus the name PageRank has a double reference to both webpages and one of its founding fathers.

Table 4.1 First few iterates using (4.1.2) on Figure 4.1

Iteration 0	Iteration 1	Iteration 2	Rank at Iter. 2
$r_0(P_1) = 1/6$	$r_1(P_1) = 1/18$	$r_2(P_1) = 1/36$	5
$r_0(P_2) = 1/6$	$r_1(P_2) = 5/36$	$r_2(P_2) = 1/18$	4
$r_0(P_3) = 1/6$	$r_1(P_3) = 1/12$	$r_2(P_3) = 1/36$	5
$r_0(P_4) = 1/6$	$r_1(P_4) = 1/4$	$r_2(P_4) = 17/72$	1
$r_0(P_5) = 1/6$	$r_1(P_5) = 5/36$	$r_2(P_5) = 11/72$	3
$r_0(P_6) = 1/6$	$r_1(P_6) = 1/6$	$r_2(P_6) = 14/72$	2

4.2 MATRIX REPRESENTATION OF THE SUMMATION EQUATIONS

Equations (4.1.1) and (4.1.2) compute PageRank one page at a time. Using *matrices*, we replace the tedious $\sum$ symbol, and at each iteration, compute a PageRank vector, which uses a single $1 \times n$ *vector* to hold the PageRank values for all pages in the index. In order to do this, we introduce an $n \times n$ matrix $\mathbf{H}$ and a $1 \times n$ row vector $\boldsymbol{\pi}^T$. The matrix $\mathbf{H}$ is a row normalized *hyperlink* matrix with $\mathbf{H}_{ij} = 1/|P_i|$ if there is a link from node i to node j, and 0, otherwise. Although $\mathbf{H}$ has the same nonzero structure as the binary *adjacency matrix* for the graph (called L in the Matlab Crawler m-file on page 17), its nonzero elements are probabilities. Consider once again the tiny web graph of Figure 4.1.

The $\mathbf{H}$ matrix for this graph is

$$\mathbf{H} = \begin{array}{c} \\ P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{array} \begin{array}{c} \begin{array}{cccccc} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}.$$

The nonzero elements of row i correspond to the outlinking pages of page i, whereas the nonzero elements of column i correspond to the inlinking pages of page i. We now introduce a row vector $\boldsymbol{\pi}^{(k)T}$, which is the PageRank vector at the k^{th} iteration. Using this matrix notation, equation (4.1.2) can be written compactly as

$$\boldsymbol{\pi}^{(k+1)T} = \boldsymbol{\pi}^{(k)T}\mathbf{H}. \tag{4.2.1}$$

If you like, verify with the example of Figure 4.1 that the iterates of equation (4.2.1) match those of equation (4.1.2).

Matrix equation (4.2.1) yields some immediate observations.

1. Each iteration of equation (4.2.1) involves one vector-matrix multiplication, which generally requires $O(n^2)$ *computation*, where n is the size of the square matrix $\mathbf{H}$.

2. $\mathbf{H}$ is a very *sparse* matrix (a large proportion of its elements are 0) because most webpages link to only a handful of other pages. Sparse matrices, such as the one shown in Figure 4.2, are welcome for several reasons. First, they require minimal storage, since sparse storage schemes, which store only the nonzero elements of the

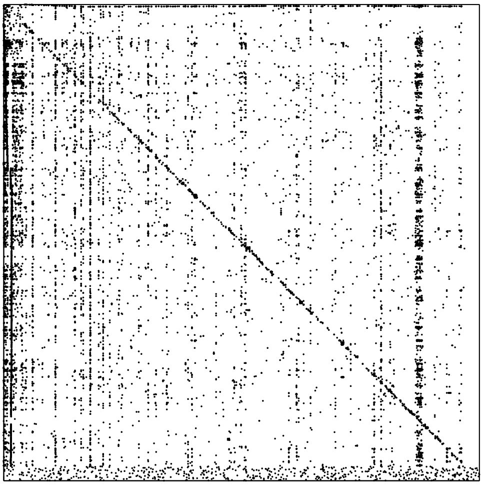

Figure 4.2 Example of a sparse matrix. The nonzero elements are indicated by pixels.

matrix and their location [145], exist. Second, vector-matrix multiplication involving a sparse matrix requires much less effort than the $O(n^2)$ *dense* computation. In fact, it requires $O(nnz(\mathbf{H}))$ computation, where $nnz(\mathbf{H})$ is the number of nonzeros in $\mathbf{H}$. Estimates show that the average webpage has about 10 outlinks, which means that $\mathbf{H}$ has about $10n$ nonzeros, as opposed to the n^2 nonzeros in a completely dense matrix. This means that the vector-matrix multiplication of equation (4.2.1) reduces to $O(n)$ effort.

3. The iterative process of equation (4.2.1) is a simple *linear stationary process* of the form studied in most numerical analysis classes [82, 127]. In fact, it is the classical *power method* applied to $\mathbf{H}$.

4. $\mathbf{H}$ looks a lot like a *stochastic transition probability matrix* for a *Markov chain*. The **dangling nodes** of the network, those nodes with no outlinks, create 0 rows in the matrix. All the other rows, which correspond to the nondangling nodes, create stochastic rows. Thus, $\mathbf{H}$ is called *substochastic*.

These four observations are important to the development and execution of the PageRank model, and we will return to them throughout the chapter. For now, we spend more time examining the iterative matrix equation (4.2.1).

4.3 PROBLEMS WITH THE ITERATIVE PROCESS

Equation (4.2.1) probably caused readers, especially our mathematical readers, to ask several questions. For example,

- Will this iterative process continue indefinitely or will it converge?
- Under what circumstances or properties of $\mathbf{H}$ is it guaranteed to converge?

- Will it converge to something that makes sense in the context of the PageRank problem?
- Will it converge to just one vector or multiple vectors?
- Does the convergence depend on the starting vector $\boldsymbol{\pi}^{(0)T}$?
- If it will converge eventually, how long is "eventually"? That is, how many iterations can we expect until convergence?

We'll answer these questions in the next few sections. However, our answers depend on how Brin and Page chose to resolve some of the problems they encountered with their equation (4.2.1).

Brin and Page originally started the iterative process with $\boldsymbol{\pi}^{(0)T} = 1/n\,\mathbf{e}^T$, where $\mathbf{e}^T$ is the row vector of all 1s. They immediately ran into several problems when using equation (4.2.1) with this initial vector. For example, there is the problem of **rank sinks**, those pages that accumulate more and more PageRank at each iteration, monopolizing the scores and refusing to share. In the simple example of Figure 4.3, the dangling node 3 is a rank sink. In the more complicated example of Figure 4.1, the cluster of nodes 4, 5, and 6 conspire to hoard PageRank. After just 13 iterations of equation (4.2.1), $\boldsymbol{\pi}^{(13)T} = (\,0 \quad 0 \quad 0 \quad 2/3 \quad 1/3 \quad 1/5\,)$. This conspiring can be malicious or inadvertent. (See the asides on search engine optimization and link farms on pages 43 and 52, respectively.) The example with $\boldsymbol{\pi}^{(13)T}$ also shows another problem caused by sinks. As nodes hoard PageRank, some nodes may be left with none. Thus, ranking nodes by their PageRank values is tough when a majority of the nodes are tied with PageRank 0. Ideally, we'd prefer the PageRank vector to be positive, i.e., contain all positive values.

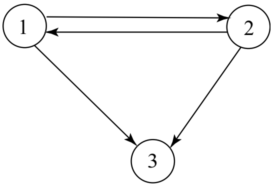

Figure 4.3 Simple graph with rank sink

There's also the problem of **cycles**. Consider the simplest case in Figure 4.4. Page 1 only points to page 2 and vice versa, creating an infinite loop or cycle. Suppose the iterative process of equation (4.2.1) is run with $\boldsymbol{\pi}^{(0)T} = (\,1 \quad 0\,)$. The iterates will not converge no matter how long the process is run. The iterates $\boldsymbol{\pi}^{(k)T}$ flip-flop indefinitely between $(\,1 \quad 0\,)$ when k is even and $(\,0 \quad 1\,)$ when k is odd.

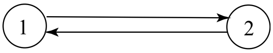

Figure 4.4 Simple graph with cycle

4.4 A LITTLE MARKOV CHAIN THEORY

Before we get to Brin and Page's adjustments to equation (4.2.1), which solve the problems of the previous section, we pause to introduce a little theory for Markov chains. (We urge readers who are less familiar with Markov chains to read the Mathematics Chapter, Chapter 15, before proceeding.) In observations 3 and 4, we noted that equation (4.2.1) resembled the power method applied to a Markov chain with *transition probability matrix* $\mathbf{H}$. These observations are very helpful because the theory of Markov chains is well developed,[2] and very applicable to the PageRank problem. With Markov theory we can make adjustments to equation (4.2.1) that insure desirable results, convergence properties, and encouraging answers to the questions on page 34. In particular, we know that, for any starting vector, the power method applied to a Markov matrix $\mathbf{P}$ converges to a unique positive vector called the *stationary vector* as long as $\mathbf{P}$ is *stochastic, irreducible*, and *aperiodic*. (Aperiodicity plus irreducibility implies primitivity.) Therefore, the PageRank convergence problems caused by sinks and cycles can be overcome if $\mathbf{H}$ is modified slightly so that it is a Markov matrix with these desired properties.

Markov properties affecting PageRank

A unique positive PageRank vector exists when the Google matrix is stochastic and irreducible. Further, with the additional property of aperiodicity, the power method will converge to this PageRank vector, regardless of the starting vector for the iterative process.

4.5 EARLY ADJUSTMENTS TO THE BASIC MODEL

In fact, this is exactly what Brin and Page did. They describe their adjustments to the basic PageRank model in their original 1998 papers. It is interesting to note that none of their papers used the phrase "Markov chain," not even once. Although, most surely, if they were unaware of it in 1998, they now know the connection their original model has to Markov chains, as Markov chain researchers have excitedly and steadily jumped on the PageRank bandwagon, eager to work on what some call the grand application of Markov chains.

Rather than using Markov chains and their properties to describe their adjustments, Brin and Page use the notion of a **random surfer**. Imagine a web surfer who bounces along randomly following the hyperlink structure of the Web. That is, when he arrives at a page with several outlinks, he chooses one at random, hyperlinks to this new page, and continues this random decision process indefinitely. In the long run, the proportion of time the random surfer spends on a given page is a measure of the relative importance of that page. If he spends a large proportion of his time on a particular page, then he must have, in randomly following the hyperlink structure of the Web, repeatedly found himself returning to that page. Pages that he revisits often must be important, because they must be pointed to by other important pages. Unfortunately, this random surfer encounters some problems. He gets caught whenever he enters a dangling node. And on the Web there are plenty of nodes dangling, e.g., pdf files, image files, data tables, etc. To fix this, Brin and Page define

[2] Almost 100 years ago in 1906, Andrei Andreyevich Markov invented the chains that after 1926 bore his name [20].

their first adjustment, which we call the **stochasticity adjustment** because the $\mathbf{0}^T$ rows of $\mathbf{H}$ are replaced with $1/n\,\mathbf{e}^T$, thereby making $\mathbf{H}$ stochastic. As a result, the random surfer, after entering a dangling node, can now hyperlink to any page at random. For the tiny 6-node web of Figure 4.1, the **stochastic matrix** called $\mathbf{S}$ is

$$\mathbf{S} = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

Writing this stochasticity fix mathematically reveals that $\mathbf{S}$ is created from a *rank-one update* to $\mathbf{H}$. That is, $\mathbf{S} = \mathbf{H} + \mathbf{a}(1/n\,\mathbf{e}^T)$, where $a_i = 1$ if page i is a dangling node and 0 otherwise. The binary vector $\mathbf{a}$ is called the **dangling node vector**. $\mathbf{S}$ is a combination of the raw original hyperlink matrix $\mathbf{H}$ and a rank-one matrix $1/n\,\mathbf{a}\mathbf{e}^T$.

This adjustment guarantees that $\mathbf{S}$ is stochastic, and thus, is the transition probability matrix for a Markov chain. However, it alone cannot guarantee the convergence results desired. (That is, that a unique positive $\boldsymbol{\pi}^T$ exists and that equation (4.2.1) will converge to this $\boldsymbol{\pi}^T$ quickly.) Brin and Page needed another adjustment–this time a **primitivity adjustment**. With this adjustment, the resulting matrix is stochastic and *primitive*. A primitive matrix is both irreducible and aperiodic. Thus, the stationary vector of the chain (which is the PageRank vector in this case) exists, is unique, and can be found by a simple power iteration. Brin and Page once again use the random surfer to describe these Markov properties.

The random surfer argument for the primitivity adjustment goes like this. While it is true that surfers follow the hyperlink structure of the Web, at times they get bored and abandon the hyperlink method of surfing by entering a new destination in the browser's URL line. When this happens, the random surfer, like a Star Trek character, "teleports" to the new page, where he begins hyperlink surfing again, until the next teleportation, and so on. To model this activity mathematically, Brin and Page invented a new matrix $\mathbf{G}$, such that

$$\mathbf{G} = \alpha\mathbf{S} + (1-\alpha)1/n\,\mathbf{e}\mathbf{e}^T,$$

where α is a scalar between 0 and 1. $\mathbf{G}$ is called the **Google matrix**. In this model, α is a parameter that controls the proportion of time the random surfer follows the hyperlinks as opposed to teleporting. Suppose $\alpha = .6$. Then 60% of the time the random surfer follows the hyperlink structure of the Web and the other 40% of the time he teleports to a random new page. The teleporting is random because the teleportation matrix $\mathbf{E} = 1/n\,\mathbf{e}\mathbf{e}^T$ is uniform, meaning the surfer is equally likely, when teleporting, to jump to any page.

There are several consequences of the primitivity adjustment.

- $\mathbf{G}$ is *stochastic*. It is the *convex combination* of the two stochastic matrices $\mathbf{S}$ and $\mathbf{E} = 1/n\,\mathbf{e}\mathbf{e}^T$.

- $\mathbf{G}$ is *irreducible*. Every page is directly connected to every other page, so irreducibility is trivially enforced.
- $\mathbf{G}$ is *aperiodic*. The self-loops ($\mathbf{G}_{ii} > 0$ for all i) create aperiodicity.
- $\mathbf{G}$ is *primitive* because $\mathbf{G}^k > 0$ for some k. (In fact, this holds for $k = 1$.) This implies that a unique positive $\boldsymbol{\pi}^T$ exists, and the power method applied to $\mathbf{G}$ is guaranteed to converge to this vector.
- $\mathbf{G}$ is completely *dense*, which is a very bad thing, computationally. Fortunately, $\mathbf{G}$ can be written as a *rank-one update* to the very sparse hyperlink matrix $\mathbf{H}$. This is computationally advantageous, as we show later in section 4.6.

$$\begin{aligned}\mathbf{G} &= \alpha\mathbf{S} + (1-\alpha)1/n\,\mathbf{e}\mathbf{e}^T \\ &= \alpha(\mathbf{H} + 1/n\,\mathbf{a}\mathbf{e}^T) + (1-\alpha)\,1/n\,\mathbf{e}\mathbf{e}^T \\ &= \alpha\mathbf{H} + (\alpha\mathbf{a} + (1-\alpha)\mathbf{e})\,1/n\,\mathbf{e}^T.\end{aligned}$$

- $\mathbf{G}$ is artificial in the sense that the raw hyperlink matrix $\mathbf{H}$ has been twice modified in order to produce desirable convergence properties. A stationary vector (thus, a PageRank vector) does not exist for $\mathbf{H}$, so Brin and Page creatively cheated to achieve their desired result. For the twice-modified $\mathbf{G}$, a unique PageRank vector exists, and as it turns out, this vector is remarkably good at giving a global importance value to webpages.

Notation for the PageRank Problem

$\mathbf{H}$	very sparse, raw substochastic hyperlink matrix
$\mathbf{S}$	sparse, stochastic, most likely reducible matrix
$\mathbf{G}$	completely dense, stochastic, primitive matrix called the Google Matrix
$\mathbf{E}$	completely dense, rank-one teleportation matrix
n	number of pages in the engine's index = order of $\mathbf{H}$, $\mathbf{S}$, $\mathbf{G}$, $\mathbf{E}$
α	scaling parameter between 0 and 1
$\boldsymbol{\pi}^T$	stationary row vector of $\mathbf{G}$ called the PageRank vector
$\mathbf{a}^T$	binary dangling node vector

In summary, Google's adjusted PageRank method is

$$\boldsymbol{\pi}^{(k+1)T} = \boldsymbol{\pi}^{(k)T}\mathbf{G}, \tag{4.5.1}$$

which is simply the *power method* applied to $\mathbf{G}$.

