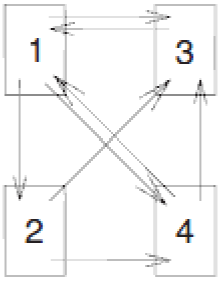
A Note on Google's PageRank

According to Google, google-search on a given topic results in a listing of most relevant web pages related to the topic. Google ranks the importance of webpages according to an eigenvector of a weighted link matrix. The following offers an insight into how this is done and is a basic application of the eigenvalue problem from linear algebra. It is based on the *Bryan, Leise paper.*

A Tiny Web Example

- Core idea: in assigning a score to any given web page, the page's score (ranking) is derived from the links made *to* that page from other web pages.
- The links *to* a given page are called the *backlinks* for that page
- The web is represented as a directed graph $G = (E, V)$ with vertices being the web pages and edges the links. There is a directed edge from page i to page j if page i contains a hyperlink to page j.
- Denote the importance score of page k by x_k ($x_i > x_j$ means that page i is more important)
- As in the paper, the approach that *doesn't work* is to take x_k as the number of backlinks for page k.

  E.g. here $x_1 = 2, x_2 = 1, x_3 = 3, x_4 = 2$:



But we want a link to page k from an important page to boost page k's importance score more than a link from an unimportant page. (A page's importance is presumably higher when, say, the US Supreme Court's webpage links to it than when just Joe Blow's web page links to it.)

E.g., in the above graph, pages 1 and 4 have the same score, but one of page 1's backlinks is from the seemingly important page 3 (which seems important because everybody else links to it), while one of page 4's backlinks is from the relatively unimportant page 1. Thus, we'd be rating page 1's importance higher than page 4's.

- In an attempt to fix this, we can try to compute the score of page j as the *sum* of the scores of all pages linking to page j.

  For example, the score of page 1 would be determined by the relation $x_1 = x_3 + x_4$, because pages 3 and 4 are 1's backlilnks and their scores are x_3 and x_4.

- However, there's a bit of a problem with this: we don't want a single individual webpage to gain influence merely by casting multiple votes (just as in elections, we don't want a single individual to gain undue influence by casting multiple votes)

- So we make a correction: if page j contains n_j *out*links, one of which links to page k, then we boost page k's score by x_j/n_j rather than by x_j.

- Notice that in this scheme each web page gets a total of one vote, weighted by that web page's score, that is evenly divided up among all of its outgoing links.

  Let $L_k = \{1, 2, \ldots, n\}$ denote the set of pages with a link to page k, that is, L_k is the set of k's backlinks. For each k require:

$$x_k = \sum_{j \in L_k} \frac{x_j}{n_j} \tag{1}$$

  where n_j is the set of outgoing links from page j.

Assigning a Score to a Page, an Example

- For the web in the above figure, using the outlined scheme, we have:

$$\begin{aligned} x_1 &= x_3/1 + x_4/2 \\ x_2 &= x_1/3 \\ x_3 &= x_1/3 + x_2/2 + x_4/2 \\ x_4 &= x_1/3 + x_2/3 \end{aligned}$$

- These linear equations can be written as $\mathbf{Ax} = \mathbf{x}$:

$$\begin{bmatrix} 0 & 0 & 1 & 1/2 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 1/2 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

- *Thus, we have reduced the web ranking problem to the problem of finding an eigenvector for the link matrix* $\mathbf{A}$*:* $\mathbf{Ax} = \lambda\mathbf{x}$. *In particular, we are looking for the eigenvector corresponding to the eigenvalue* $\lambda = 1$. (Note that $\mathbf{A}$ is not the graph adjacency matrix. $\mathbf{A}^T$ is the graph adjacency matrix.)

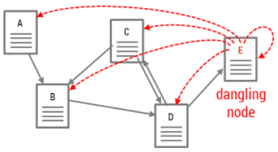


Figure 1: Example of a dangling node.

- If the web graph has no 'dangling' nodes (a dangling webpage is a page that has no outgoing links), e.g., see Fig. 1 then A will always have 1 as an eigenvalue - easy to show.

