In Section (viii): PageRank Algorithm

We are now ready to explore an algorithm that you, yourself, have probably already used today (as well as perhaps a billion other people - estimates are that there are around 3.5 billion PageRank queries run per day)! The PageRank algorithm is the engine behind Google Search, and today it commonly ranks as one of the most influential & important algorithms in the field of data science (source: Springer).

Put simply, PageRank is an algorithm used to rank websites in search engine results, by applying a numerical weighting/measure to a semantically-relevant subgraph of the web.

As we shall see, this ranking procedure bears a strong resemblance to several of the centrality computations - particularly Katz centrality - that we have encountered.

The key difference between PageRank and several of these other centrality measures concerns what might be called "prestige dilution." Recall that with Katz centrality, nodes/webpages of high prestige or importance tend to freely propagate this prestige to their neighbors, and so on with their neighbors' neighbors. Depending on the particular real-world network of

interest. This "dilution" phenomenon might not be malum in se. However, with respect to ranking the importance of linked documents/webpages, such centrality dilution is potentially perilous. For instance, the wikipedia webpage should, naturally, carry a high centrality measure (and indeed it does according to PageRank's criteria, as any empirical Google search will attest). But suppose that wikipedia links directly a great many ancillary webpages (that is, webpages with an intrinsically small centrality (importance) - indeed it does link to many of these sorts of webpages. If one used Katz centrality to rank pages in this network, then the prestige of wikipedia would spread to a multitude of less important pages. Certainly this kind of dilution is generally undesirable for webpage rankings. PageRank proposes a simple solution.

We define centrality now as a variation of Katz centrality in which the centrality a vertex derives from its neighbors is proportional their centrality divided by their out-degree. (Note that our graph is a digraph here).

Mathematically, then, we define the centrality of node x_i as:

$$x_i = \alpha \sum_j A_{ij} \left(\frac{x_j}{k_j^{out}} \right) + \beta$$

where, in the previous formula, A_{ij} are elements of the adjacency matrix for the network, α is an "attenuation factor" (see below), and β is a "free" parameter essentially enabling vertices with zero in-degree to collect centrality - as seen with Katz centrality measures; k_j^{out} denotes the out-degree of vertex j in the graph.

Put succinctly, this variant of "PageRank centrality" (note that the algorithm/computation presented here represents a simplified version of the true PageRank algorithm - some of the finer details are, of course, well-kept industry trade secrets), is basically a "normalized" version of Katz centrality.

Looking at the formula on the previous page, the astute reader should question: what happens if $k_j^{out} = 0$? In this case our formula contains the indeterminate expression $\left(\frac{0}{0}\right)$, which we would like to avoid. We eschew this problem by artificially setting $k_j^{out} = 1$ for all such vertices (note that this "inflationary" term does not affect the exact solution / iterative convergence of the PR algorithm).

As before, we prefer to write the previous formula in matrix form:

$$\vec{x} = \alpha A D^{-1} \vec{x} + \beta \vec{1}$$

, with α & β as defined earlier; the matrix D here is the Digraph Matrix defined as $D_{ii} = \max\left(1, k_i^{out}\right)$.

Where does D^{-1} come from? Recall that for a diagonal matrix,

$$D = \begin{bmatrix} a_{11} & & \phi \\ & a_{22} & \\ \phi & & \ddots \\ & & & a_{nn} \end{bmatrix}, \quad D^{-1} \text{ is simply the new diagonal matrix consisting of the reciprocal values of } D.$$

Thus $D^{-1} = \begin{bmatrix} \frac{1}{a_{11}} & & \phi \\ & \frac{1}{a_{22}} & \\ \phi & & \ddots \\ & & & \frac{1}{a_{nn}} \end{bmatrix}$, noting that $a_{ii} \neq 0$ for all $1 \leq i \leq n$.

Recalling the formula: $\vec{x} = \alpha A D^{-1} \vec{x} + \beta \vec{1}$, we can now see that the left-term $(\alpha A D^{-1} \vec{x})$ propagates centrality from a vertex to its neighbors, "normalized" by vertex-out-degree & scaled additionally by the attenuation factor α. The right-term $(\beta \vec{1})$ effectively scales this centrality to avoid penalizing the zero in-degree vertices.