We close this section with an example. Returning again to Figure 4.1, for $\alpha = .9$, the stochastic, primitive matrix $\mathbf{G}$ is

$$\mathbf{G} = .9\mathbf{H} + (.9\begin{pmatrix}0\\1\\0\\0\\0\\0\end{pmatrix} + .1\begin{pmatrix}1\\1\\1\\1\\1\\1\end{pmatrix})\,1/6\,(\,1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1\,)$$

$$= \begin{pmatrix} 1/60 & 7/15 & 7/15 & 1/60 & 1/60 & 1/60 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 19/60 & 19/60 & 1/60 & 1/60 & 19/60 & 1/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 7/15 & 7/15 \\ 1/60 & 1/60 & 1/60 & 7/15 & 1/60 & 7/15 \\ 1/60 & 1/60 & 1/60 & 11/12 & 1/60 & 1/60 \end{pmatrix}.$$

Google's PageRank vector is the stationary vector of $\mathbf{G}$ and is given by

$$\begin{array}{cc} & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \boldsymbol{\pi}^T = & \begin{pmatrix} .03721 & .05396 & .04151 & .3751 & .206 & .2862 \end{pmatrix}. \end{array}$$

The interpretation of $\pi_1 = .03721$ is that 3.721% of the time the random surfer visits page 1. Therefore, the pages in this tiny web can be ranked by their importance as $(\,4 \quad 6 \quad 5 \quad 2 \quad 3 \quad 1\,)$, meaning page 4 is the most important page and page 1 is the least important page, according to the PageRank definition of importance.

4.6 COMPUTATION OF THE PAGERANK VECTOR

The PageRank problem can be stated in two ways:

1. Solve the following *eigenvector* problem for $\boldsymbol{\pi}^T$.

$$\begin{aligned} \boldsymbol{\pi}^T &= \boldsymbol{\pi}^T\mathbf{G}, \\ \boldsymbol{\pi}^T\mathbf{e} &= 1. \end{aligned}$$

2. Solve the following *linear homogeneous system* for $\boldsymbol{\pi}^T$.

$$\begin{aligned} \boldsymbol{\pi}^T(\mathbf{I} - \mathbf{G}) &= \mathbf{0}^T, \\ \boldsymbol{\pi}^T\mathbf{e} &= 1. \end{aligned}$$

In the first system, the goal is to find the normalized *dominant left-hand eigenvector* of $\mathbf{G}$ corresponding to the *dominant eigenvalue* $\lambda_1 = 1$. ($\mathbf{G}$ is a stochastic matrix, so $\lambda_1 = 1$.) In the second system, the goal is to find the normalized left-hand null vector of $\mathbf{I} - \mathbf{G}$. Both systems are subject to the normalization equation $\boldsymbol{\pi}^T\mathbf{e} = 1$, which insures that $\boldsymbol{\pi}^T$ is a probability vector. In the example in section 4.5, $\mathbf{G}$ is a 6×6 matrix, so we used Matlab's `eig` command to solve for $\boldsymbol{\pi}^T$, then normalized the result (by dividing the vector by its sum) to get the PageRank vector. However, for a web-sized matrix like Google's, this will not do. Other more advanced and computationally efficient methods must be used. Of course, $\boldsymbol{\pi}^T$ is the stationary vector of a Markov chain with transition matrix $\mathbf{G}$, and much research has been done on computing the stationary vector for a general Markov chain. See William J. Stewart's book *Introduction to the Numerical Solution of Markov Chains* [154], which contains over a dozen methods for finding $\boldsymbol{\pi}^T$. However, the specific features of the

PageRank matrix $\mathbf{G}$ make one numerical method, the power method, the clear favorite. In this section, we discuss the power method, which is the original method proposed by Brin and Page for finding the PageRank vector. We describe other more advanced methods in Chapter 9.

The World's Largest Matrix Computation

Cleve Moler, the founder of Matlab, wrote an article [131] for his October 2002 newsletter *Matlab News* that cited PageRank as "The World's Largest Matrix Computation." Then Google was applying the power method to a sparse matrix of order 2.7 billion. Now it's up to 8.1 billion!

The power method is one of the oldest and simplest iterative methods for finding the *dominant eigenvalue and eigenvector* of a matrix.[3] Therefore, it can be used to find the stationary vector of a Markov chain. (The stationary vector is simply the dominant left-hand eigenvector of the Markov matrix.) However, the power method is known for its tortoise-like speed. Of the available iterative methods (Gauss-Seidel, Jacobi, restarted GMRES, BICGSTAB, etc. [18]), the power method is generally the slowest. So why did Brin and Page choose a method known for its sluggishness? There are several good reasons for their choice.

First, the power method is simple. The implementation and programming are elementary. (See the box on page 42 for a Matlab implementation of the PageRank power method.) In addition, the power method applied to $\mathbf{G}$ (equation (4.5.1)) can actually be expressed in terms of the very sparse $\mathbf{H}$.

$$\begin{aligned}\boldsymbol{\pi}^{(k+1)T} &= \boldsymbol{\pi}^{(k)T}\mathbf{G}\\ &= \alpha\,\boldsymbol{\pi}^{(k)T}\mathbf{S} + \frac{1-\alpha}{n}\,\boldsymbol{\pi}^{(k)T}\,\mathbf{e}\,\mathbf{e}^T\\ &= \alpha\,\boldsymbol{\pi}^{(k)T}\mathbf{H} + (\alpha\,\boldsymbol{\pi}^{(k)T}\mathbf{a} + 1 - \alpha)\,\mathbf{e}^T/n. \qquad (4.6.1)\end{aligned}$$

The vector-matrix multiplications ($\boldsymbol{\pi}^{(k)T}\mathbf{H}$) are executed on the extremely sparse $\mathbf{H}$, and $\mathbf{S}$ and $\mathbf{G}$ are never formed or stored, only their rank-one components, $\mathbf{a}$ and $\mathbf{e}$, are needed. Recall that each vector-matrix multiplication is $O(n)$ since $\mathbf{H}$ has about 10 nonzeros per row. This is probably the main reason for Brin and Page's use of the power method in 1998. But why is the power method still the predominant method in PageRank research papers today, and why have most improvements been novel modifications to the PageRank power method, rather than experiments with other methods? The other advantages of the PageRank power method answer these questions.

The power method, like many other iterative methods, is matrix-free, which is a term that refers to the storage and handling of the coefficient matrix. For matrix-free methods, the coefficient matrix is only accessed through the vector-matrix multiplication routine. No manipulation of the matrix is done. Contrast this with direct methods, which manipulate elements of the matrix during each step. Modifying and storing elements of the Google

[3] The power method goes back at least to 1913. With the help of James H. Wilkinson, the power method became the standard method in the 1960s for finding the eigenvalues and eigenvectors of a matrix with a digital computer [152, p. 69–70].

matrix is not feasible. Even though $\mathbf{H}$ is very sparse, its enormous size and lack of structure preclude the use of direct methods. Instead, matrix-free methods, such as the class of iterative methods, are preferred.

The power method is also storage-friendly. In addition to the sparse matrix $\mathbf{H}$ and the dangling node vector $\mathbf{a}$, only one vector, the current iterate $\boldsymbol{\pi}^{(k)T}$, must be stored. This vector is completely dense, meaning n real numbers must be stored. For Google, $n = 8.1$ billion, so one can understand their frugal mentality when it comes to storage. Other iterative methods, such as GMRES or BICGSTAB, while faster, require the storage of multiple vectors. For example, a restarted GMRES(10) requires the storage of 10 vectors of length n at each iteration, which is equivalent to the amount of storage required by the entire $\mathbf{H}$ matrix, since $nnz(\mathbf{H}) \approx 10n$.

The last reason for using the power method to compute the PageRank vector concerns the number of iterations it requires. Brin and Page reported in their 1998 papers, and others have confirmed, that only 50-100 power iterations are needed before the iterates have converged, giving a satisfactory approximation to the exact PageRank vector. Recall that each iteration of the power method requires $O(n)$ effort because $\mathbf{H}$ is so sparse. As a result, it's hard to find a method that can beat 50 $O(n)$ power iterations. Algorithms whose run time and computational effort are linear (or sublinear) in the problem size are very fast, and rare.

The next logical question is: why does the power method applied to $\mathbf{G}$ require only about 50 iterations to converge? Is there something about the structure of $\mathbf{G}$ that indicates this speedy convergence? The answer comes from the theory of Markov chains. In general, the *asymptotic rate of convergence* of the power method applied to a matrix depends on the ratio of the two eigenvalues that are largest in magnitude, denoted λ_1 and λ_2. Precisely, the asymptotic convergence rate is the rate at which $|\lambda_2/\lambda_1|^k \to 0$. For stochastic matrices such as $\mathbf{G}$, $\lambda_1 = 1$, so $|\lambda_2|$ governs the convergence. Since $\mathbf{G}$ is also primitive, $|\lambda_2| < 1$. In general, numerically finding λ_2 for a matrix requires computational effort that one is not willing to spend just to get an estimate of the asymptotic rate of convergence. Fortunately, for the PageRank problem, it's easy to show [127, p. 502], [90, 108] that if the respective spectrums are $\sigma(\mathbf{S}) = \{1, \mu_2, \ldots, \mu_n\}$ and $\sigma(\mathbf{G}) = \{1, \lambda_2, \ldots, \lambda_n\}$, then

$$\lambda_k = \alpha\mu_k \quad \text{for} \quad k = 2, 3, \ldots, n.$$

(A short proof of this statement is provided at the end of this chapter.) Furthermore, the link structure of the Web makes it very likely that $|\mu_2| = 1$ (or at least $|\mu_2| \approx 1$), which means that $|\lambda_2(\mathbf{G})| = \alpha$ (or $|\lambda_2(\mathbf{G})| \approx \alpha$). As a result, the convex combination parameter α explains the reported convergence after just 50 iterations. In their papers, Google founders Brin and Page use $\alpha = .85$, and at last report, this is still the value used by Google. $\alpha^{50} = .85^{50} \approx .000296$, which implies that at the 50th iteration one can expect roughly 2-3 *places of accuracy* in the approximate PageRank vector. This degree of accuracy is apparently adequate for Google's ranking needs. Mathematically, ten places of accuracy may be needed to distinguish between elements of the PageRank vector (see Section 8.3), but when PageRank scores are combined with content scores, high accuracy may be less important.

Subdominant Eigenvalue of the Google Matrix

For the Google matrix $\mathbf{G} = \alpha\mathbf{S} + (1-\alpha)1/n\,\mathbf{e}\mathbf{e}^T$,

$$|\lambda_2(\mathbf{G})| \leq \alpha.$$

- For the case when $|\lambda_2(\mathbf{S})| = 1$ (which occurs often due to the reducibility of the web graph), $|\lambda_2(\mathbf{G})| = \alpha$. Therefore, the asymptotic rate of convergence of the PageRank power method of equation (4.6.1) is the rate at which $\alpha^k \rightarrow 0$.

We can now give positive answers to the six questions of section 4.3. With the stochasticity and primitivity adjustments, the power method applied to $\mathbf{G}$ is guaranteed to converge to a unique positive vector called the PageRank vector, regardless of the starting vector. Because the resulting PageRank vector is positive, there are no undesirable ties at 0. Further, to produce PageRank scores with approximately τ digits of accuracy about $-\tau/log_{10}\alpha$ iterations must be completed.

Matlab m-file for PageRank Power Method

This m-file is a Matlab implementation of the PageRank power method given in equation (4.6.1).

```
function [pi,time,numiter]=\hbox{PageRank}(pi0,H,n,alpha,epsilon);

% \hbox{PageRank}  computes the \hbox{PageRank} vector for an n-by-n Markov
%            matrix H with starting vector pi0 (a row vector)
%            and scaling parameter alpha (scalar).  Uses power
%            method.
%
% EXAMPLE: [pi,time,numiter]=\hbox{PageRank}(pi0,H,1000,.9,1e-8);
%
% INPUT:  pi0 = starting vector at iteration 0 (a row vector)
%         H = row-normalized hyperlink matrix (n-by-n sparse matrix)
%         n = size of H matrix (scalar)
%         alpha = scaling parameter in \hbox{PageRank} model (scalar)
%         epsilon = convergence tolerance (scalar, e.g. 1e-8)
%
% OUTPUT:   pi = \hbox{PageRank} vector
%           time = time required to compute \hbox{PageRank} vector
%           numiter = number of iterations until convergence
%
% The starting vector is usually set to the uniform vector,
% pi0=1/n*ones(1,n).
% NOTE: Matlab stores sparse matrices by columns, so it is faster
%       to do some operations on H', the transpose of H.
% get "a", the dangling node vector, where a(i)=1, if node i
%   is dangling node and 0, o.w.

rowsumvector=ones(1,n)*H';
nonzerorows=find(rowsumvector);
zerorows=setdiff(1:n,nonzerorows); l=length(zerorows);
a=sparse(zerorows,ones(l,1),ones(l,1),n,1);


k=0;
residual=1;
pi=pi0;
tic;

while (residual >= epsilon)
  prevpi=pi;
  k=k+1;
  pi=alpha*pi*H + (alpha*(pi*a)+1-alpha)*((1/n)*ones(1,n));
  residual=norm(pi-prevpi,1);
end
numiter=k;
time=toc;
```

Search within a Site

In the competitive business of search, Google is refreshingly generous at times. For example, at no charge, Google lets website authors employ its technology to search within their site. (Clicking on the "more" button on Google's home page will lead you to the latest information on their services.) For queries within a site, Google restricts the set of relevant pages to only in-site pages. These in-site relevant pages are then ranked using the global PageRank scores. In essence, this in-site search extracts the site from Google's massive index of billions of pages and untangles the part of the Web pertaining to the site. Looking at an individual subweb makes for a much more manageable hyperlink graph.

ASIDE: Search Engine Optimization

As more and more sales move online, large and small businesses alike turn to search engine optimizers (SEOs) to help them boost profits. SEOs carefully craft webpages and links in order to "optimize" the chances that their clients' pages will appear in the first few pages of search engine results. SEOs can be classified as ethical or unethical. Ethical SEOs are good netizens, citizens of the net, who offer only sound advice, such as the best way to display text and label pictures and tags. They encourage webpage authors to maintain good content, as page rankings are the combination of the content score and the popularity score. They also warn authors that search engines punish pages they perceive as deliberately spamming. Ethical SEOs and search engines consider themselves partners who, by exchanging information and tips, together improve search quality. Unethical SEOs, on the other hand, intentionally try to outwit search engines and promote spamming techniques. See the aside on page 52 for a specific case of unethical SEO practices. Since the Web's infancy, search engines have been *embroiled in an eternal battle with unethical SEOs. The battle rages all over the Web, from visible webpage content to hidden metatags, from links to anchor text, and from inside servers to out on link farms (again, see the aside on page 52).*

SEOs had success against the early search engines by using term spamming and hiding techniques [84]. In term spamming, spam words are included in the body of the page, often times repeatedly, in the title, metatags, anchor text, and URL text. Hiding techniques use color schemes and cloaking to deceive search engines. For example, using white text on a white background makes spam invisible to human readers, which means search engines are less likely to receive helpful complaints about pages with hidden spam. Cloaking refers to the technique of returning one spam-loaded webpage for normal user requests and another spam-free page for requests from search engine crawlers. As long as authors can clearly identify web crawling agents, they can send the agent away with a clean, spam-free page. Because these techniques are so easy for webpage authors to use, search engines had to retaliate. They did so by increasing the IQ of their spiders and indexers. Many spiders and indexers are trained to ignore metatags, since by the late 1990s these rarely held accurate page information. They also ignore repeated keywords. However, cloaking is harder to counteract. Search engines request help from users to stop cloaking. For example, Google asks surfers to act as referees and to blow the whistle whenever they find a suspicious page that instantaneously redirects them to a new page.

In 1998, search engines added link analysis to their bag of tricks. As a result, content spam and cloaking alone could no longer fool the link analysis engines and garner spammers unjustifiably high rankings. Spammers and SEOs adapted by learning how link analysis works. The SEO community has always been active—its members, then and now, hold conferences, write papers and books, host weblogs, and sell their secrets. The most famous and informative SEO papers were written by Chris Ridings, "PageRank explained: Everything you've always wanted to know about PageRank" [143] and "PageRank uncovered" [144]. These papers offer practical strategies for hoarding PageRank and avoiding such undesirable things as PageRank leak. Search engines constantly tune their algorithms in order to stay one step ahead of the SEO gamers. While search engines consider unethical SEOs to be adversaries, some web analysts call them an essential part of the web food chain, because they drive innovation and research and development.

ASIDE: How Do Search Engines Make Money?

We are asked this question often. It's a good question. Search engines provide free and unlimited access to their services, so just where do the billions of dollars in search revenue come from? Search engines have multiple sources of income. First, there's the inclusion fee that some search engines charge website authors. Some impatient authors want a guarantee that their new site will be indexed soon (in a day or two) rather than in a month or two, when a spider finally gets to it in the to-be-crawled URL list. Search engines supply this guarantee for a small fee, and for a slightly larger fee, authors can guarantee that their site be reindexed on a more frequent, perhaps monthly, basis.

Most search engines also generate revenue by selling profile data to interested parties. Search engines collect enormous amounts of user data on a daily basis. This data are used to improve the quality of search and predict user needs, but it is also sold in an aggregate form to various companies. For example, search engine optimization companies who are interested in popular query words or the percentage of searches that are commercial in nature can buy this information directly from a search engine.

While search engines do not sell access to their search capabilities to individual users, they do sell search services to companies. For example, Netscape pays Google to use Google search as the default search provided by its browser. At one point, GoTo (which was bought by Overture, which is now part of Yahoo) sold its top seven results for each query term to Yahoo and AltaVista, who, in turn, used the seven results as their top results.

