The Giving Game: Google Page Rank
Nick Korevaar

The Game

Imagine a game in which you repeatedly distribute something desirable to your friends, according to a fixed template. For example, maybe you’re giving away “play–doh” or pennies! (Or it could be you’re a web page, and you’re giving credibility to other web pages by linking to them. Or maybe, you’re a football team, and you’re voting for yourself, along with any teams that have beaten you.)

Let’s play a small–sized game. Maybe there are four friends in your group, and at each stage you split your material into equal sized lumps, and pass it along to your friends, according to the template at the right.

The question at the heart of the basic Google page rank algorithm is: in a voting game like this, with billions of linked web pages and some initial vote distribution, does the way the votes are distributed settle down in the limit? If so, sites with more limiting votes must ultimately be receiving a lot of votes, so must be considered important by a lot of sites, or at least by sites which themselves are receiving a lot of votes. Let’s play!

Exercise 1. Decide on your initial material allocations. I recommend giving it all to one person at the start, even though that doesn’t seem fair. If you’re using pennies, 33 is a nice number for this template. At each stage, split your current amount into equal portions and distribute it to your friends, according to the template above. If you have remainder pennies, distribute them randomly. Play the game many (20?) times, and see what ultimately happens to the amounts of material each person controls. Compare results from different groups, with different initial allocations.

Modeling the game algebraically

The game we just played is an example of a discrete dynamical system, with constant transition matrix. Let the initial fraction of play dough distributed to the four players be given by

\[ x_0 = \begin{bmatrix} x_{0,1} \\ x_{0,2} \\ x_{0,3} \\ x_{0,4} \end{bmatrix}, \quad \sum_{i=1}^{4} x_{0,i} = 1 \]

Then for the game template above, we get the fractions at later stages by

\[
\begin{bmatrix} x_{k+1,1} \\ x_{k+1,2} \\ x_{k+1,3} \\ x_{k+1,4} \end{bmatrix} = x_{k,1} \begin{bmatrix} 0 \\ 0.5 \\ 0.5 \\ 0 \end{bmatrix} + x_{k,2} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + x_{k,3} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + x_{k,4} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0.5 \end{bmatrix}
\]

So in matrix form, \( x_k = A^k x_0 \) for the transition matrix \( A \) given above.
Exercise 2. Compute a large power of $A$ (the matrix at the bottom of the previous page). What do you notice, and how is this related to the experiment of exercise 1?

Exercise 3. Not all giving games have happy endings. What happens for the following templates? Encode each template as a matrix and compute a high power for each.

Here’s what separates good giving-game templates, like the page 1 example, from the bad examples above.

Definition. A square matrix $S$ is called **stochastic** if all its entries are positive, and the entries in each column add up to exactly one.

Definition. A square matrix $A$ is **almost stochastic** if all its entries are non-negative, the entries in each column add up to one, and if there is a positive power $k$ so that $A^k$ is stochastic.

Theoretical basis for Google page rank

**Theorem.** (Perron–Frobenius) Let $A$ be almost stochastic. Let $x_0$ be any “fraction vector” i.e. all its entries are non-negative and their sum is one. Then the discrete dynamical system

$$x_k = A^k x_0$$

has a unique limiting fraction vector $z$, and each entry of $z$ is positive. Furthermore, the matrix powers $A^k$ converge to a limit matrix, each of whose columns are equal to $z$.

The Google fudge factor

Sergey Brin and Larry Page realized that the world wide web is not almost stochastic. However, in addition to realizing that the Perron–Frobenius theorem was potentially useful for ranking URLs, they figured out a simple way to guarantee stochasticity—the “Google fudge factor.”

Rather than using the voting matrix $A$ described previously, they take a combination of $A$ with a matrix of all 1s which we will denote $\mathbf{1}$. (Note that $\mathbf{1}$ is not the identity matrix!) For $\varepsilon$, a small number and $n$ equal to the number of nodes/pages, consider the Google matrix:

$$G = (1 - \varepsilon)A + \frac{\varepsilon}{n} \mathbf{1}.$$
If $A$ is almost stochastic, then each column of $G$ also sums to 1 and each entry is at least $\varepsilon/n$. This $G$ is stochastic! In other words, if you use this transition matrix everyone gets a piece of your play-doh, but you still get to give more to your friends.

**Exercise 4.** Consider the third giving game from exercise 3. Its transition matrix:

$$
A = \begin{bmatrix}
0 & 0 & 0 & 0.5 & 0 & 0 \\
0.5 & 0 & 0 & 0 & 0 & 0 \\
0.5 & 1 & 0 & 0.5 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 \\
\end{bmatrix}
$$

is not almost stochastic. For $n = 6$ and $\varepsilon = .3$, work out the Google matrix $G$, along with the limit rankings for the six pages. If you were upset that page 4 was ranked as equal to page 3 in the game you played in exercise 1, you may be happier now.

**Exercise 5.** Here’s a more complicated “bad” game. Try to figure out what happens with this template by inspection. Next, encode it into a matrix and compute a high power of it, to check if you were right. Finally, create the Google matrix $G$ with $n = 8$ and $\varepsilon = .15$, and determine a limit ranking for each page.