Project 3 NLA: PageRank Implementations

Iris Vukovic

December 21, 2024

A brief reflection with some theoretical comments. There are some implementation details within the notebook itself because in some cases I felt that it made more chronological sense to add explanations among the code blocks.

The PageRank algorithm estimates the importance of a website within a network of websites. Importance is measured by the number of webpages linking to a certain webpage, but also by how important the webpage that is linking to another webpage is. The values in the PageRank vector add up to one and each value is a fraction of the total score that the associated page earned. Each page is assigned a ranking based on these scores, a higher score representing a more important page.

We use the power method to compute the PageRank vector. The power method is an iterative algorithm that solves an eigenvalue problem. PageRank is an eigenproblem where the corresponding eigenvalue is 1 and the eigenvector is in fact the PageRank vector. The matrix A is know as the transition matrix and is the dot product of the link matrix G, which is composed of ones and zeroes based on if there is a link between two pages or not, and the diagonal matrix D, which contains the total links going out of each site on its diagonal.

When there are no outgoing links linking groups of connected networks, the issue of disconnected networks arises. Each separate cluster has its own independent PR vector because not all the websites in the network are taken into account when calculating rankings, only the websites within the connected clusters. Thus, the PR vector is not only inaccurate to the whole network, but is also not unique. Dangling nodes are websites with no outgoing links which are problematic because they can cause the algorithm to get stuck on one site that leads nowhere. Thus, we introduce the damping factor which simulates what is sometimes referred to as teleportation. This represents how often we will switch to a page at random, which ensures that all the websites will be accessed at some point, thereby evenly distributing the PageRank vector.

The transition matrix A represents the probability of moving from one page to another. If $M_m = (1-m)A + mS$ is our modified transition matrix constructed to handle disconnected networks, mS ensures that all isolated networks still have a nonzero ranking and are treated as part of the network. The link matrix can get very dense with zeroes representing the sites that don't have any links to them from other sites. We make the link matrix G sparse in

exercise 1 and we don't store matrices in exercise 2 both with the intention of reducing the memory needed to store the matrices.

I was also curious why we used the infinity norm specifically to check for convergence. Infinity norm is the max absolute value of all of a vector's components. In PageRank algorithm, we are interested in when the largest change becomes small enough to count as convergence and the infinity norm of the difference vector clearly tells us if iterations are approaching convergence or not.