Despite these sources of income, by far the most profitable and fastest-growing revenue source for search engines is advertising. It is estimated that in 2004 $3 billion in search revenue will be generated from advertising. Google's IPO filing on June 21, 2004 made the company's dependence on advertising very clear: advertising accounted for over 97% of their 2003 revenue. Many search engines sell banner ads that appear on their homepages and results pages. Others sell pay-for-placement ads. These controversial ads allow a company to buy their way to the top of the ranking. Many web analysts argue that these pay-for-placement ads pollute the search results. However, search engines using this technique (GoTo is a prime example) retort that this method of ranking is excellent for commercial searches. Since recent surveys estimate that 15-30% of all searches are commercial in nature, engines like Overture provide a valuable service for this class of queries. On the other hand, many searches are research-oriented, and the results of pay-for-placement engines frustrate these users.

Google takes a different approach to advertisements and rankings. They present the unpaid results in a main list while pay-for-placement sites appear separately on the side as "sponsored links." Google, and now Yahoo, are the only remaining companies not to mingle paid links with pure links. Google uses a cost-per-click advertising scheme to present sponsored links. Companies choose a keyword associated with their product or service, and then bid on a price they are willing to pay each time a searcher clicks on their link. For example, a bike shop in Raleigh may bid 5 cents for every query on "bike Raleigh." The bike shop is billed only if a searcher actually clicks on their ad. However, another company may bid 17 cents for the same query. The ad for the second company is likely to appear first because, although there is some fine tuning and optimization, sponsored ads generally are listed in order from the highest bid to the lowest bid.

Cost-per-click advertising is an innovation in marketing. Small businesses who traditionally spent little on advertising are now spending much more on web advertising because cost-per-click advertising is so cost-effective. If a searcher clicks on the link, he or she is indicating an intent to buy, something that other means of advertising such as billboards or mail circulars cannot deliver. Interestingly, like many other things on the Web, it was only a matter of time before cost-per-click advertising turned into a battleground between competitors. Without protection (which can be purchased in the form of a software program) naive companies buying cost-per-click advertising can easily be sabotaged by competitors. Competitors repeatedly click on the naive company's ads, running up their tab and exhausting the company's advertising budget.

4.7 THEOREM AND PROOF FOR SPECTRUM OF THE GOOGLE MATRIX

In this chapter, we defined the Google matrix as $\mathbf{G} = \alpha\mathbf{S} + (1-\alpha)1/n\,\mathbf{e}\mathbf{e}^T$. However, in the Section 5.3 of the next chapter, we broaden this to include a more general Google matrix, where the fudge factor matrix $\mathbf{E}$ changes from the uniform matrix $1/n\,\mathbf{e}\mathbf{e}^T$ to $\mathbf{e}\mathbf{v}^T$, where $\mathbf{v}^T > \mathbf{0}$ is a probability vector. In this section, we present the theorem and proof for the second eigenvalue of this more general Google matrix.

Theorem 4.7.1. *If the spectrum of the stochastic matrix* $\mathbf{S}$ *is* $\{1, \lambda_2, \lambda_3, \ldots, \lambda_n\}$, *then the spectrum of the Google matrix* $\mathbf{G} = \alpha\mathbf{S} + (1-\alpha)\mathbf{e}\mathbf{v}^T$ *is* $\{1, \alpha\lambda_2, \alpha\lambda_3, \ldots, \alpha\lambda_n\}$, *where* $\mathbf{v}^T$ *is a probability vector.*

Proof. Since $\mathbf{S}$ is stochastic, $(1, \mathbf{e})$ is an eigenpair of $\mathbf{S}$. Let $\mathbf{Q} = (\,\mathbf{e} \quad \mathbf{X}\,)$ be a nonsingular matrix that has the eigenvector $\mathbf{e}$ as its first column. Let $\mathbf{Q}^{-1} = \begin{pmatrix} \mathbf{y}^T \\ \mathbf{Y}^T \end{pmatrix}$. Then $\mathbf{Q}^{-1}\mathbf{Q} = \begin{pmatrix} \mathbf{y}^T\mathbf{e} & \mathbf{y}^T\mathbf{X} \\ \mathbf{Y}^T\mathbf{e} & \mathbf{Y}^T\mathbf{X} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$, which gives two useful identities, $\mathbf{y}^T\mathbf{e} = 1$ and $\mathbf{Y}^T\mathbf{e} = \mathbf{0}$. As a result, the similarity transformation

$$\mathbf{Q}^{-1}\mathbf{S}\mathbf{Q} = \begin{pmatrix} \mathbf{y}^T\mathbf{e} & \mathbf{y}^T\mathbf{S}\mathbf{X} \\ \mathbf{Y}^T\mathbf{e} & \mathbf{Y}^T\mathbf{S}\mathbf{X} \end{pmatrix} = \begin{pmatrix} 1 & \mathbf{y}^T\mathbf{S}\mathbf{X} \\ 0 & \mathbf{Y}^T\mathbf{S}\mathbf{X} \end{pmatrix}$$

reveals that $\mathbf{Y}^T\mathbf{S}\mathbf{X}$ contains the remaining eigenvalues of $\mathbf{S}$, $\lambda_2, \ldots, \lambda_n$. Applying the similarity transformation to $\mathbf{G} = \alpha\mathbf{S} + (1-\alpha)\mathbf{e}\mathbf{v}^T$ gives

$$\begin{aligned}
\mathbf{Q}^{-1}(\alpha\mathbf{S} + (1-\alpha)\mathbf{e}\mathbf{v}^T)\mathbf{Q} &= \alpha\mathbf{Q}^{-1}\mathbf{S}\mathbf{Q} + (1-\alpha)\mathbf{Q}^{-1}\mathbf{e}\mathbf{v}^T\mathbf{Q} \\
&= \begin{pmatrix} \alpha & \alpha\mathbf{y}^T\mathbf{S}\mathbf{X} \\ 0 & \alpha\mathbf{Y}^T\mathbf{S}\mathbf{X} \end{pmatrix} + (1-\alpha)\begin{pmatrix} \mathbf{y}^T\mathbf{e} \\ \mathbf{Y}^T\mathbf{e} \end{pmatrix}(\,\mathbf{v}^T\mathbf{e} \quad \mathbf{v}^T\mathbf{X}\,) \\
&= \begin{pmatrix} \alpha & \alpha\mathbf{y}^T\mathbf{S}\mathbf{X} \\ 0 & \alpha\mathbf{Y}^T\mathbf{S}\mathbf{X} \end{pmatrix} + \begin{pmatrix} (1-\alpha) & (1-\alpha)\mathbf{v}^T\mathbf{X} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \\
&= \begin{pmatrix} 1 & \alpha\mathbf{y}^T\mathbf{S}\mathbf{X} + (1-\alpha)\mathbf{v}^T\mathbf{X} \\ \mathbf{0} & \alpha\mathbf{Y}^T\mathbf{S}\mathbf{X} \end{pmatrix}.
\end{aligned}$$

Therefore, the eigenvalues of $\mathbf{G} = \alpha\mathbf{S} + (1-\alpha)\mathbf{e}\mathbf{v}^T$ are $\{1, \alpha\lambda_2, \alpha\lambda_3, \ldots, \alpha\lambda_n\}$. ■

Chapter Fifteen

The Mathematics Guide

Appreciating the subtleties of PageRank, HITS, and other ranking schemes requires knowledge of some mathematical concepts. In particular, it's necessary to understand some aspects of linear algebra, discrete Markov chains, and graph theory. Rather than presenting a comprehensive survey of these areas, our purpose here is to touch on only the most relevant topics that arise in the mathematical analysis of Web search concepts. Technical proofs are generally omitted.

The common ground is linear algebra, so this is where we start. The reader that wants more detail or simply wants to review elementary linear algebra to an extent greater than that given here should consult [127].

15.1 LINEAR ALGEBRA

In the context of Web search the matrices encountered are almost always real, but because real matrices can generate complex numbers (e.g., eigenvalues) it's often necessary to consider complex numbers, vectors, and matrices. Throughout this chapter real numbers, real vectors, and real matrices are respectively denoted by $\Re$, $\Re^n$, and $\Re^{m\times n}$, while complex numbers, vectors, and matrices are respectively denoted by $\mathcal{C}$, $\mathcal{C}^n$, and $\mathcal{C}^{m\times n}$. The following basic concepts of arise in the mathematical analysis of Web search problems.

Norms

The most common way to measure the magnitude of a row (or column) vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ of real or complex numbers is by means of the euclidean norm (sometimes called the 2-norm) that is defined by

$$\|\mathbf{x}\|_2 = \sum_{i=1}^{n} |x_i|^2.$$

However, in the applications involving PageRank and Markov chains, it's more natural (and convenient) to use the vector 1-norm defined by

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|$$

because, for example, if $\mathbf{p}$ is a PageRank (or probability) vector, (i.e., a nonnegative vector with components summing to one) then $\|\mathbf{p}\|_1 = 1$. Occasionally the vector ∞-norm

$$\|\mathbf{x}\|_\infty = \max_i |x_i|$$

is used. All norms satisfy the three properties

$$\begin{aligned} \|\mathbf{x}\| &\geq 0 \quad \text{where} \quad \|\mathbf{x}\| = 0 \quad \text{if and only if} \quad \mathbf{x} = \mathbf{0}, \\ \|\alpha\mathbf{x}\| &= |\alpha| \; \|\mathbf{x}\| \quad \text{for all scalars } \alpha, \\ \|\mathbf{x} + \mathbf{y}\| &\leq \|\mathbf{x}\| + \|\mathbf{y}\| \quad \text{(the triangle inequality).} \end{aligned}$$

Associated with each vector norm is an *induced matrix norm.* If $\mathbf{A}$ is $m \times n$ and $\mathbf{x}$ is $n \times 1$, and if $\|{*}\|_\star$ is any vector norm, then the corresponding induced matrix norm is defined to be

$$\|\mathbf{A}\|_\star = \max_{\|\mathbf{x}\|_\star = 1} \|\mathbf{A}\mathbf{x}\|_\star .$$

The respective matrix norms that are induced by the 1- 2-, and ∞- vector norms are

$$\begin{aligned} \|\mathbf{A}\|_1 &= \max_j \sum_i |a_{ij}| = \text{the largest absolute column sum,} \\ \|\mathbf{A}\|_2 &= \sqrt{\lambda_{\max}}, \quad \text{where } \lambda_{\max} = \text{largest eigenvalue of } \mathbf{A}^T\mathbf{A}, \\ &\qquad \text{(replace transpose by conjugate transpose if } \mathbf{A} \text{ is complex),} \\ \|\mathbf{A}\|_\infty &= \max_i \sum_j |a_{ij}| = \text{the largest absolute row sum.} \end{aligned}$$

The details surrounding these properties can be found in [127].

The nice thing about induced matrix norms is that each of them is *compatible* with its corresponding vector norm in the sense that

$$\|\mathbf{A}\mathbf{x}\|_\star \leq \|\mathbf{A}\|_\star \, \|\mathbf{x}\|_\star .$$

However, this compatibility condition holds only for right-hand matrix-vector multiplication. For left-hand vector-matrix multiplication, which is common in Markov chain applications, transposition is needed to convert back to right-hand matrix-vector multiplication, and this results in different compatibility rules. If $\mathbf{x}^T$ is $1 \times n$ and $\mathbf{A}$ is $m \times n$, then

$$\|\mathbf{x}^T\mathbf{A}\|_1 \leq \|\mathbf{x}^T\|_1 \, \|\mathbf{A}\|_\infty, \|\mathbf{x}^T\mathbf{A}\|_\infty \leq \|\mathbf{x}^T\|_\infty \, \|\mathbf{A}\|_1.$$

Sensitivity of Linear Systems

It is assumed that the reader is familiar with Gaussian elimination methods for solving a system $\mathbf{A}_{m\times n}\mathbf{x}_{n\times 1} = \mathbf{b}_{m\times 1}$ of m linear equations in n unknowns. If not, read [127]. Algorithms for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ are important, but the general behavior of a solution to small uncertainties or perturbations in the coefficients is particularly relevant, especially in light of the fact that the PageRank vector is the solution to a particular linear system.

While greater generality is possible, it suffices to consider a square nonsingular system $\mathbf{A}\mathbf{x} = \mathbf{b}$ in which both $\mathbf{A}$ and $\mathbf{b}$ are subject to uncertainties that might be the result of modeling error, numerical round-off error, measurement error, or small perturbations of any kind. How much uncertainty (or sensitivity) can the solution $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ exhibit?

An answer is provided by using calculus. Consider the entries of $\mathbf{A} = \mathbf{A}(t)$ and $\mathbf{b} = \mathbf{b}(t)$ to vary with a differentiable parameter t, and compute the relative size of the derivative of $\mathbf{x} = \mathbf{x}(t)$ by differentiating $\mathbf{b} = \mathbf{Ax}$ to obtain $\mathbf{b}' = (\mathbf{Ax})' = \mathbf{A}'\mathbf{x} + \mathbf{Ax}'$ (with $\star'$ denoting $d\star/dt$). Taking norms (the choice of norm is not important) yields

$$\begin{aligned}\|\mathbf{x}'\| &= \left\|\mathbf{A}^{-1}\mathbf{b}' - \mathbf{A}^{-1}\mathbf{A}'\mathbf{x}\right\| \leq \left\|\mathbf{A}^{-1}\mathbf{b}'\right\| + \left\|\mathbf{A}^{-1}\mathbf{A}'\mathbf{x}\right\| \\ &\leq \left\|\mathbf{A}^{-1}\right\| \|\mathbf{b}'\| + \left\|\mathbf{A}^{-1}\right\| \|\mathbf{A}'\| \|\mathbf{x}\|.\end{aligned}$$

Consequently,

$$\begin{aligned}\frac{\|\mathbf{x}'\|}{\|\mathbf{x}\|} &\leq \frac{\left\|\mathbf{A}^{-1}\right\| \|\mathbf{b}'\|}{\|\mathbf{x}\|} + \left\|\mathbf{A}^{-1}\right\| \|\mathbf{A}'\| \\ &\leq \|\mathbf{A}\| \left\|\mathbf{A}^{-1}\right\| \frac{\|\mathbf{b}'\|}{\|\mathbf{A}\| \|\mathbf{x}\|} + \|\mathbf{A}\| \left\|\mathbf{A}^{-1}\right\| \frac{\|\mathbf{A}'\|}{\|\mathbf{A}\|} \\ &\leq \kappa\frac{\|\mathbf{b}'\|}{\|\mathbf{b}\|} + \kappa\frac{\|\mathbf{A}'\|}{\|\mathbf{A}\|} = \kappa\left(\frac{\|\mathbf{b}'\|}{\|\mathbf{b}\|} + \frac{\|\mathbf{A}'\|}{\|\mathbf{A}\|}\right),\end{aligned}$$

where $\kappa = \|\mathbf{A}\| \left\|\mathbf{A}^{-1}\right\|$. The terms $\|\mathbf{x}'\| / \|\mathbf{x}\|$, $\|\mathbf{b}'\| / \|\mathbf{b}\|$ and $\|\mathbf{A}'\| / \|\mathbf{A}\|$ represent the respective relative sensitivities of $\mathbf{x}$, $\mathbf{b}$, and $\mathbf{A}$ to small changes. Because κ represents a magnification of the sum of the relative sensitivities in $\mathbf{b}$, and $\mathbf{A}$, κ is called a *condition number* for $\mathbf{A}$. The situation can summarize the situation as follows.

Sensitivity of Linear Systems

For a nonsingular system $\mathbf{Ax} = \mathbf{b}$, the relative sensitivity of $\mathbf{x}$ to uncertainties or perturbations in $\mathbf{A}$ and $\mathbf{b}$ is never more than the sum of the relative changes in $\mathbf{A}$ and $\mathbf{b}$ magnified by the condition number $\kappa = \|\mathbf{A}\| \left\|\mathbf{A}^{-1}\right\|$.

A Practical Rule of Thumb. If Gaussian elimination with partial pivoting is used to solve a well-scaled (row norms in $\mathbf{A}$ are approximately one) nonsingular system $\mathbf{Ax} = \mathbf{b}$ using t-digit floating-point arithmetic, and if κ is of order 10^p, then, assuming no other source of error exists, the computed solution can be expected to be accurate to at least $t - p$ significant digits, more or less. In other words, one expects to lose roughly p significant figures. This doesn't preclude the possibility of getting lucky and attaining a higher degree of accuracy—it just says that you shouldn't bet the farm on it.

Rank-One Updates

Suppose that $\mathbf{A} \in \Re^{n\times n}$ is the coefficient matrix of a nonsingular system $\mathbf{Ax} = \mathbf{b}$ that contains information that periodically requires updating, and each time new information is received, the system must be re-solved. Rather than starting from scratch each time, it makes sense to try to perturb the solution from the previous period in a simple but predictable way. Theoretically, the solution is always $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$, so the problem of updating the solution to a linear system is equivalent to the problem of updating the inverse matrix $\mathbf{A}^{-1}$. If the new information can be formatted as a rank-one matrix $\mathbf{cd}^T$, where $\mathbf{c}, \mathbf{d} \in \Re^{n\times 1}$, then there is a formula for updating $\mathbf{A}^{-1}$.

