1. Ranking

Note. A *ranking* of subjects is an arrangement of the subjects where their relative positions reflect the possession of some particular quantifiable property. This property could be the number of wins by an athletic team, the popularity of a consumer item, or the preference for a particular political candidate in a field of several candidates (based on an opinion poll, say). In this chapter we'll consider transition probabilities, stochastic matrices, eigenvalues and eigenvectors, and the Perron-Frobenius Theorem. These ideas will be applied to ranking problems.

Section 1.1. Motivation: Google Problem

Note. To illustrate the "Google Problem," consider the simplified network N1 in Figure 1.1.

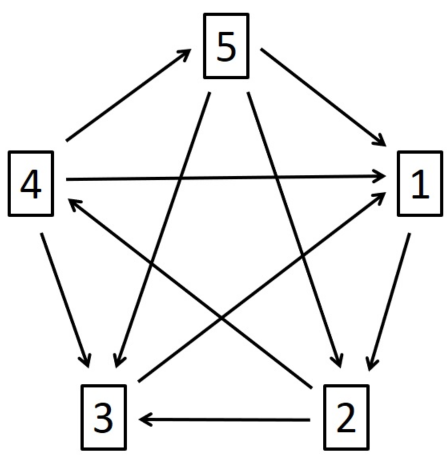

Figure 1.1. Network N1

This represents a network of web pages (numbered 1 through 5) and the arrows represent hyperlinks between the pages. We wish to rank the web pages according to some criteria. First, we consider a page "popular" sites if there are lots of other pages linked to it. This leads to a ranking (notice that we need to deal with "ties"):

Web page:	1	2	3	4	5
# of links to:	3	2	3	1	1
Rank:	I-II	III	I-II	IV-V	IV-V

Notice that pages 1 and 3 are tied for popularity, but page 1 is linked from pages 3, 4, 5 and page 3 is linked from pages 2, 4, 5. Comparing these incoming links, we see that 4 and 5 are shared and that these are equally popular based on our first estimate. But the third pages, page 3 for 1 and page 2 for 3, are *not* equally popular. Page 3 is more popular than page 2 so we could justify a claim that page 1 is linked from more popular sites than page 3 is (think of this as following a link from a starting page and then following a new link to a third page). This revised ranking is based on an idea that the most popular web pages themselves are linked from the (other) most popular web pages. This iterated process can be dealt with by introducing transition matrices, a topic potentially address in Linear Algebra (MATH 2010); see my online notes for Linear Algebra on Section 1.7. Applications to Population Distributions.

Definition. If a population of size n is partitioned into n ranks (or "states"), $1, 2, \ldots, n$, and the probability p_{ij} of transition from from rank j to rank i over a (discrete) time increment, then we create the $n \times n$ *transition matrix* $P = [p_{ij}]$.

Note. In the Google Problem, we take $p_{ij} = 0$ if there are no hyperlinks from page j to page i in the given network, and otherwise take $p_{ij} = 1/(\#$ of hyperlinks from $j)$ (so we are treating all links on page j as equally likely to be followed). Notice that the sum of the entries in each column of a transition matrix is 1. For network N1, the transition matrix is

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

We introduce the column vector $x = (x_1, x_2, \ldots, x_n)^T$ where $x_i \geq 0$ denotes the (initially unknown) ranking of the ith web page. Notice that we do not distinguish here between scalars and vectors; we do not use bold faced fonts, $\mathbf{v}$, nor little arrows, $\vec{v}$, to indicate vectors but instead use the context to determine with a lower case letter represents a vector or a scalar. Also, if all the components of vector x are nonnegative, we write $x \geq 0$. We are looking for a stable situation where the rankings do not change. That is, we want $x_i = \sum_{j=1}^{n} p_{ij} x_j$ for $i = 1, 2, \ldots, n$. As a matrix equation, this can be written as $x = Px$ where we want $x \geq 0$ and $x \neq 0$. We formalize this now with a definition.

Definition. The *Google Problem* is to find a nonzero vector $x \in \mathbb{R}^n$ with $x \geq 0$ such that, for transition matrix P defined from a network based on n web pages (as described above), we have $x = Px$. That is, we want:

$$x = Px,\ x \geq 0,\ x \neq 0. \qquad (\mathcal{R})$$

Note. The condition $x = Px$ for $n \neq 0$ is equivalent to x being an eigenvector of matrix P with associated eigenvalue $\lambda = 1$. We explore this in the next section.

Revised: 4/4/2021