  By construction, the link matrix $\mathbf{A}$ is such that $A_{ij} = 1/n_j$ if page j links to page i and 0 otherwise[1]. Thus the j-th column of $\mathbf{A}$ contains n_j non-zero entries summing up to 1 (each is $1/n_j$). So $\mathbf{A}$'s columns all sum up to 1. Such matrix is called *column-stochastic*.

  Claim: Every column-stochastic matrix has 1 as an eigenvalue. (*proof*: take a vector of all ones, $\mathbf{e}$, and consider $\mathbf{A}^T\mathbf{e} = \mathbf{e}$, which obviously holds because the rows of $\mathbf{A}^T$ add up to one. Thus 1 is an eigenvalue of $\mathbf{A}^T$. Recalling that the eigenvlues of $\mathbf{A}^T$ and $\mathbf{A}$ are the same, proves the claim.)

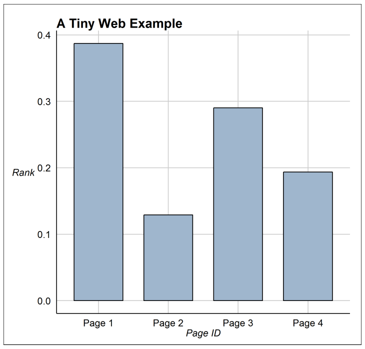
- In this small example the eigenvector corresponding to $\lambda = 1$ is easy to find 'by hand'. The following `MATLAB` code also finds the eigenvalus and eigenvectors of $\mathbf{A}$:

```
A = [  0    0    1   1/2;
      1/3   0    0    0 ;
      1/3  1/2   0   1/2;
      1/3  1/2   0    0 ];

[V D] = eig(A);

V(:,1)*(12/V(1,1))
% scale the eigenvectors
x = V(:,1)/sum(V(:,1))
```

  We can output (note that we agree to scale the eigenvector so that its components sum up to 1) and plot the rankings as well:

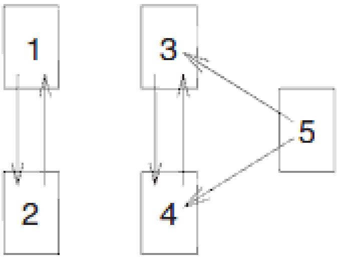
```
      12/31
       4/31
       9/31
       6/31
```

[1] The number of non-zero entries in the ith row of $\mathbf{A}$ is the *in-degree* of node i - i.e. how many other pages link to it. And the number of non-zero entries in the jth column is the *out-degree* of node j - i.e. how many other pages j links to.



Case of Disconnectd SubWebs

- There maybe more than one eigenvector that solves $\mathbf{Ax} = \mathbf{x}$ if the web graph is not connected, e.g., see the graph below. In this case the solution is not unique, and hence the dimensionality of the eigenspace $V_1(\mathbf{A})$ corresponding to $\lambda = 1$ is larger that one.



- To solve this problem, we will replace the matrix $\mathbf{A}$ with the matrix:

$$\mathbf{M} = (1 - m)\mathbf{A} + m\mathbf{S},$$

  where $\mathbf{S}$ is $n \times n$ matrix with all entries 1/n. This means we add 'weak' links form every webpage to every other. The value of m originally used by Google is reportedly 0.15. For any $m \in [0, 1]$ the matrix $\mathbf{M}$ is column-stochastic and we can show that $V_1(\mathbf{M})$ is always one-dimensional if $m \in (0, 1]$ if there is no dangling nodes.

- The equation $\mathbf{x} = \mathbf{Mx}$ can also be cast as

$$\mathbf{x} = (1 - m)\mathbf{Ax} + m\mathbf{s},$$

  where $\mathbf{s}$ is a column vector with all entries 1/n. Note that $\mathbf{Sx} = \mathbf{s}$ if $\sum_{i=1}^{n} x_i = 1$, where $\mathbf{x} = (x_1, \ldots, x_n)^T$.