Sherman–Morrison Rank-One Updating Formula

If $\mathbf{A}_{n\times n}$ is nonsingular and if $\mathbf{c}$ and $\mathbf{d}$ are columns such that $1+\mathbf{d}^T\mathbf{A}^{-1}\mathbf{c}\neq 0$, then the sum $\mathbf{A}+\mathbf{c}\mathbf{d}^T$ is nonsingular, and

$$\left(\mathbf{A}+\mathbf{c}\mathbf{d}^T\right)^{-1}=\mathbf{A}^{-1}-\frac{\mathbf{A}^{-1}\mathbf{c}\mathbf{d}^T\mathbf{A}^{-1}}{1+\mathbf{d}^T\mathbf{A}^{-1}\mathbf{c}}. \tag{15.1.1}$$

The Sherman–Morrison formula makes it clear that when a nonsingular system $\mathbf{A}\mathbf{x}=\mathbf{b}$ is updated to produce another nonsingular system $(\mathbf{A}+\mathbf{c}\mathbf{d}^T)\mathbf{z}=\mathbf{b}$, where $\mathbf{b},\mathbf{c},\mathbf{d}\in\Re^{n\times 1}$, the solution of the updated system is

$$\begin{aligned}\mathbf{z}=(\mathbf{A}+\mathbf{c}\mathbf{d}^T)^{-1}\mathbf{b}&=\left(\mathbf{A}^{-1}-\frac{\mathbf{A}^{-1}\mathbf{c}\mathbf{d}^T\mathbf{A}^{-1}}{1+\mathbf{d}^T\mathbf{A}^{-1}\mathbf{c}}\right)\mathbf{b}\\&=\mathbf{A}^{-1}\mathbf{b}-\frac{\mathbf{A}^{-1}\mathbf{c}\mathbf{d}^T\mathbf{A}^{-1}\mathbf{b}}{1+\mathbf{d}^T\mathbf{A}^{-1}\mathbf{c}}=\mathbf{x}-\frac{\mathbf{A}^{-1}\mathbf{c}\mathbf{d}^T\mathbf{x}}{1+\mathbf{d}^T\mathbf{A}^{-1}\mathbf{c}}.\end{aligned}$$

The Sherman–Morrison formula is particularly useful when an update involves only one row or column of $\mathbf{A}$. For example, suppose that the only the i^{th} row of $\mathbf{A}$ is affected—say row $\mathbf{A}_{i*}$ is updated to become $\mathbf{B}_{i*}$, and let $\boldsymbol{\epsilon}_i^T=\mathbf{B}_{i*}-\mathbf{A}_{i*}$. If $\mathbf{e}_i$ denotes the i^{th} unit column (the i^{th} column of the identity matrix $\mathbf{I}$), then the updated matrix can be written as

$$\mathbf{B}=\mathbf{A}+\mathbf{e}_i\boldsymbol{\epsilon}_i^T,$$

so that $\mathbf{e}_i$ plays the role of $\mathbf{c}$ in (15.1.1), and $\mathbf{A}^{-1}\mathbf{c}=\mathbf{A}^{-1}\mathbf{e}_i=[\mathbf{A}^{-1}]_{*i}$, the i^{th} column of $\mathbf{A}^{-1}$. Consequently, $\mathbf{B}^{-1}$ can be constructed directly from the entries in $\mathbf{A}^{-1}$ and the perturbation vector $\boldsymbol{\epsilon}^T$ by writing.

$$\mathbf{B}^{-1}=\left(\mathbf{A}+\mathbf{e}_i\boldsymbol{\epsilon}_i^T\right)^{-1}=\mathbf{A}^{-1}-\frac{[\mathbf{A}^{-1}]_{*i}\boldsymbol{\epsilon}_i^T\mathbf{A}^{-1}}{1+\boldsymbol{\epsilon}_i^T[\mathbf{A}^{-1}]_{*i}}.$$

Eigenvalues and Eigenvectors

For a matrix $\mathbf{A}\in\mathcal{C}^{n\times n}$, the scalars λ and the vectors $\mathbf{x}_{n\times 1}\neq\mathbf{0}$ satisfying $\mathbf{A}\mathbf{x}=\lambda\mathbf{x}$ are the respective *eigenvalues* and *eigenvectors* for $\mathbf{A}$. A row vector $\mathbf{y}^T$ is a *left-hand eigenvector* if $\mathbf{y}^T\mathbf{A}=\lambda\mathbf{y}^T$.

The set $\sigma(\mathbf{A})$ of *distinct* eigenvalues is called the *spectrum* of $\mathbf{A}$, and the *spectral radius* of $\mathbf{A}$ is the nonnegative number

$$\rho(\mathbf{A})=\max_{\lambda\in\sigma(\mathbf{A})}|\lambda|.$$

The circle in the complex plane that is centered at the origin and has radius $\rho(\mathbf{A})$ is called the *spectral circle* , and it is a straightforward exercise to verify that

$$\rho(\mathbf{A})\leq\|\mathbf{A}\| \tag{15.1.2}$$

for all matrix norms.

The eigenvalues of $\mathbf{A}_{n\times n}$ are the roots of the characteristic polynomial $p(\lambda)=\det(\mathbf{A}-\lambda\mathbf{I})$, where $\det(\star)$ denotes determinant. The degree of $p(\lambda)$ is n, so, altogether,

$\mathbf{A}$ has n eigenvalues, but some may be complex numbers (even if the entries of $\mathbf{A}$ are real numbers), and some eigenvalues may be repeated. If $\mathbf{A}$ contains only real numbers, then its complex eigenvalues must occur in conjugate pairs—i.e., if $\lambda \in \sigma(\mathbf{A})$, then $\overline{\lambda} \in \sigma(\mathbf{A})$.

The *algebraic multiplicity* of an eigenvalue λ of $\mathbf{A}$ is the number of times that λ is repeated as a root of the characteristic equation. If *alg mult*$_{\mathbf{A}}(\lambda) = 1$, then λ is said to be a *simple eigenvalue*.

The *geometric multiplicity* of an eigenvalue λ of $\mathbf{A}$ is the number of linearly independent eigenvectors that are associated with λ. In more formal terms, *geo mult*$_{\mathbf{A}}(\lambda) = \dim N(\mathbf{A} - \lambda\mathbf{I})$, where $N(\star)$ denotes the nullspace or kernel of a matrix. It is always the case that *geo mult*$_{\mathbf{A}}(\lambda) \leq$ *alg mult*$_{\mathbf{A}}(\lambda)$. If *geo mult*$_{\mathbf{A}}(\lambda) =$ *alg mult*$_{\mathbf{A}}(\lambda)$, then λ is said to be a *semisimple eigenvalue*.

The *index of an eigenvalue* $\lambda \in \sigma(\mathbf{A})$ is defined to be the smallest positive integer k such that *rank*$\big((\mathbf{A} - \lambda\mathbf{I})^k\big) =$ *rank*$\big((\mathbf{A} - \lambda\mathbf{I})^{k+1}\big)$. It is understood that *index*$(\lambda) = 0$ when $\lambda \notin \sigma(\mathbf{A})$.

There are several different ways to characterize index. For $\lambda \in \sigma(\mathbf{A}_{n\times n})$, saying that $k =$ *index*(λ) is equivalent to saying that k is the smallest positive integer such that any of the following statements hold.

- $R\big((\mathbf{A} - \lambda\mathbf{I})^k\big) = R\big((\mathbf{A} - \lambda\mathbf{I})^{k+1}\big)$, where $R(\star)$ denotes range.
- $N\big((\mathbf{A} - \lambda\mathbf{I})^k\big) = N\big((\mathbf{A} - \lambda\mathbf{I})^{k+1}\big)$, where $N(\star)$ denotes nullspace (or kernel).
- $R\big((\mathbf{A} - \lambda\mathbf{I})^k\big) \cap N\big((\mathbf{A} - \lambda\mathbf{I})^k\big) = \mathbf{0}$.
- $\mathcal{C}^n = R\big((\mathbf{A} - \lambda\mathbf{I})^k\big) \oplus N\big((\mathbf{A} - \lambda\mathbf{I})^k\big)$, where $\oplus$ denotes direct sum.

The Jordan Form

Eigenvalues and eigenvectors are for matrices what DNA is for biological entities, and the Jordan form for a square matrix $\mathbf{A}$ completely characterizes the eigenstructure of $\mathbf{A}$. The theoretical basis for why the Jordan form looks as it does is somewhat involved, but the "form" itself is easy to understand, and that's all you need to deal with the issues that arise in understanding Web searching concepts.

Given a matrix $\mathbf{A}_{n\times n}$, a *Jordan block* associated with an eigenvalue $\lambda \in \sigma(\mathbf{A})$ is defined to be a matrix of the form

$$\mathbf{J}_\star(\lambda) = \begin{pmatrix} \lambda & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda \end{pmatrix}. \tag{15.1.3}$$

A *Jordan segment* $\mathbf{J}(\lambda)$ associated with $\lambda \in \sigma(\mathbf{A})$ is defined to be a block-diagonal matrix containing one or more Jordan blocks. In other words, a Jordan segment looks like

$$\mathbf{J}(\lambda) = \begin{pmatrix} \mathbf{J}_1(\lambda) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_2(\lambda) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{J}_t(\lambda) \end{pmatrix} \quad \text{with each } \mathbf{J}_\star(\lambda) \text{ being a Jordan block.}$$

The *Jordan canonical form* (or simply the *Jordan form*) for $\mathbf{A}$ is a block-diagonal matrix composed of the Jordan segments for each distinct eigenvalue. In other words, if $\sigma(\mathbf{A}) = \{\lambda_1, \lambda_2, \ldots, \lambda_s\}$, then the Jordan form for $\mathbf{A}$ is

$$\mathbf{J} = \begin{pmatrix} \mathbf{J}(\lambda_1) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{J}(\lambda_2) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{J}(\lambda_s) \end{pmatrix}. \tag{15.1.4}$$

There is only one Jordan segment for each eigenvalue, but each segment can contain several Jordan blocks of varying size. The formula that governs the sizes and numbers of Jordan blocks is given in the following complete statement concerning the Jordan form.

Jordan's Theorem

For every $\mathbf{A} \in \mathcal{C}^{n\times n}$ there is a nonsingular matrix $\mathbf{P}$ such that

$$\mathbf{P}^{-1}\mathbf{A}\mathbf{P} = \mathbf{J} \tag{15.1.5}$$

is the Jordan form (15.1.4) that is characterized by the following features.

- $\mathbf{J}$ contains one Jordan segment $\mathbf{J}(\lambda)$ for each distinct eigenvalue $\lambda \in \sigma(\mathbf{A})$.
- Each segment $\mathbf{J}(\lambda)$ contains $t = \dim N(\mathbf{A} - \lambda\mathbf{I})$ Jordan blocks.
- The number of $i \times i$ Jordan blocks in $\mathbf{J}(\lambda)$ is given by

  $$\nu_i(\lambda) = r_{i-1}(\lambda) - 2r_i(\lambda) + r_{i+1}(\lambda), \quad \text{where} \quad r_i(\lambda) = \textit{rank}\left((\mathbf{A} - \lambda\mathbf{I})^i\right).$$

- The largest Jordan block in each segment $\mathbf{J}(\lambda)$ is $k \times k$, where $k = \textit{index}\,(\lambda)$.

The structure of $\mathbf{J}$ is unique in the sense that the number and sizes of the Jordan blocks in each segment is uniquely determined by the entries in $\mathbf{A}$. Two $n \times n$ matrices $\mathbf{A}$ and $\mathbf{B}$ are *similar* (i.e., $\mathbf{B} = \mathbf{Q}^{-1}\mathbf{A}\mathbf{Q}$ for some nonsingular $\mathbf{Q}$) if and only if $\mathbf{A}$ and $\mathbf{B}$ have the same Jordan form.

The matrix $\mathbf{P}$ in (15.1.5) is not unique, but its columns always form *Jordan chains* (or *generalized eigenvectors*) in the following sense. For each Jordan block $\mathbf{J}_\star(\lambda)$, there is a set of columns $\mathbf{P}_\star$ of corresponding size and position in $\mathbf{P} = \left[\cdots \mid \mathbf{P}_\star \mid \cdots\right]$ such that

$$\mathbf{P}_\star = \left[\, (\mathbf{A} - \lambda\mathbf{I})^i \mathbf{x}_\star \,\middle|\, (\mathbf{A} - \lambda\mathbf{I})^{i-1} \mathbf{x}_\star \,\middle|\, \cdots \,\middle|\, (\mathbf{A} - \lambda\mathbf{I})\,\mathbf{x}_\star \,\middle|\, \mathbf{x}_\star \right]_{(i+1)\times n}$$

for some i and some $\mathbf{x}_\star$, where $(\mathbf{A} - \lambda\mathbf{I})^i \mathbf{x}_\star$ is a particular eigenvector associated with λ. Formulas exist for determining i and $\mathbf{x}_\star$ [127, p. 594], but the computations can be complicated. Fortunately, we rarely need to compute $\mathbf{P}$.

An important corollary of Jordan's theorem (15.1.5) is the following statement concerning the diagonalizability of a square matrix.

Diagonalizability

Each of the following statements is equivalent to saying that $\mathbf{A} \in \mathcal{C}^{n\times n}$ is similar to a diagonal matrix—i.e., $\mathbf{J}$ is diagonal (all Jordan blocks are 1×1).

- *index* $(\lambda) = 1$ for each $\lambda \in \sigma(\mathbf{A})$ (i.e., every eigenvalue is semisimple).
- *alg mult*$_{\mathbf{A}}(\lambda) =$ *geo mult*$_{\mathbf{A}}(\lambda)$ for each $\lambda \in \sigma(\mathbf{A})$.
- $\mathbf{A}$ has a complete set of n linearly independent eigenvectors (i.e., each column of $\mathbf{P}$ is an eigenvector for $\mathbf{A}$).

Functions of a Matrix

An important use of the Jordan form is to define functions of $\mathbf{A} \in \mathcal{C}^{n\times n}$. That is, given a function $f : \mathcal{C} \to \mathcal{C}$, what should $f(\mathbf{A})$ mean? The answer is straightforward. Suppose that $\mathbf{A} = \mathbf{PJP}^{-1}$, where $\mathbf{J} = \begin{pmatrix} \ddots & & \\ & \mathbf{J}_\star & \\ & & \ddots \end{pmatrix}$ is in Jordan form with the $\mathbf{J}_\star$'s representing the Jordan blocks described in (15.1.3) It's natural to define the value of f at $\mathbf{A}$ to be

$$f(\mathbf{A}) = \mathbf{P}f(\mathbf{J})\mathbf{P}^{-1} = \mathbf{P}\begin{pmatrix} \ddots & & \\ & f(\mathbf{J}_\star) & \\ & & \ddots \end{pmatrix}\mathbf{P}^{-1}, \tag{15.1.6}$$

but the trick is correctly defining $f(\mathbf{J}_\star)$. It turns out that right way to do this is by setting

$$f(\mathbf{J}_\star) = f\begin{pmatrix} \lambda & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda \end{pmatrix} = \begin{pmatrix} f(\lambda) & f'(\lambda) & \frac{f''(\lambda)}{2!} & \cdots & \frac{f^{(k-1)}(\lambda)}{(k-1)!} \\ & f(\lambda) & f'(\lambda) & \ddots & \vdots \\ & & \ddots & \ddots & \frac{f''(\lambda)}{2!} \\ & & & f(\lambda) & f'(\lambda) \\ & & & & f(\lambda) \end{pmatrix}. \tag{15.1.7}$$

Matrix Functions

Let $\mathbf{A} \in \mathcal{C}^{n\times n}$ with $\sigma(\mathbf{A}) = \{\lambda_1, \lambda_2, \ldots, \lambda_s\}$, and let $f : \mathcal{C} \to \mathcal{C}$ be such that $f(\lambda_i),\ f'(\lambda_i),\ \ldots,\ f^{(k_i-1)}(\lambda_i)$ exist for each i, where $k_i =$ *index* (λ_i). Define

$$f(\mathbf{A}) = \mathbf{P}f(\mathbf{J})\mathbf{P}^{-1} = \mathbf{P}\begin{pmatrix} \ddots & & \\ & f(\mathbf{J}_\star) & \\ & & \ddots \end{pmatrix}\mathbf{P}^{-1}, \tag{15.1.8}$$

where $\mathbf{J}$ is the Jordan form for $\mathbf{A}$ and $f(\mathbf{J}_\star)$ is given by (15.1.7).

There are at least two other equivalent and useful ways to view functions of matrices. The first of these is called the *spectral theorem for matrix functions,* and this arises by expanding the product on the right-hand side of (15.1.8) expand to yield the following.

Spectral Theorem for General Matrices

If $\mathbf{A} \in \mathcal{C}^{n\times n}$ with $\sigma(\mathbf{A}) = \{\lambda_1, \lambda_2, \ldots, \lambda_s\}$, then

$$f(\mathbf{A}) = \sum_{i=1}^{s}\sum_{j=0}^{k_i-1} \frac{f^{(j)}(\lambda_i)}{j!}(\mathbf{A}-\lambda_i\mathbf{I})^j\mathbf{G}_i, \tag{15.1.9}$$

where each $\mathbf{G}_i$ has the following properties.

