18. Further Modifications

The algorithm we just described gives us a simple way of ranking the pages on the internet, but it has various issues that could be improved. Note that we didn't consider self-links in our count, which is why the diagonal of our adjacency matrix was zero. But what if an unscrupulous programmer from Wikipedia heads over to Buzzfeed and posts Wikipedia links everywhere? It's not hard to check and see that this causes the rank of Wikipedia to rise drastically. We want a system that is robust against a single site copying the same link over and over. A simple way to do this is to normalize the values in each column to add to one. This gets rid of the ability of any one website to largely skew the ranking. Also, this has probabilistic implications, but those are beyond the level of this class (for more information look into Markov Processes).

If we do this we are left with the following matrix

$$\mathbf{S} = \begin{pmatrix} 0 & \frac{2}{3} & \frac{2}{5} \\ \frac{1}{2} & 0 & \frac{3}{5} \\ \frac{1}{2} & \frac{1}{3} & 0 \end{pmatrix}.$$

Problem 18.1 Find the largest absolute-value eigenvalue and corresponding eigenvector of the above matrix.

Solution

We follow the steps we went through previously, first finding the eigenvalues by using

$$0 = \det(\mathbf{S} - \lambda \mathbf{I})$$

And when we plug in numbers, getting

$$0 = \det \begin{pmatrix} -\lambda & \frac{2}{3} & \frac{2}{5} \\ \frac{1}{2} & -\lambda & \frac{3}{5} \\ \frac{1}{2} & \frac{1}{3} & -\lambda \end{pmatrix}$$

And writing out the determinant gives

$$0 = -\lambda^3 + \frac{1}{5} + \frac{1}{15} + \frac{1}{5}\lambda + \frac{1}{3}\lambda + \frac{1}{5}\lambda.$$

Simplifying we see that

$$0 = \lambda^3 - \frac{11}{15}\lambda - \frac{4}{15}.$$

Note that immediately we can see that $\lambda_1 = 1$ is an eigenvalue, and we can rewrite the equation as

$$0 = (\lambda - 1)(\lambda^2 + \lambda + \frac{4}{15}\lambda)$$

We note that the second term has no real roots, so we only have one real eigenvalue. To now find the eigenvector, we need to find a vector in the nullspace of

$$\begin{pmatrix} -1 & \frac{2}{3} & \frac{2}{5} \\ \frac{1}{2} & -1 & \frac{3}{5} \\ \frac{1}{2} & \frac{1}{3} & -1 \end{pmatrix}.$$

We now proceed with Gauss-Jordan elimination, first eliminating the 2nd and 3rd entries in the first column:

$$\begin{pmatrix} 1 & -\frac{2}{3} & -\frac{2}{5} \\ 0 & -\frac{2}{3} & \frac{4}{5} \\ 0 & \frac{2}{3} & -\frac{4}{5} \end{pmatrix}.$$

Now using the second row to eliminate the first nonzero term in the third, along with zeroing out the second term in the first row, we get

$$\begin{pmatrix} 1 & 0 & -\frac{6}{5} \\ 0 & 1 & -\frac{6}{5} \\ 0 & 0 & 0 \end{pmatrix}.$$

We can immediately see that the vector

$$\mathbf{v} = \begin{pmatrix} 1 \\ 1 \\ \frac{5}{6} \end{pmatrix}$$

is an eigenvector of $\mathbf{S}$. Note that because we normalized by the number of links on each page, the many additional links on Buzzfeed make a much smaller contribution, and the rankings of Wikipedia and Facebook are much closer to that of Buzzfeed.

Hide

Meaning of the algorithm

It might have seemed interesting to you that one of the eigenvalues for the matrix was one. This is not a coincidence. When we normalized the matrix so that each column added to one, we were essentially giving the likelihood that someone on a given webpage would then go to any other webpage. If we imagine a large number of people surfing the internet, then at any given time they will each be occupying a particular page, and the people will migrate to other pages proportional to the number of links from their current page to their new page. Eventually in some sense we might expect a steady state to arise, where the number of people on any given page is constant, but people are constantly coming and going. If we have a vector represent the number of people at each website at one time, then multiplying it by our matrix corresponds to the number of people at each website after everyone clicks one link. In this case, the steady state corresponds to an eigenvector of people on each page that has an eigenvalue of one, meaning the distribution of people is the same after everyone clicks a link. Thus the page rank algorithm gives us the distribution of people on the internet if everyone is randomly surfing around, and captures the fact that a popular page will tend to have more people hanging out on it.

Hide

18. Further Modifications

Hide Discussion

Topic: Unit 2: Linear Algebra, Part 2 / 18. Further Modifications

Add a Post

Show all posts ▼ by recent activity ▼

Monopoly...

Community TA

1

Learn About Verified Certificates