What is α? Google Search sets $\boxed{\alpha = .85}$ for PageRank computations. In one sense (recall convergence of the eigenvector centrality measures) this damping/attenuation < 1 guarantees convergence of the iterated calculations of PR measures. Secondly, we can think of $\alpha = .85$ as an "empirically-discovered" (i.e. it "works") hyperparameter value, reflecting the fact that we believe in general that there is an $1 - \alpha = \boxed{15\%}$ chance a given user won't follow any of the ranked pages suggested by Google Search.

As is typical, we set $\boxed{\beta = 1}$ & solve for $\vec{x}$:

$$\vec{x} = \alpha A D^{-1} \vec{x} + \vec{I}$$

$(-\alpha A D^{-1}\vec{x})$ $\quad$ $(-\alpha A D^{-1}\vec{x})$

$$\rightarrow \quad \vec{x} - \alpha A D^{-1} \vec{x} = \vec{I}$$

$$\rightarrow \quad (I - \alpha A D^{-1})\vec{x} = \vec{I}$$

($I - \alpha A D^{-1}$: invertible)

$$\rightarrow \quad (I - \alpha A D^{-1})^{-1} \cdot (I - \alpha A D^{-1})\vec{x} = (I - \alpha A D^{-1})^{-1} \vec{I}$$

$(I - \alpha A D^{-1})^{-1} \cdot (I - \alpha A D^{-1}) = I$

$$\rightarrow \quad \vec{x} = (I - \alpha A D^{-1})^{-1} \vec{I}$$

Notice the two instances of inverse operations. This is computationally inefficient, so we clean up.

$$\rightarrow \quad \vec{x} = \left((D - \alpha A) D^{-1}\right)^{-1} \vec{I}$$

(Recall: $(AB)^{-1} = B^{-1} A^{-1}$)

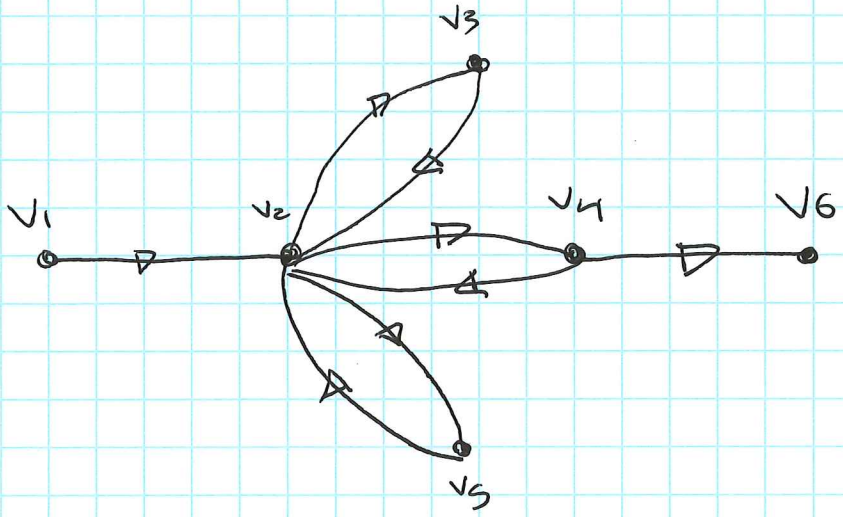
pull out D^{-1}

$$\rightarrow \quad \boxed{\vec{x} = D (D - \alpha A)^{-1} \vec{I}}$$

key formula!

PageRank works well for web-ranking for the reason that having links to your page from important pages elsewhere is a strong indication that your page may also be important. The added ingredient of normalizing centrality by out-degrees of pages that merely point to a great many pages do not freely bestow their centrality to a multitude of other pages.

Ex. We compute the PageRank score for each vertex in the following network.



$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Recall: $D_{ii} = \max(1, k_i^{out})$

$$A = \begin{array}{c} \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{matrix} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

Adjacency Matrix

We compute the solution to the PR algorithm system:

$\vec{x} = D(D - \alpha A)^{-1}\vec{1}$, with $\boxed{\alpha = .85}$. This yields:

$$\vec{v} = \begin{bmatrix} 3.706 \\ 11.23 \\ 3.706 \\ 4.88 \\ 3.706 \\ 2.763 \end{bmatrix}.$$

Consequently, the ranking (from high to low) of vertices by the PR algorithm is as follows:

$\{v_2, v_4, v_1, v_3, v_5, v_6\}$.