- $\mathbf{G}_i$ is a projector (i.e., $\mathbf{G}_i^2 = \mathbf{G}_i$) onto $N\big((\mathbf{A}-\lambda_i\mathbf{I})^{k_i}\big)$ along $R\big((\mathbf{A}-\lambda_i\mathbf{I})^{k_i}\big)$.
- $\mathbf{G}_1 + \mathbf{G}_2 + \cdots + \mathbf{G}_s = \mathbf{I}$.
- $\mathbf{G}_i\mathbf{G}_j = \mathbf{0}$ when $i \neq j$.
- $(\mathbf{A}-\lambda_i\mathbf{I})\mathbf{G}_i = \mathbf{G}_i(\mathbf{A}-\lambda_i\mathbf{I})$ is nilpotent of index k_i.

The $\mathbf{G}_i$'s are called the *spectral projectors* associated with matrix $\mathbf{A}$.

Another useful way to deal with functions of a matrix is by means of infinite series.

Infinite Series Representations

If $\sum_{j=0}^{\infty} c_j(z-z_0)^j$ converges to $f(z)$ at each point in a circle $|z-z_0| = r$, and if $|\lambda - z_0| < r$ for each eigenvalue $\lambda \in \sigma(\mathbf{A})$, then $\sum_{j=0}^{\infty} c_j(\mathbf{A}-z_0\mathbf{I})^j$ converges, and

$$f(\mathbf{A}) = \sum_{j=0}^{\infty} c_j(\mathbf{A}-z_0\mathbf{I})^j.$$

If $\mathbf{A}$ is diagonalizable—i.e., if is similar to a diagonal matrix

$$\mathbf{A} = \mathbf{P}\begin{pmatrix} \lambda_1\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \lambda_2\mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \lambda_s\mathbf{I} \end{pmatrix}\mathbf{P}^{-1},$$

then

$$f(\mathbf{A}) = \mathbf{P}\begin{pmatrix} f(\lambda_1)\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & f(\lambda_2)\mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & f(\lambda_s)\mathbf{I} \end{pmatrix}\mathbf{P}^{-1},$$

and formula (15.1.9) yields the following spectral theorem for diagonalizable matrices

Spectral Theorem for Diagonalizable Matrices

If $\mathbf{A}$ is diagonalizable with with $\sigma(\mathbf{A}) = \{\lambda_1, \lambda_2, \ldots, \lambda_s\}$, then

$$\mathbf{A} = \lambda_1 \mathbf{G}_1 + \lambda_2 \mathbf{G}_2 + \cdots + \lambda_s \mathbf{G}_s, \tag{15.1.10}$$

and

$$f(\mathbf{A}) = f(\lambda_1)\mathbf{G}_1 + f(\lambda_2)\mathbf{G}_2 + \cdots + f(\lambda_s)\mathbf{G}_s, \tag{15.1.11}$$

where the spectral projectors $\mathbf{G}_i$ have the following properties.

- $\mathbf{G}_i = \mathbf{G}_i^2$ is the projector onto the eigenspace $N(\mathbf{A} - \lambda_i \mathbf{I})$ along $R(\mathbf{A} - \lambda_i \mathbf{I})$,
- $\mathbf{G}_1 + \mathbf{G}_2 + \cdots + \mathbf{G}_s = \mathbf{I}$,
- $\mathbf{G}_i \mathbf{G}_j = \mathbf{0}$ when $i \neq j$,
- $\mathbf{G}_i = \prod_{\substack{j=1 \\ j \neq i}}^{k} (\mathbf{A} - \lambda_j \mathbf{I}) \Big/ \prod_{\substack{j=1 \\ j \neq i}}^{k} (\lambda_i - \lambda_j)$ for $i = 1, 2, \ldots, k$.
- If λ_i happens to be a simple eigenvalue, then

$$\mathbf{G}_i = \mathbf{x}\mathbf{y}^* / \mathbf{y}^* \mathbf{x} \tag{15.1.12}$$

in which $\mathbf{x}$ and $\mathbf{y}^*$ are respective right-hand and left-hand eigenvectors associated with λ_i.

Powers of Matrices and Convergence

A fundamental issue in analyzing PageRank concerns convergence of powers of matrices. It follows from (15.1.8) that each power of $\mathbf{A} \in \mathcal{C}^{n \times n}$ is given by

$$\mathbf{A}^k = \mathbf{P}\mathbf{J}^k\mathbf{P}^{-1} = \mathbf{P}\begin{pmatrix} \ddots & & \\ & \mathbf{J}_\star^k & \\ & & \ddots \end{pmatrix}\mathbf{P}^{-1}, \quad \text{where} \quad \mathbf{J}_\star = \begin{pmatrix} \lambda & 1 & \\ & \ddots & \ddots \\ & & \lambda \end{pmatrix},$$

and

$$\mathbf{J}_\star^k = \begin{pmatrix} \lambda^k & \binom{k}{1}\lambda^{k-1} & \binom{k}{2}\lambda^{k-2} & \cdots & \binom{k}{m-1}\lambda^{k-m+1} \\ & \lambda^k & \binom{k}{1}\lambda^{k-1} & \ddots & \vdots \\ & & \ddots & \ddots & \binom{k}{2}\lambda^{k-2} \\ & & & \lambda^k & \binom{k}{1}\lambda^{k-1} \\ & & & & \lambda^k \end{pmatrix}_{m \times m}. \tag{15.1.13}$$

This observation leads to the following limiting properties.

Convergence to Zero and The Neumann Series

For $\mathbf{A} \in \mathcal{C}^{n\times n}$, the following statements are equivalent.

- $\rho(\mathbf{A}) < 1.$ (15.1.14)
- $\lim_{k\to\infty} \mathbf{A}^k = \mathbf{0}.$ (15.1.15)
- The *Neumann series* series $\sum_{k=0}^{\infty} \mathbf{A}^k$ converges to $(\mathbf{I} - \mathbf{A})^{-1}$. (15.1.16)

It may be the case that the powers $\mathbf{A}^k$ converge, but not to the zero matrix. The complete story concerning $\lim_{k\to\infty} \mathbf{A}^k$ is as follows.

Limits of Powers

For $\mathbf{A} \in \mathcal{C}^{n\times n}$, $\lim_{k\to\infty} \mathbf{A}^k$ exists if and only if $\rho(\mathbf{A}) < 1$, in which case $\lim_{k\to\infty} \mathbf{A}^k = \mathbf{0}$, or else $\rho(\mathbf{A}) = 1$, with $\lambda = 1$ being semisimple and the only eigenvalue on the unit circle. When it exists,

$$\lim_{k\to\infty} \mathbf{A}^k = \mathbf{G} = \text{the projector onto } N(\mathbf{I} - \mathbf{A}) \text{ along } R(\mathbf{I} - \mathbf{A}). \qquad (15.1.17)$$

Averages and Summability

With each scalar sequence $\{\alpha_1, \alpha_2, \alpha_3, \ldots\}$ there is an associated sequence of averages $\{\mu_1, \mu_2, \mu_3, \ldots\}$ in which

$$\mu_1 = \alpha_1, \quad \mu_2 = \frac{\alpha_1 + \alpha_2}{2}, \quad \ldots, \quad \mu_n = \frac{\alpha_1 + \alpha_2 + \cdots + \alpha_n}{n}.$$

This sequence of averages is called the *Cesàro sequence*, and when $\lim_{n\to\infty} \mu_n = \alpha$, we say that $\{\alpha_n\}$ is *Cesàro summable* (or merely *summable*) to α. It can be proven that if $\{\alpha_n\}$ converges to α, then $\{\mu_n\}$ converges to α, but not conversely. In other words, convergence implies summability, but summability doesn't insure convergence. To see that a sequence can be summable without being convergent, notice that the oscillatory sequence $\{0, 1, 0, 1, \ldots\}$ doesn't converge, but it is summable to $1/2$, the mean value of $\{0, 1\}$. Averaging has a smoothing effect, so oscillations that prohibit convergence of the original sequence tend to be smoothed away or averaged out in the Cesàro sequence.

Similar statements hold for sequences of vectors and matrices, but Cesàro summability is particularly interesting when it is applied to the sequence $\mathcal{P} = \{\mathbf{A}^k\}_{k=0}^{\infty}$ of powers of a square matrix $\mathbf{A}$.

Summability

$\mathbf{A} \in \mathcal{C}^{n \times n}$ is Cesàro summable if and only if $\rho(\mathbf{A}) < 1$ or else $\rho(\mathbf{A}) = 1$ with each eigenvalue on the unit circle being semisimple. When it exists, the limit

$$\lim_{k \to \infty} \frac{\mathbf{I} + \mathbf{A} + \cdots + \mathbf{A}^{k-1}}{k} = \mathbf{G} \tag{15.1.18}$$

is the projector onto $N(\mathbf{I} - \mathbf{A})$ along $R(\mathbf{I} - \mathbf{A})$.

Notice that $\mathbf{G} \neq \mathbf{0}$ if and only if $1 \in \sigma(\mathbf{A})$, in which case $\mathbf{G}$ is the spectral projector associated with $\lambda = 1$. Furthermore, if $\lim_{k\to\infty} \mathbf{A}^k = \mathbf{G}$, then $\mathbf{A}$ is summable to $\mathbf{G}$, but not conversely.

The Power Method

Google's original method of choice for computing the PageRank vector was the *power method*, which is an iterative technique for computing a dominant eigenpair $(\lambda_1, \mathbf{x})$ of a diagonalizable matrix $\mathbf{A} \in \Re^{m \times m}$ with eigenvalues

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_k|.$$

For the Google matrix, the dominant eigenvalue is $\lambda_1 = 1$, but since the analysis of the power method is not dependent on this fact, we will allow λ_1 to be more general. However, notice that the hypothesis $|\lambda_1| > |\lambda_2|$ implies λ_1 is real—otherwise $\overline{\lambda_1}$ (the complex conjugate) is another eigenvalue with the same magnitude as λ_1. Consider the function $f(z) = (z/\lambda_1)^n$, and use the spectral representation (15.1.11) along with $|\lambda_i/\lambda_1| < 1$ for $i = 2, 3, \ldots, k$ to conclude that

$$\begin{aligned}\left(\frac{\mathbf{A}}{\lambda_1}\right)^n &= f(\mathbf{A}) = f(\lambda_1)\mathbf{G}_1 + f(\lambda_2)\mathbf{G}_2 + \cdots + f(\lambda_k)\mathbf{G}_k \\ &= \mathbf{G}_1 + \left(\frac{\lambda_2}{\lambda_1}\right)^n \mathbf{G}_2 + \cdots + \left(\frac{\lambda_k}{\lambda_1}\right)^n \mathbf{G}_k \to \mathbf{G}_1 \text{ as } n \to \infty. \quad (15.1.19)\end{aligned}$$

For every $\mathbf{x}_0$ we have $(\mathbf{A}^n\mathbf{x}_0/\lambda_1^n) \to \mathbf{G}_1\mathbf{x}_0 \in N(\mathbf{A} - \lambda_1\mathbf{I})$, so, if $\mathbf{G}_1\mathbf{x}_0 \neq \mathbf{0}$, then $\mathbf{A}^n\mathbf{x}_0/\lambda_1^n$ converges to an eigenvector associated with λ_1. This means that the direction of $\mathbf{A}^n\mathbf{x}_0$ tends toward the direction of an eigenvector because λ_1^n acts only as a scaling factor to keep the length of $\mathbf{A}^n\mathbf{x}_0$ under control. Rather than using λ_1^n, we can scale $\mathbf{A}^n\mathbf{x}_0$ with something more convenient. For example, $\|\mathbf{A}^n\mathbf{x}_0\|$ (for any vector norm) is a reasonable scaling factor, but there are better choices. For vectors $\mathbf{v}$, let $m(\mathbf{v})$ denote the component of maximal magnitude, and if there is more than one maximal component, let $m(\mathbf{v})$ be the *first* maximal component—e.g., $m(1, 3, -2) = 3$, and $m(-3, 3, -2) = -3$. The power method can be summarized as follows.

Power Method

Start with an arbitrary guess $\mathbf{x}_0$. (Actually it can't be completely arbitrary because you need $\mathbf{x}_0 \notin R(\mathbf{A} - \lambda_1\mathbf{I})$ to ensure $\mathbf{G}_1\mathbf{x}_0 \neq \mathbf{0}$, but it's highly unlikely that randomly chosen vector $\mathbf{x}_0$ will satisfy $\mathbf{G}_1\mathbf{x}_0 = \mathbf{0}$.) It can be shown [127, p. 534] that if we set

$$\mathbf{y}_n = \mathbf{A}\mathbf{x}_n, \quad \nu_n = m(\mathbf{y}_n), \quad \mathbf{x}_{n+1} = \frac{\mathbf{y}_n}{\nu_n}, \quad \text{for } n = 0, 1, 2, \ldots, \qquad (15.1.20)$$

then $\mathbf{x}_n \to \mathbf{x}$ and $\nu_n \to \lambda_1$, where $\mathbf{A}\mathbf{x} = \lambda_1\mathbf{x}$.

There are several reasons why the power method might be attractive for computing Google's PageRank vector.

- Each iteration requires only one matrix-vector product, and this can be exploited to reduce the computational effort when $\mathbf{A}$ is large and sparse (mostly zeros), as is the case in Google's application.
- Computations can be done in parallel by simultaneously computing inner products of rows of $\mathbf{A}$ with $\mathbf{x}_n$.
- It's clear from (15.1.19) that, for a diagonalizable matrix, the rate at which (15.1.20) converges depends on how fast $(\lambda_2/\lambda_1)^n \to 0$. As discussed in section 4.7, Google can regulate $|\lambda_2|$ through the choice of the Google parameter α, so they can control the rate of convergence (it's just assumed that Google's matrix is diagonalizable).
- Since $\lambda_1 = 1$ for Google's PageRank problem, there is no need for the scaling factor ν_n. In other words, the iterations are simply $\mathbf{x}_{n+1} = \mathbf{A}\mathbf{x}_n$.

Linear Stationary Iterations

Solving systems of linear equations $\mathbf{A}_{n\times n}\mathbf{x} = \mathbf{b}$ is a frequent necessity for Web search applications, but the magnitude of n is usually too large for direct solution methods based on Gaussian elimination to be effective. Consequently, iterative techniques are often the only choice, and, because of size, sparsity, and memory considerations, the preferred algorithms are the simpler methods based on matrix-vector products that require no additional storage beyond that of the original data. Linear stationary iterative methods are the most common.

Linear Stationary Iterations

Let $\mathbf{Ax} = \mathbf{b}$ be a linear system that is square but otherwise arbitrary. Writing $\mathbf{A}$ as $\mathbf{A} = \mathbf{M} - \mathbf{N}$ in which $\mathbf{M}^{-1}$ exists is called a *splitting* of $\mathbf{A}$, and the product $\mathbf{H} = \mathbf{M}^{-1}\mathbf{N}$ is called the associated *iteration matrix.* For $\mathbf{d} = \mathbf{M}^{-1}\mathbf{b}$ and for an initial vector $\mathbf{x}(0)$, the sequence defined by

$$\mathbf{x}(k) = \mathbf{Hx}(k-1) + \mathbf{d} \qquad k = 1, 2, 3, \ldots \tag{15.1.21}$$

is called a *linear stationary iteration.* The primary result governing the convergence of (15.1.21) is the fact that if $\rho(\mathbf{H}) < 1$, then $\mathbf{A}$ is nonsingular, and

$$\lim_{k\to\infty} \mathbf{x}(k) = \mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \quad \text{(the solution to } \mathbf{Ax} = \mathbf{b}\text{) for every } \mathbf{x}(0). \tag{15.1.22}$$

In theory, the convergence rate of (15.1.21) is governed by the size of $\rho(\mathbf{H})$ along with the index of its associated eigenvalue—look at (15.1.13). But for practical work an indication of how many digits of accuracy can be expected to be gained per iteration is needed. Suppose that $\mathbf{H}_{n\times n}$ is diagonalizable with

$$\sigma(\mathbf{H}) = \{\lambda_1, \lambda_2, \ldots, \lambda_s\}, \quad \text{where} \quad 1 > |\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_s|$$

(which is frequently the case in applications), and let $\boldsymbol{\epsilon}(k) = \mathbf{x}(k) - \mathbf{x}$ denote the error after the k^{th} iteration. Subtracting $\mathbf{x} = \mathbf{Hx} + \mathbf{d}$ (the limiting value in (15.1.21)) from $\mathbf{x}(k) = \mathbf{Hx}(k-1) + \mathbf{d}$ produces (for large k)

$$\boldsymbol{\epsilon}(k) = \mathbf{H}\boldsymbol{\epsilon}(k-1) = \mathbf{H}^k\boldsymbol{\epsilon}(0) = (\lambda_1^k\mathbf{G}_1 + \lambda_2^k\mathbf{G}_2 + \cdots + \lambda_s^k\mathbf{G}_s)\boldsymbol{\epsilon}(0) \approx \lambda_1^k\mathbf{G}_1\boldsymbol{\epsilon}(0),$$

where the $\mathbf{G}_i$'s are the spectral projectors occurring in the spectral decomposition (15.1.11) of $\mathbf{H}^k$. Similarly, $\boldsymbol{\epsilon}(k-1) \approx \lambda_1^{k-1}\mathbf{G}_1\boldsymbol{\epsilon}(0)$, so comparing the i^{th} components of $\boldsymbol{\epsilon}(k-1)$ and $\boldsymbol{\epsilon}(k)$ reveals that after several iterations,

$$\left|\frac{\boldsymbol{\epsilon}_i(k-1)}{\boldsymbol{\epsilon}_i(k)}\right| \approx \frac{1}{|\lambda_1|} = \frac{1}{\rho(\mathbf{H})} \quad \text{for each} \quad i = 1, 2, \ldots, n.$$

To understand the significance of this, suppose for example that

$$|\boldsymbol{\epsilon}_i(k-1)| = 10^{-q} \quad \text{and} \quad |\boldsymbol{\epsilon}_i(k)| = 10^{-p} \quad \text{with} \quad p \geq q > 0,$$

so that the error in each entry is reduced by $p - q$ digits per iteration, and we have

$$p - q = \log_{10}\left|\frac{\boldsymbol{\epsilon}_i(k-1)}{\boldsymbol{\epsilon}_i(k)}\right| \approx -\log_{10}\rho(\mathbf{H}).$$

Below is a summary.

Asymptotic Convergence Rate

The number $R = -\log_{10}\rho(\mathbf{H})$, called the *asymptotic convergence rate* for (15.1.21), is used to compare different linear stationary iterative algorithms because it is an indication of the number of digits of accuracy that can be expected to be eventually gained on each iteration of $\mathbf{x}(k) = \mathbf{Hx}(k-1) + \mathbf{d}$.

Each different splitting $\mathbf{A} = \mathbf{M} - \mathbf{N}$ produces a different iterative algorithm, but there are three particular splittings that have found widespread use.

The Three Classical Splittings

- **Jacobi's method** is the result of splitting $\mathbf{A} = \mathbf{D} - \mathbf{N}$, where $\mathbf{D}$ is the diagonal part of $\mathbf{A}$ (assuming each $a_{ii} \neq 0$), and $(-\mathbf{N})$ is the matrix containing the off-diagonal entries of $\mathbf{A}$. The Jacobi iteration is $\mathbf{x}(k) = \mathbf{D}^{-1}\mathbf{N}\mathbf{x}(k-1) + \mathbf{D}^{-1}\mathbf{b}$.
- **The Gauss-Seidel method** is the result of splitting $\mathbf{A} = (\mathbf{D}-\mathbf{L}) - \mathbf{U}$, where $\mathbf{D}$ is the diagonal part of $\mathbf{A}$ (assuming each $a_{ii} \neq 0$), and where $(-\mathbf{L})$ and $(-\mathbf{U})$ contain the entries occurring below and above the diagonal of $\mathbf{A}$, respectively. The iteration matrix is $\mathbf{H} = (\mathbf{D} - \mathbf{L})^{-1}\mathbf{U}$, and $\mathbf{d} = (\mathbf{D} - \mathbf{L})^{-1}\mathbf{b}$. The Gauss-Seidel iteration is $\mathbf{x}(k) = (\mathbf{D} - \mathbf{L})^{-1}\mathbf{U}\mathbf{x}(k-1) + (\mathbf{D} - \mathbf{L})^{-1}\mathbf{b}$.
- **The successive overrelaxation (SOR) method** incorporates a *relaxation parameter* $\omega \neq 0$ into the Gauss-Seidel method to build a splitting $\mathbf{A} = \mathbf{M} - \mathbf{N}$, where $\mathbf{M} = \omega^{-1}\mathbf{D} - \mathbf{L}$ and $\mathbf{N} = (\omega^{-1} - 1)\mathbf{D} + \mathbf{U}$.

It can be shown that Jacobi's method as well as the Gauss-Seidel method converge when $\mathbf{A}$ is *diagonally dominant* (i.e., when $|a_{ii}| > \sum_{j \neq i} |a_{ij}|$ for each $i = 1, 2, \ldots, n$.) This along with other convergence details can be found in [127].

M-matrices

Because the PageRank vector can be view as the solution to a Markov chain, and because $\mathbf{I} - \mathbf{P}$ is an M-matrix whenever $\mathbf{P}$ is a probability transition matrix, it's handy to know a few facts about M-matrices (named in honor Hermann Minkowski).

M-matrices

A square (real) matrix $\mathbf{A}$ is called an M-matrix whenever there exists a matrix $\mathbf{B} \geq \mathbf{0}$ (i.e., $b_{ij} \geq 0$) and a real number $r \geq \rho(\mathbf{B})$ such that $\mathbf{A} = r\mathbf{I} - \mathbf{B}$.

If $r > \rho(\mathbf{B})$ in the above definition then $\mathbf{A}$ is a *nonsingular* M-matrix. Below are some of the important properties of nonsingular M-matrices.

- $\mathbf{A}$ is a nonsingular M-matrix if and only if $a_{ij} \leq 0$ for all $i \neq j$ and $\mathbf{A}^{-1} \geq \mathbf{0}$.
- If $\mathbf{A}$ is a nonsingular M-matrix, then $\text{Re}(\lambda) > 0$ for all $\lambda \in \sigma(\mathbf{A})$. Conversely, all matrices with nonpositive off-diagonal entries whose spectrums are in the right-hand halfplane are nonsingular M-matrices.
- Principal submatrices of nonsingular M-matrices are also nonsingular M-matrices.
- If $\mathbf{A}$ is an M-matrix, then all of its principal minors are nonnegative. If $\mathbf{A}$ is a nonsingular M-matrix, then all principal minors are positive.

- All matrices with nonpositive off-diagonal entries whose principal minors are nonnegative are M-matrices. All matrices with nonpositive off-diagonal entries whose principal minors are positive are nonsingular M-matrices.

- If $\mathbf{A} = \mathbf{M} - \mathbf{N}$ is a splitting of a nonsingular M-matrix for which $\mathbf{M}^{-1} \geq \mathbf{0}$, then the linear stationary iteration (15.1.21) converges for all initial vectors $\mathbf{x}(0)$ and for all right-hand sides $\mathbf{b}$. In particular, Jacobi's method converges.

15.2 PERRON–FROBENIUS THEORY

At a mathematics conference held a few years ago our friend Hans Schneider gave a memorable presentation titled "Why I Love Perron–Frobenius" in which he made the case that the Perron–Frobenius theory of nonnegative matrices is not only among the most elegant theories in mathematics, but it is also among the most useful. One might sum up Hans's point by saying that Perron–Frobenius is a testament to the fact that beautiful mathematics eventually tends to be useful, and useful mathematics eventually tends to be beautiful. The applications involving PageRank, HITS, and other ranking schemes [103] help to underscore this principle.

A matrix $\mathbf{A}$ is said to be *nonnegative* when each entry is a nonnegative number (denote this by writing $\mathbf{A} \geq \mathbf{0}$). Similarly, $\mathbf{A}$ is a *positive matrix* when each $a_{ij} > 0$ (write $\mathbf{A} > \mathbf{0}$). For example, the hyperlink matrix $\mathbf{H}$ and the stochastic matrix $\mathbf{S}$ (from Chapter 4) that are at the foundation of PageRank are nonnegative matrices, and the Google matrix $\mathbf{G}$ is a positive matrix. Consequently, properties of positive and nonnegative matrices govern the behavior of PageRank, and the Perron–Frobenius theory reveals these properties by describing the nature of the dominant eigenvalues and eigenvectors of positive and nonnegative matrices.

Perron

So much of the mathematics of PageRank, HITS, and associated ideas involves nonnegative matrices and graphs. This section provides you with the needed ammunition to handle these concepts. Perron's 1907 theorem provides the insight for understanding the eigenstructure of positive matrices. Perron's theorem for positive matrices is stated below, and the proof is in [127].

Frobenius Form

For each imprimitive matrix $\mathbf{A}$ with index of imprimitivity $h > 1$, there exists a permutation matrix $\mathbf{P}$ such that

$$\mathbf{P}^T\mathbf{A}\mathbf{P} = \begin{pmatrix} \mathbf{0} & \mathbf{A}_{12} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{23} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{A}_{h-1,h} \\ \mathbf{A}_{h1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{pmatrix}, \qquad (15.2.5)$$

where the zero blocks on the main diagonal are square.

15.3 MARKOV CHAINS

The mathematical component of Google's PageRank vector is the stationary distribution of a discrete-time, finite-state Markov chain. So, to understand and analyze the mathematics of PageRank, it's necessary to have an appreciation of Markov chain concepts, and that's the purpose of this section. Let's begin with some definitions.

- A *stochastic matrix* is a nonnegative matrix $\mathbf{P}_{n\times n}$ in which each row sum is equal to 1. Some authors say "row-stochastic" to distinguish this from the case when each column sum is 1.
- A *stochastic process* is a set of random variables $\{X_t\}_{t=0}^{\infty}$ having a common range $\{S_1, S_2, \ldots, S_n\}$, which is called the *state space* for the process. Parameter t is generally thought of as time, and X_t represents the state of the process at time t. For example, consider the process of surfing the Web by successively clicking on links to move from one Web page to another. The state space is the set of all Web pages, and the random variable X_t is the Web page being viewed at time t.
    - To emphasize that time is considered discretely rather than continuously the phrase "*discrete-time* process" is often used, and the phrase "*finite-state* process" can be used to emphasize that the state space is finite rather than infinite. Our discussion is limited to discrete-time finite-state processes.
- A *Markov chain* is a stochastic process that satisfies the *Markov property*

  $$P(X_{t+1} = S_j \,|\, X_t{=}S_{i_t},\, X_{t-1}{=}S_{i_{t-1}},\, \ldots, X_0{=}S_{i_0}) = P(X_{t+1} = S_j \,|\, X_t = S_{i_t})$$

  for each $t = 0, 1, 2, \ldots$. The notation $P(E \,|\, F)$ denotes the conditional probability that event E occurs given event F occurs—a review some elementary probability is in order if this is not already a familiar concept.
    - The Markov property asserts that the process is memoryless in the sense that the state of the chain at the next time period depends only on the current state and not on the past history of the chain. For example, the process of surfing the Web is a Markov chain provided that the next page that the Web surfer visits doesn't depend on the pages that were visited in the past—the choice depends

only on the current page. In other words, if the surfer randomly selects a link on the current page in order to get to the next Web page, then the process is a Markov chain. This kind of chain is referred to as a *random walk* on the link structure of the Web.

- The *transition probability* $p_{ij}(t) = P(X_t = S_j \,|\, X_{t-1} = S_i)$ is the probability of being in state S_j at time t given that the chain is in state S_i at time $t-1$, so think of this simply as the probability of moving from S_i to S_j at time t.

- The *transition probability matrix* $\mathbf{P}_{n\times n}(t) = [p_{ij}(t)]$ is clearly a nonnegative matrix, and a little thought should convince you that each row sum must be 1. In other words, $\mathbf{P}(t)$ is a stochastic matrix for each t.

- A *stationary Markov chain* is a chain in which the transition probabilities do not vary with time—i.e., $p_{ij}(t) = p_{ij}$ for all t. Stationary chains are also known as *homogeneous chains.*
  - In this case the transition probability matrix is a constant stochastic matrix $\mathbf{P} = [p_{ij}]$. Stationarity is assumed in the sequel.
  - In such a way, every Markov chain defines a stochastic matrix, but the converse is also true—every stochastic matrix $\mathbf{P}_{n\times n}$ defines an n-state Markov chain because the entries p_{ij} define a set of transition probabilities that can be interpreted as a stationary Markov chain on n states.

- An *irreducible Markov chain* is a chain for which the transition probability matrix $\mathbf{P}$ is an irreducible matrix. A chain is said to be *reducible* when $\mathbf{P}$ is a reducible matrix.
  - A *periodic Markov chain* is an irreducible chain whose transition probability matrix $\mathbf{P}$ is an imprimitive matrix. These chains are called periodic because each state can be occupied only at periodic points in time, where the period is the index of imprimitivity. For example, consider an irreducible chain whose index of imprimitivity is $h = 3$. The Frobenius form (15.2.5) means that the states can be reorder (relabeled) to create three clusters of states for which the transition matrix and its powers have the form

    $$\mathbf{P} = \begin{pmatrix} 0 & \star & 0 \\ 0 & 0 & \star \\ \star & 0 & 0 \end{pmatrix}, \; \mathbf{P}^2 = \begin{pmatrix} 0 & 0 & \star \\ \star & 0 & 0 \\ 0 & \star & 0 \end{pmatrix}, \; \mathbf{P}^3 = \begin{pmatrix} \star & 0 & 0 \\ 0 & \star & 0 \\ 0 & 0 & \star \end{pmatrix}, \; \mathbf{P}^4 = \begin{pmatrix} 0 & \star & 0 \\ 0 & 0 & \star \\ \star & 0 & 0 \end{pmatrix} \cdots,$$

    where this pattern continues indefinitely. If the chain begins in a state in cluster i, then this periodic pattern ensures that the chain can occupy a state in cluster i only at the end of every third step—see transient properties on page 179.
  - An *aperiodic Markov chain* is an irreducible chain whose transition probability matrix $\mathbf{P}$ is a primitive matrix.

- A *probability distribution vector* (or "probability vector" for short) is defined to be a nonnegative row vector $\mathbf{p}^T = (p_1, p_2, \ldots, p_n)$ such that $\sum_k p_k = 1$. (Every row in a stochastic matrix is probability vector.)

- A *stationary probability distribution vector* for a Markov chain whose transition probability matrix is $\mathbf{P}$ is a probability vector $\boldsymbol{\pi}^T$ such that $\boldsymbol{\pi}^T\mathbf{P} = \boldsymbol{\pi}^T$.

- The k^{th}*step probability distribution vector* for an n-state chain is defined to be

  $$\mathbf{p}^T(k) = \big(p_1(k), p_2(k), \ldots, p_n(k)\big), \quad \text{where} \quad p_j(k) = P(X_k = Sj).$$

  In other words, $p_j(k)$ is the probability of being in the j^{th} state after the k^{th} step, but before the $(k+1)^{st}$ step.

- The *initial distribution vector* is

  $$\mathbf{p}^T(0) = \big(p_1(0), p_2(0), \ldots, p_n(0)\big), \quad \text{where} \quad p_j(0) = P(X_0 = Sj).$$

  In other words, $p_j(0)$ is the probability that the chain starts in S_j.

To illustrate these concepts, consider the tiny three-page web shown in Figure 15.2,

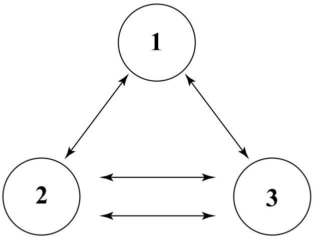

Figure 15.2

where the arrows indicate links—e.g., page 2 contains two links to page 3, and vice versa. The Markov chain defined by a random walk on this link structure evolves as a Web surfer clicks on a randomly selected link on the page currently being viewed, and the transition probability matrix for this chain is the irreducible stochastic matrix

$$\mathbf{H} = \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1/3 & 0 & 2/3 \\ 1/3 & 2/3 & 0 \end{pmatrix}.$$

In this example $\mathbf{H}$ (the *hyperlink matrix*) is stochastic, but if there had been a dangling node (a page containing no links to click on), then $\mathbf{H}$ would have a zero row, in which case $\mathbf{H}$ would not be stochastic and the process would not be a Markov chain. [1]

If our Web surfer starts on page 2 in Figure 15.2, then the initial distribution vector for the chain is $\mathbf{p}^T(0) = (0,1,0) = \mathbf{e}_2^T$. But if the surfer simply selects an initial page at random, then $\mathbf{p}^T(0) = (1/3, 1/3, 1/3) = \mathbf{e}^T/3$ is the *uniform distribution vector.* A standard eigenvalue calculation reveals that $\sigma(\mathbf{H}) = \{1, -1/3, /, -2/3\}$, so it's apparent that $\mathbf{H}$ is a nonnegative matrix having spectral radius $\rho(\mathbf{H}) = 1$.

The fact that $\rho(\mathbf{H}) = 1$ is a feature of all stochastic matrices $\mathbf{P}_{n\times n}$ because having row sums equal to 1 means that $\|\mathbf{P}\|_\infty = 1$ or, equivalently, $\mathbf{Pe} = \mathbf{e}$, where $\mathbf{e}$ is the column of all 1's. Because $(1, \mathbf{e})$ is an eigenpair for every stochastic matrix, and because $\rho(\star) \leq \|\star\|$ for every matrix norm, it follows that it follows that

$$1 \leq \rho(\mathbf{P}) \leq \|\mathbf{P}\|_\infty = 1 \implies \rho(\mathbf{P}) = 1. \tag{15.3.1}$$

[1] As explained earlier, this is why Google alters the raw hyperlink matrix before computing PageRank.

Furthermore, $\mathbf{e}$ is a positive eigenvector associated with $\rho(\mathbf{P}) = 1$. But be careful! This doesn't mean that you necessarily can call $\mathbf{e}$ the Perron vector for $\mathbf{P}$ because $\mathbf{P}$ might not be irreducible, [2] e.g., consider $\mathbf{P} = \begin{pmatrix} .5 & .5 \\ 0 & 1 \end{pmatrix}$.

Almost all Markovian analysis revolves around questions concerning the transient behavior of the chain as well as the limiting behavior, and standard goals are as follows.

- Describe the k^{th} step distribution $\mathbf{p}^T(k)$ for any initial distribution vector $\mathbf{p}^T(0)$.
- Determine if $\lim_{k\to\infty} \mathbf{p}^T(k)$ exists, and, if so, find the value of $\lim_{k\to\infty} \mathbf{p}^T(k)$.
- When $\lim_{k\to\infty} \mathbf{p}^T(k)$ doesn't exist, determine if the Cesàro limit
$$\lim_{k\to\infty}\left[\frac{\mathbf{p}^T(0) + \mathbf{p}^T(1) + \cdots + \mathbf{p}^T(k-1)}{k}\right]$$
exists, and, if so, find its value and interpret its meaning.

Transient Behavior

Given an initial distribution vector $\mathbf{p}^T(0) = \big(p_1(0), p_2(0), \ldots, p_n(0)\big)$, the first aim is to calculate the probability of being in any given state after the first transition (but before the second)—i.e., determine $\mathbf{p}^T(1) = \big(p_1(1), p_2(1), \ldots, p_n(1)\big)$. Let $\wedge$ and $\vee$ respectively denote *AND* and *OR* . It follows from elementary probability theory that for each j,

$$\begin{aligned} p_j(1) = P(X_1{=}S_j) &= P\Big[X_1{=}S_j \wedge (X_0{=}S_1 \vee X_0{=}S_2 \vee \cdots \vee X_0{=}S_n)\Big] \\ &= P\Big[(X_1{=}S_j \wedge X_0{=}S_1) \vee (X_1{=}S_j \wedge X_0{=}S_2) \vee \cdots \vee (X_1{=}S_j \wedge X_0{=}S_n)\Big] \\ &= \sum_{i=1}^{n} P\Big[X_1{=}S_j \wedge X_0{=}S_i\Big] = \sum_{i=1}^{n} P\Big[X_0 = S_i\Big] P\Big[X_1 = S_j \,|\, X_0 = Si\Big] \\ &= \sum_{i=1}^{n} p_i(0)p_{ij}. \end{aligned}$$

In other words, $\mathbf{p}^T(1) = \mathbf{p}^T(0)\mathbf{P}$, which describes the evolution from the initial distributions to the distribution after one step. The "no memory" Markov property provides the state of affairs at the end of two steps—it says to simply start over but with $\mathbf{p}^T(1)$ as the initial distribution. Consequently, $\mathbf{p}^T(2) = \mathbf{p}^T(1)\mathbf{P}$, and $\mathbf{p}^T(3) = \mathbf{p}^T(2)\mathbf{P}$, etc., and successive substitution yields

$$\mathbf{p}^T(k) = \mathbf{p}^T(0)\mathbf{P}^k, \tag{15.3.2}$$

which is simply a special case of the power method (15.1.20) except that left-hand vector-matrix multiplication is used. Furthermore, if $\mathbf{P}^k = \big[p_{ij}^{(k)}\big]$, then setting $\mathbf{p}^T(0) = \mathbf{e}_i^T$ in (15.3.2) yields $p_j(k) = p_{ij}^{(k)}$ for each $i = 1, 2, \ldots, n$. Below is a summary.

[2] The need to force irreducibility is another reason why Google modifies the raw hyperlink matrix.

Transient Properties

If $\mathbf{P}$ is the transition probability matrix for a Markov chain on states $\{S_1, S_2, \ldots, S_n\}$, then each of the following is true.

- The matrix $\mathbf{P}^k$ represents the *k-step transition probability matrix* in the sense that its (i, j)-entry $[\mathbf{P}^k]_{ij} = p_{ij}^{(k)}$ is the probability of moving from S_i to S_j in exactly k steps.
- The k^{th} step distribution vector is given by $\mathbf{p}^T(k) = \mathbf{p}^T(0)\mathbf{P}^k$.

Limiting Behavior

Analyzing limiting properties of Markov chains requires that the class of stochastic matrices (and hence the class of stationary Markov chains) be divided into four mutually exclusive categories.

(1)	$\mathbf{P}$ is irreducible	with	$\lim_{k\to\infty} \mathbf{P}^k$ existing	(i.e., $\mathbf{P}$ is primitive).
(2)	$\mathbf{P}$ is irreducible	with	$\lim_{k\to\infty} \mathbf{P}^k$ not existing	(i.e., $\mathbf{P}$ is imprimitive).
(3)	$\mathbf{P}$ is reducible	with	$\lim_{k\to\infty} \mathbf{P}^k$ existing.	
(4)	$\mathbf{P}$ is reducible	with	$\lim_{k\to\infty} \mathbf{P}^k$ not existing.	

In case (1) (an aperiodic chain) $\lim_{k\to\infty} \mathbf{P}^k$ can be easily evaluated. The Perron vector for $\mathbf{P}$ is $\mathbf{e}/n$ (the uniform distribution vector), so if $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_n)^T$ is the Perron vector for $\mathbf{P}^T$, (i.e., $\boldsymbol{\pi}^T\mathbf{P} = \boldsymbol{\pi}^T$) then, by (15.2.4),

$$\lim_{k\to\infty} \mathbf{P}^k = \frac{(\mathbf{e}/n)\boldsymbol{\pi}^T}{\boldsymbol{\pi}^T(\mathbf{e}/n)} = \frac{\mathbf{e}\boldsymbol{\pi}^T}{\boldsymbol{\pi}^T\mathbf{e}} = \mathbf{e}\boldsymbol{\pi}^T = \begin{pmatrix} \pi_1 & \pi_2 & \cdots & \pi_n \\ \pi_1 & \pi_2 & \cdots & \pi_n \\ \vdots & \vdots & & \vdots \\ \pi_1 & \pi_2 & \cdots & \pi_n \end{pmatrix} > \mathbf{0}. \tag{15.3.3}$$

Therefore, if $\mathbf{P}$ is primitive, then a limiting probability distribution exists and is given by

$$\lim_{k\to\infty} \mathbf{p}^T(k) = \lim_{k\to\infty} \mathbf{p}^T(0)\mathbf{P}^k = \mathbf{p}^T(0)\mathbf{e}\boldsymbol{\pi}^T = \boldsymbol{\pi}^T. \tag{15.3.4}$$

Notice that because $\sum_k p_k(0) = 1$, the term $\mathbf{p}^T(0)\mathbf{e}$ drops away, so the value of the limit is *independent* of the value of the initial distribution $\mathbf{p}^T(0)$, which isn't too surprising.

In case (2), where $\mathbf{P}$ is irreducible but imprimitive, (15.2.4) insures that $\lim_{k\to\infty} \mathbf{P}^k$ cannot exist, and hence $\lim_{k\to\infty} \mathbf{p}^T(k)$ cannot exist (otherwise taking $\mathbf{p}^T(0) = \mathbf{e}_i^T$ for each i would insure that $\mathbf{P}^k$ has a limit). However, the results on page 173 insure that the eigenvalues of $\mathbf{P}$ lying on the unit circle are each simple, so, by (15.1.18), $\mathbf{P}$ is Cesàro summable to the spectral projector $\mathbf{G}$ associated with the eigenvalue $\lambda = 1$. By recalling (15.1.12) and using the fact that $\mathbf{e}/n$ is the Perron vector for $\mathbf{P}$, it follows that if

$\boldsymbol{\pi}^T = (\pi_1, \pi_2, \ldots, \pi_n)$ is the left-hand Perron vector, then

$$\lim_{k\to\infty} \frac{\mathbf{I}+\mathbf{P}+\cdots+\mathbf{P}^{k-1}}{k} = \frac{(\mathbf{e}/n)\boldsymbol{\pi}^T}{\boldsymbol{\pi}^T(\mathbf{e}/n)} = \frac{\mathbf{e}\boldsymbol{\pi}^T}{\boldsymbol{\pi}^T\mathbf{e}} = \mathbf{e}\boldsymbol{\pi}^T = \begin{pmatrix} \pi_1 & \pi_2 & \cdots & \pi_n \\ \pi_1 & \pi_2 & \cdots & \pi_n \\ \vdots & \vdots & & \vdots \\ \pi_1 & \pi_2 & \cdots & \pi_n \end{pmatrix},$$

which is exactly the same form as the limit (15.3.3) for the primitive case. Consequently, the k^{th} step distributions have a Cesàro limit given by

$$\begin{aligned}\lim_{k\to\infty} \left[\frac{\mathbf{p}^T(0)+\mathbf{p}^T(1)+\cdots+\mathbf{p}^T(k-1)}{k}\right] &= \lim_{k\to\infty} \mathbf{p}^T(0)\left[\frac{\mathbf{I}+\mathbf{P}+\cdots+\mathbf{P}^{k-1}}{k}\right] \\ &= \mathbf{p}^T(0)\mathbf{e}\boldsymbol{\pi}^T = \boldsymbol{\pi}^T,\end{aligned}$$

and, just as in the primitive case (15.3.4), this Cesàro limit is independent of the initial distribution. To interpret the meaning of this Cesàro limit, focus on one state, say S_j, and let $\{Z_k\}_{k=0}^{\infty}$ be random variables that count the number of visits to S_j by setting

$$Z_0 = \begin{cases} 1 & \text{if the chain starts in } S_j, \\ 0 & \text{otherwise,} \end{cases}$$

and for $i > 1$,

$$Z_i = \begin{cases} 1 & \text{if the chain is in } S_j \text{ after the } i^{th} \text{ move,} \\ 0 & \text{otherwise.} \end{cases}$$

Notice that $Z_0 + Z_1 + \cdots + Z_{k-1}$ counts the number of visits to S_j before the k^{th} move, so $(Z_0 + Z_1 + \cdots + Z_{k-1})/k$ represents the fraction of times that S_j is hit before the k^{th} move. The expected (or mean) value of each Z_i is

$$E[Z_i] = 1 \cdot P(Z_i{=}1) \;+\; 0 \cdot P(Z_i{=}0) = P(Z_i{=}1) = p_j(i).$$

Since expectation is linear, the expected fraction of times that S_j is hit before move k is

$$\begin{aligned} E\left[\frac{Z_0+Z_1+\cdots+Z_{k-1}}{k}\right] &= \frac{E[Z_0]+E[Z_1]+\cdots+E[Z_{k-1}]}{k} \\ = \frac{p_j(0)+p_j(1)+\cdots+p_j(k-1)}{k} &= \left[\frac{\mathbf{p}^T(0)+\mathbf{p}^T(1)+\cdots+\mathbf{p}^T(k-1)}{k}\right]_j \\ &\to \pi_j. \end{aligned}$$

In other words, the long-run fraction of time that the chain spends in S_j is π_j, which is the j^{th} component of the Cesàro limit or, equivalently, the j^{th} component of the left-hand Perron vector for $\mathbf{P}$. When $\lim_{k\to\infty} \mathbf{p}^T(k)$ exists, it is easily argued that

$$\lim_{k\to\infty} \mathbf{p}^T(k) = \lim_{k\to\infty} \left[\frac{\mathbf{p}^T(0){+}\mathbf{p}^T(1){+}\cdots{+}\mathbf{p}^T(k{-}1)}{k}\right]$$

so the interpretation of the limiting distribution $\lim_{k\to\infty} \mathbf{p}^T(k)$ for the primitive case is exactly the same as the interpretation of the Cesàro limit in the imprimitive case. Below is a summary of irreducible chains.

Irreducible Markov Chains

Let $\mathbf{P}$ be the transition probability matrix for an irreducible Markov chain on states $\{S_1, S_2, \ldots, S_n\}$, and let $\boldsymbol{\pi}^T$ be the left-hand Perron vector for $\mathbf{P}$ (i.e., $\boldsymbol{\pi}^T\mathbf{P} = \boldsymbol{\pi}^T$, $\|\boldsymbol{\pi}\|_1 = 1$). The following hold for every initial distribution $\mathbf{p}^T(0)$.

- The k^{th} step transition matrix is $\mathbf{P}^k$. In other words, the (i,j)-entry in $\mathbf{P}^k$ is the probability of moving from S_i to S_j in exactly k steps.
- The k^{th} step distribution vector is given by $\mathbf{p}^T(k) = \mathbf{p}^T(0)\mathbf{P}^k$.
- If $\mathbf{P}$ is primitive (so the chain is aperiodic), and if $\mathbf{e}$ is the column of all 1's, then

$$\lim_{k\to\infty} \mathbf{P}^k = \mathbf{e}\boldsymbol{\pi}^T \lim_{k\to\infty} \mathbf{p}^T(k) = \boldsymbol{\pi}^T.$$

- If $\mathbf{P}$ is imprimitive (so the chain is periodic), then

$$\lim_{k\to\infty} \frac{\mathbf{I} + \mathbf{P} + \cdots + \mathbf{P}^{k-1}}{k} = \mathbf{e}\boldsymbol{\pi}^T$$

  and

$$\lim_{k\to\infty} \left[\frac{\mathbf{p}^T(0)+\mathbf{p}^T(1)+\cdots+\mathbf{p}^T(k-1)}{k}\right] = \boldsymbol{\pi}^T.$$

- Regardless of whether $\mathbf{P}$ is primitive or imprimitive, the j^{th} component π_j of $\boldsymbol{\pi}^T$ represents the long-run fraction of time that the chain is in S_j.
- The vector $\boldsymbol{\pi}^T$ is the unique *stationary distribution vector* for the chain because it is the unique probability distribution vector satisfying $\boldsymbol{\pi}^T\mathbf{P} = \boldsymbol{\pi}^T$.

Reducible Markov Chains

The Perron–Frobenius theorem is not directly applicable to reducible chains (chains for which $\mathbf{P}$ is a reducible matrix), so the strategy for analyzing reducible chains is to deflate the situation, as much as possible, back to the irreducible case. If $\mathbf{P}$ is reducible, then, by definition, there is a permutation matrix $\mathbf{Q}$ and square matrices $\mathbf{X}$ and $\mathbf{Z}$ such that

$$\mathbf{Q}^T\mathbf{P}\mathbf{Q} = \begin{pmatrix} \mathbf{X} & \mathbf{Y} \\ \mathbf{0} & \mathbf{Z} \end{pmatrix}. \text{ For convenience, denote this by writing } \mathbf{P} \sim \begin{pmatrix} \mathbf{X} & \mathbf{Y} \\ \mathbf{0} & \mathbf{Z} \end{pmatrix}.$$

If $\mathbf{X}$ or $\mathbf{Z}$ is reducible, then another symmetric permutation can be performed to produce

$$\begin{pmatrix} \mathbf{X} & \mathbf{Y} \\ \mathbf{0} & \mathbf{Z} \end{pmatrix} \sim \begin{pmatrix} \mathbf{R} & \mathbf{S} & \mathbf{T} \\ \mathbf{0} & \mathbf{U} & \mathbf{V} \\ \mathbf{0} & \mathbf{0} & \mathbf{W} \end{pmatrix}, \quad \text{where } \mathbf{R},\ \mathbf{U}, \text{ and } \mathbf{W} \text{ are square.}$$

Repeating this process eventually yields

$$\mathbf{P} \sim \begin{pmatrix} \mathbf{X}_{11} & \mathbf{X}_{12} & \cdots & \mathbf{X}_{1k} \\ \mathbf{0} & \mathbf{X}_{22} & \cdots & \mathbf{X}_{2k} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{X}_{kk} \end{pmatrix}, \quad \text{where each } \mathbf{X}_{ii} \text{ is irreducible or } \mathbf{X}_{ii} = [0]_{1\times 1}.$$

Finally, if there exist rows having nonzero entries only in diagonal blocks, then symmetrically permute all such rows to the bottom to produce

$$\mathbf{P} \sim \left(\begin{array}{cccc|cccc} \mathbf{P}_{11} & \mathbf{P}_{12} & \cdots & \mathbf{P}_{1r} & \mathbf{P}_{1,r+1} & \mathbf{P}_{1,r+2} & \cdots & \mathbf{P}_{1m} \\ \mathbf{0} & \mathbf{P}_{22} & \cdots & \mathbf{P}_{2r} & \mathbf{P}_{2,r+1} & \mathbf{P}_{2,r+2} & \cdots & \mathbf{P}_{2m} \\ \vdots & & \ddots & \vdots & \vdots & \vdots & \cdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{P}_{rr} & \mathbf{P}_{r,r+1} & \mathbf{P}_{r,r+2} & \cdots & \mathbf{P}_{rm} \\ \hline \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{P}_{r+1,r+1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{P}_{r+2,r+2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{P}_{mm} \end{array}\right), \tag{15.3.5}$$

where each $\mathbf{P}_{11}, \ldots, \mathbf{P}_{rr}$ is either irreducible or $[0]_{1\times 1}$, and $\mathbf{P}_{r+1,r+1}, \ldots, \mathbf{P}_{mm}$ are irreducible (they can't be zero because each has row sums equal to 1). As mentioned on page 171, the effect of a symmetric permutation is simply to relabel nodes in $\mathcal{G}(\mathbf{P})$ or, equivalently, to reorder the states in the chain. When the states of a chain have been reordered so that $\mathbf{P}$ assumes the form on the right-hand side of (15.3.5), we say that $\mathbf{P}$ is in the *canonical form for reducible matrices.*

The results on page 173 guarantee that if an irreducible stochastic matrix $\mathbf{P}$ has h eigenvalues on the unit circle, then these h eigenvalues are the h^{th} roots of unity, and each is a simple eigenvalue for $\mathbf{P}$. The same can't be said for reducible stochastic matrices, but (15.3.5) leads to the next best result (the proof of which is in [127]).

Unit Eigenvalues

The *unit eigenvalues* are those eigenvalues that are on the unit circle. For every stochastic matrix $\mathbf{P}_{n\times n}$, the following statements are true.

- Every unit eigenvalue of $\mathbf{P}$ is semisimple.
- Every unit eigenvalue has form $\lambda = \mathrm{e}^{2k\pi\mathrm{i}/h}$ for some $k < h \leq n$.
- In particular, $\rho(\mathbf{P}) = 1$ is always a semisimple eigenvalue of $\mathbf{P}$.

The discussion on page 163 says that a matrix $\mathbf{A}_{n\times n}$ is Cesàro summable if and only if $\rho(\mathbf{A}) < 1$ or $\rho(\mathbf{A}) = 1$ with each eigenvalue on the unit circle being semisimple. Since the result above says that the latter holds for all stochastic matrices $\mathbf{P}$, we have the following powerful realization concerning all stochastic matrices.

All Stochastic Matrices Are Summable

Every stochastic matrix $\mathbf{P}$ is Cesàro summable in the sense that

$$\lim_{k\to\infty} \frac{\mathbf{I} + \mathbf{P} + \cdots + \mathbf{P}^{k-1}}{k} = \mathbf{G}$$

always exists and, as discussed on page 163, the value of the limit is the spectral projector $\mathbf{G}$ onto $N(\mathbf{I} - \mathbf{P})$ along $R(\mathbf{I} - \mathbf{P})$.

The structure and interpretation of the Cesàro limit when $\mathbf{P}$ is an irreducible stochastic matrix was developed on page 181 so to complete the picture all that remains is to analyze the nature of $\lim_{k\to\infty}(\mathbf{I}+\mathbf{P}+\cdots+\mathbf{P}^{k-1})/k$ for the reducible case.

Suppose that $\mathbf{P}=\begin{pmatrix}\mathbf{T}_{11} & \mathbf{T}_{12}\\ \mathbf{0} & \mathbf{T}_{22}\end{pmatrix}$ is a reducible stochastic matrix that is in the canonical form (15.3.5), where

$$\mathbf{T}_{11}=\begin{pmatrix}\mathbf{P}_{11} & \cdots & \mathbf{P}_{1r}\\ & \ddots & \vdots\\ & & \mathbf{P}_{rr}\end{pmatrix},\ \mathbf{T}_{12}=\begin{pmatrix}\mathbf{P}_{1,r+1} & \cdots & \mathbf{P}_{1m}\\ \vdots & & \vdots\\ \mathbf{P}_{r,r+1} & \cdots & \mathbf{P}_{rm}\end{pmatrix},\ \mathbf{T}_{22}=\begin{pmatrix}\mathbf{P}_{r+1,r+1} & & \\ & \ddots & \\ & & \mathbf{P}_{mm}\end{pmatrix}.$$

Because each row in $\mathbf{T}_{11}$ has a nonzero off-diagonal block, it follows that $\rho(\mathbf{P}_{kk})<1$ for each $k=1,2,\ldots,r$. Consequently, $\rho(\mathbf{T}_{11})<1$, and

$$\lim_{k\to\infty}\frac{\mathbf{I}+\mathbf{T}_{11}+\cdots+\mathbf{T}_{11}^{k-1}}{k}=\lim_{k\to\infty}\mathbf{T}_{11}^{k}=\mathbf{0}.$$

Furthermore, $\mathbf{P}_{r+1,r+1},\ldots,\mathbf{P}_{mm}$ are each irreducible stochastic matrices, so if $\boldsymbol{\pi}_j^T$ is the left-hand Perron vector for $\mathbf{P}_{jj}$, $r+1\le j\le m$, then (15.1.12) combined with (15.1.18) yields

$$\lim_{k\to\infty}\frac{\mathbf{I}+\mathbf{T}_{22}+\cdots+\mathbf{T}_{22}^{k-1}}{k}=\begin{pmatrix}\mathbf{e}\boldsymbol{\pi}_{r+1}^T & & \\ & \ddots & \\ & & \mathbf{e}\boldsymbol{\pi}_m^T\end{pmatrix}=\mathbf{E}.$$

It's clear from (15.2.4) that $\lim_{k\to\infty}\mathbf{T}_{22}^k$ exists if and only if $\mathbf{P}_{r+1,r+1},\ldots,\mathbf{P}_{mm}$ are each primitive, in which case $\lim_{k\to\infty}\mathbf{T}_{22}^k=\mathbf{E}$. Therefore, the limits, be they Cesàro or ordinary (if it exists), all have the form

$$\lim_{k\to\infty}\frac{\mathbf{I}+\mathbf{P}+\cdots+\mathbf{P}^{k-1}}{k}=\begin{pmatrix}\mathbf{0} & \mathbf{Z}\\ \mathbf{0} & \mathbf{E}\end{pmatrix}=\mathbf{G}=\lim_{k\to\infty}\mathbf{P}^k \text{ (when it exists).}$$

To determine the precise nature of $\mathbf{Z}$, use the fact that $R(\mathbf{G})=N(\mathbf{I}-\mathbf{P})$ (because $\mathbf{G}$ is the projector onto $N(\mathbf{I}-\mathbf{P})$ along $R(\mathbf{I}-\mathbf{P})$) to write

$$(\mathbf{I}-\mathbf{P})\mathbf{G}=\mathbf{0}\implies\begin{pmatrix}\mathbf{I}-\mathbf{T}_{11} & -\mathbf{T}_{12}\\ \mathbf{0} & \mathbf{I}-\mathbf{T}_{22}\end{pmatrix}\begin{pmatrix}\mathbf{0} & \mathbf{Z}\\ \mathbf{0} & \mathbf{E}\end{pmatrix}=\mathbf{0}\implies(\mathbf{I}-\mathbf{T}_{11})\mathbf{Z}=\mathbf{T}_{12}\mathbf{E}.$$

Since $\mathbf{I}-\mathbf{T}_{11}$ is nonsingular (because $\rho(\mathbf{T}_{11})<1$), it follows that

$$\mathbf{Z}=(\mathbf{I}-\mathbf{T}_{11})^{-1}\mathbf{T}_{12}\mathbf{E},$$

and thus the following results concerning limits of reducible chains are produced.

Reducible Markov Chains

If the states in a reducible Markov chain have been ordered to make the transition matrix assume the canonical form

$$\mathbf{P} = \begin{pmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} \\ \mathbf{0} & \mathbf{T}_{22} \end{pmatrix}$$

that is described in (15.3.5), and if $\boldsymbol{\pi}_j^T$ is the left-hand Perron vector for $\mathbf{P}_{jj}$ $(r+1 \le j \le m)$, then $\mathbf{I} - \mathbf{T}_{11}$ is nonsingular, and

$$\lim_{k\to\infty} \frac{\mathbf{I} + \mathbf{P} + \cdots + \mathbf{P}^{k-1}}{k} = \begin{pmatrix} \mathbf{0} & (\mathbf{I} - \mathbf{T}_{11})^{-1}\mathbf{T}_{12}\mathbf{E} \\ \mathbf{0} & \mathbf{E} \end{pmatrix},$$

where

$$\mathbf{E} = \begin{pmatrix} \mathbf{e}\boldsymbol{\pi}_{r+1}^T & & \\ & \ddots & \\ & & \mathbf{e}\boldsymbol{\pi}_m^T \end{pmatrix}.$$

Furthermore, $\lim_{k\to\infty} \mathbf{P}^k$ exists if and only if the stochastic matrices $\mathbf{P}_{r+1,r+1}, \ldots, \mathbf{P}_{mm}$ in (15.3.5) are each primitive, in which case

$$\lim_{k\to\infty} \mathbf{P}^k = \begin{pmatrix} \mathbf{0} & (\mathbf{I} - \mathbf{T}_{11})^{-1}\mathbf{T}_{12}\mathbf{E} \\ \mathbf{0} & \mathbf{E} \end{pmatrix}. \tag{15.3.6}$$

Transient and Ergodic Classes

When the states of a chain are reordered so that $\mathbf{P}$ is in canonical form (15.3.5), the subset of states corresponding to $\mathbf{P}_{kk}$ for $1 \le k \le r$ is called the k^{th} *transient class* because once left, a transient class can't be reentered. The subset of states corresponding to $\mathbf{P}_{r+j,r+j}$ for $j \ge 1$ is called the j^{th} *ergodic class.* Each ergodic class is an irreducible Markov chain unto itself that is imbedded in the larger reducible chain. From now on, we will assume that the states in reducible chains have been ordered so that $\mathbf{P}$ is in canonical form (15.3.5).

Every reducible chain eventually enters one of the ergodic classes, but what happens after that depends on whether or not the ergodic class is primitive. If $\mathbf{P}_{r+j,r+j}$ is primitive, then the chain settles down to a steady state defined by the left-hand Perron vector of $\mathbf{P}_{r+j,r+j}$, but if $\mathbf{P}_{r+j,r+j}$ is imprimitive, then the process will oscillate in the j^{th} ergodic class forever. There is not much more that can be said about the limit, but there are still important questions concerning which ergodic class the chain will end up in and how long it takes to get there. This time the answer depends on where the chain starts—i.e., on the initial distribution.

For convenience, let $\mathcal{T}_i$ denote the i^{th} transient class, and let $\mathcal{E}_j$ be the j^{th} ergodic class. Suppose that the chain starts in a particular transient state—say we start in the p^{th} state of $\mathcal{T}_i$. Since the question at hand concerns only which ergodic class is hit but not what happens after it's entered, we might as well convert every state in each ergodic class into a trap by setting $\mathbf{P}_{r+j,r+j} = \mathbf{I}$ for each $j \ge 1$ in (15.3.5). The transition matrix for this modified chain is $\widetilde{\mathbf{P}} = \begin{pmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$, and it follows from (15.3.6) that $\lim_{k\to\infty} \widetilde{\mathbf{P}}^k$ exists and has the form

$$\lim_{k\to\infty} \widetilde{\mathbf{P}}^k = \begin{pmatrix} \mathbf{0} & (\mathbf{I}-\mathbf{T}_{11})^{-1}\mathbf{T}_{12} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} = \left(\begin{array}{cccc|cccc} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{L}_{1,1} & \mathbf{L}_{1,2} & \cdots & \mathbf{L}_{1s} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{L}_{2,1} & \mathbf{L}_{2,2} & \cdots & \mathbf{L}_{2s} \\ \vdots & & \ddots & \vdots & \vdots & \vdots & \cdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{L}_{r,1} & \mathbf{L}_{r,2} & \cdots & \mathbf{L}_{rs} \\ \hline \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} \end{array}\right).$$

Consequently, the (p,q)-entry in block $\mathbf{L}_{ij}$ represents the probability of eventually hitting the q^{th} state in $\mathcal{E}_j$ given that we start from the p^{th} state in $\mathcal{T}_i$. Therefore, if $\mathbf{e}$ is the vector of all 1's, then the probability of eventually entering somewhere in $\mathcal{E}_j$ is given by

$$P(\text{absorption into } \mathcal{E}_j \,|\, \text{start in } p^{th} \text{ state of } \mathcal{T}_i) = \sum_k \left[\mathbf{L}_{ij}\right]_{pk} = \left[\mathbf{L}_{ij}\mathbf{e}\right]_p.$$

If $\mathbf{p}_i^T(0)$ is an initial distribution for starting in the various states of $\mathcal{T}_i$, then

$$P\big(\text{absorption into } \mathcal{E}_j \,|\, \mathbf{p}_i^T(0)\big) = \mathbf{p}_i^T(0)\mathbf{L}_{ij}\mathbf{e}.$$

The expected number of steps required to first hit an ergodic state is determined as follows. Count the number of times the chain is in transient state S_j given that it starts in transient state S_i by reapplying the argument given in on page 180. That is, given that the chain starts in S_i, let

$$Z_0 = \begin{cases} 1 & \text{if } S_i = S_j, \\ 0 & \text{otherwise,} \end{cases} \quad Z_k = \begin{cases} 1 & \text{if the chain is in } S_j \text{ after step } k, \\ 0 & \text{otherwise.} \end{cases}$$

Since

$$E[Z_k] = 1 \cdot P(Z_k{=}1) \;+\; 0 \cdot P(Z_k{=}0) = P(Z_k{=}1) = \left[\mathbf{T}_{11}^k\right]_{ij},$$

and since $\sum_{k=0}^{\infty} Z_k$ is the total number of times the chain is in S_j, we have

$$\begin{aligned} E[\#\text{ times in } S_j \,|\, \text{ start in } S_i] &= E\left[\sum_{k=0}^{\infty} Z_k\right] = \sum_{k=0}^{\infty} E\left[Z_k\right] = \sum_{k=0}^{\infty} \left[\mathbf{T}_{11}^k\right]_{ij} \\ &= \left[(\mathbf{I}-\mathbf{T}_{11})^{-1}\right]_{ij} \quad (\text{because } \rho\left(\mathbf{T}_{11}\right) < 1). \end{aligned}$$

Summing this over all transient states produces the expected number of times the chain is in *some* transient state, which is the same as the expected number of times before first hitting an ergodic state. In other words,

$$E[\#\text{ steps until absorption} \,|\, \text{start in } i^{th} \text{ transient state}] = \left[(\mathbf{I}-\mathbf{T}_{11})^{-1}\mathbf{e}\right]_i.$$

It's often the case in practical applications that there is only one transient class, and the ergodic classes are just single absorbing states (states such that once they are entered, they are never left). If the single transient class contains r states, and if there are s absorbing states, then the canonical form for the transition matrix is

$$\mathbf{P} = \left(\begin{array}{ccc|ccc} p_{11} & \cdots & p_{1r} & p_{1,r+1} & \cdots & p_{1s} \\ \vdots & & \vdots & \vdots & & \vdots \\ p_{r1} & \cdots & p_{rr} & p_{r,r+1} & \cdots & p_{rs} \\ \hline 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 1 \end{array}\right). \tag{15.3.7}$$

In this case, $\mathbf{L}_{ij} = \left[(\mathbf{I}-\mathbf{T}_{11})^{-1}\mathbf{T}_{12}\right]_{ij}$, and the earlier development specializes to say that every absorbing chain must eventually reach one of its absorbing states. The absorption probabilities and absorption times are included in the following summary.

Absorption Probabilities and Absorption Times

For a reducible chain whose transition matrix $\mathbf{P} = \begin{pmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} \\ \mathbf{0} & \mathbf{T}_{22} \end{pmatrix}$ is in the canonical form (15.3.5), let $\mathcal{T}_i$ and $\mathcal{E}_j$ be the i^{th} and j^{th} transient and ergodic classes, respectively, and let $\mathbf{p}_i^T(0)$ be an initial distribution for starting in the various states of $\mathcal{T}_i$. If $(\mathbf{I}-\mathbf{T}_{11})^{-1}\mathbf{T}_{12}$ is partitioned as

$$(\mathbf{I}-\mathbf{T}_{11})^{-1}\mathbf{T}_{12} = \begin{pmatrix} \mathbf{L}_{1,1} & \mathbf{L}_{1,2} & \cdots & \mathbf{L}_{1s} \\ \mathbf{L}_{2,1} & \mathbf{L}_{2,2} & \cdots & \mathbf{L}_{2s} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{L}_{r,1} & \mathbf{L}_{r,2} & \cdots & \mathbf{L}_{rs} \end{pmatrix},$$

then

- $P\big(\text{absorption into } \mathcal{E}_j \,|\, \mathbf{p}_i^T(0)\big) = \mathbf{p}_i^T(0)\mathbf{L}_{ij}\mathbf{e}$,
- $P(\text{absorption into } \mathcal{E}_j \,|\, \text{start in } p^{th} \text{ state of } \mathcal{T}_i) = \sum_k \left[\mathbf{L}_{ij}\right]_{pk} = \left[\mathbf{L}_{ij}\mathbf{e}\right]_p$,
- $E[\#\text{ steps until absorption} \,|\, \text{start in } i^{th} \text{ transient state}] = \left[(\mathbf{I}-\mathbf{T}_{11})^{-1}\mathbf{e}\right]_i$.

When there is only one transient class and each ergodic class is a single absorbing state ($\mathcal{E}_j = S_{r+j}$), $\mathbf{P}$ has the form (15.3.7). If S_i and S_j are transient states, then

- $P(\text{absorption into } S_{r+j} \,|\, \text{start in } S_i) = \left[(\mathbf{I}-\mathbf{T}_{11})^{-1}\mathbf{T}_{12}\right]_{ij}$,
- $E[\#\text{ steps until absorption} \,|\, \text{start in } S_i] = \left[(\mathbf{I}-\mathbf{T}_{11})^{-1}\mathbf{e}\right]_i$,
- $E[\#\text{ times in } S_j \,|\, \text{ start in } S_i] = \left[(\mathbf{I}-\mathbf{T}_{11})^{-1}\right]_{ij}$.

15.4 PERRON COMPLEMENTATION

The theory of stochastic complementation in section 15.5 concerns the development of methods that allow the stationary distribution of a large irreducible Markov chain to be obtained by gluing together stationary distributions of smaller chains. The concepts are based on the theory of Perron complementation, which describes how the Perron vector of a large irreducible matrix can be expressed in terms of Perron vectors of smaller matrices.