How Google’s Algorithm Picks the Best Pages for You

Michael Godzik, Adam Sabol

Definitions

Markov Chain: A matrix which models the movement between states, more formally defined as steps or transitions

State Vector: A vector defined as x_n which is a vector of the probabilities that the chain exists in a certain state after n steps

Initial State Vector: The state vector defined as x_0 that describes the system as start

Random Walk: A chain of states (integer values greater than 0) where each state only has a possible transition to a state directly adjacent to it

Absorbing Boundary: When the state vector arrives at state 1 or n it remains there. A matrix with absorbing boundaries is defined below. Looking at the first column we see that the probability of a transition 1:1 is 1 and 0 for all else. Similarly, looking at the fourth column we see that the transition 4:4 is 1 and 0 for all else.

$$\begin{bmatrix} 1 & p & 0 & 0 \\ 0 & 0 & p & 0 \\ 0 & 1-p & 0 & 0 \\ 0 & 0 & 1-p & 1 \end{bmatrix}$$

Reflexive Boundary: When the state vector reaches a boundary it immediately returns to the preceding state in the next step. A matrix with reflexive boundaries is defined below. Notice that the transitions from state one are always to state two and the transitions from four are always to state three.

$$\begin{bmatrix} 0 & p & 0 & 0 \\ 1 & 0 & p & 0 \\ 0 & 1-p & 0 & 1 \\ 0 & 0 & 1-p & 0 \end{bmatrix}$$

Steady State/Equilibrium Vector: A vector such that Pq = q where P is a transition matrix and q is a probability vector.

Occupation Time: The proportion of time steps that the chain spends in each state

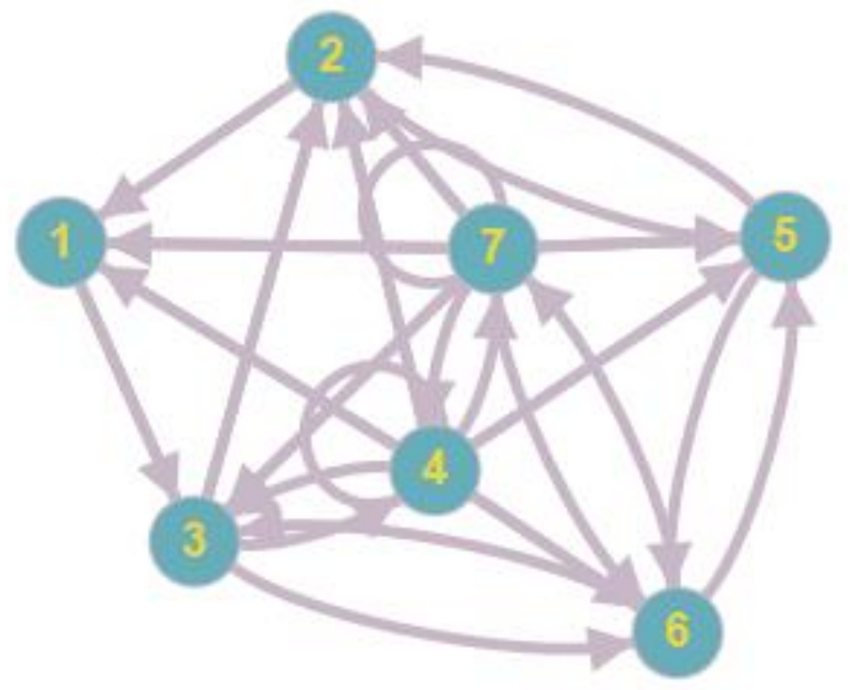
Background

Google’s PageRank algorithm was created with goal of interpreting a sites effectiveness based on its view time. The founders at Google reasoned that “important” pages would have more viewers spending more time than less important pages. But we can only call the amount of time spent at each page as the occupation time for each state in a Markov Chain. In order to understand how Google’s algorithm works, we must first become comfortable with the concept of Markov Chains. A Markov Chain is a matrix which models the movement between states, more formally defined as steps or transitions. These transitions are placed in a transition matrix where the ijth entry models the transition from state j to state i. A state vector x_n is a vector of the probabilities that the chain is in any state after n steps.

Computing the steady-state vector for Google’s PageRank is not a simple task. While modeling the google algorithm seems simple on the smaller scale, only taking in a couple pages at once, the computation becomes much more difficult as you account for all of Google’s information. The completed Google Matrix has well over 8 billion rows and columns. The following matrix helps display what the PageRank algorithm does on a smaller scale of 7 pages.

$$G = \begin{bmatrix} 0 & 1/2 & 0 & 1/7 & 0 & 0 & 1/7 \\ 0 & 0 & 1/3 & 1/7 & 1/2 & 0 & 1/7 \\ 1 & 0 & 0 & 1/7 & 0 & 1/3 & 1/7 \\ 0 & 0 & 1/3 & 1/7 & 0 & 0 & 1/7 \\ 0 & 1/2 & 0 & 1/7 & 0 & 1/3 & 1/7 \\ 0 & 0 & 1/3 & 1/7 & 1/2 & 0 & 1/7 \\ 0 & 0 & 0 & 1/7 & 0 & 1/3 & 1/7 \end{bmatrix}$$

Perhaps a better representation for the matrix could be created using graph theory, giving us the following image



Note that the arrows signify the state in which a given state can move towards. In this representation It is much easier to visualize where you can move.

Background Continued

In order to estimate the eigenvalues and the eigenvectors, the power method must be used. Since the steady-state vector of the seven-page example of the Google Matrix is an eigenvector, we can find that at least 50 iterations are needed to reach a level of accuracy that Google approves. Since new pages are always being created and different resources are becoming more popular over time, it takes several days for a new steady-state to be computed.

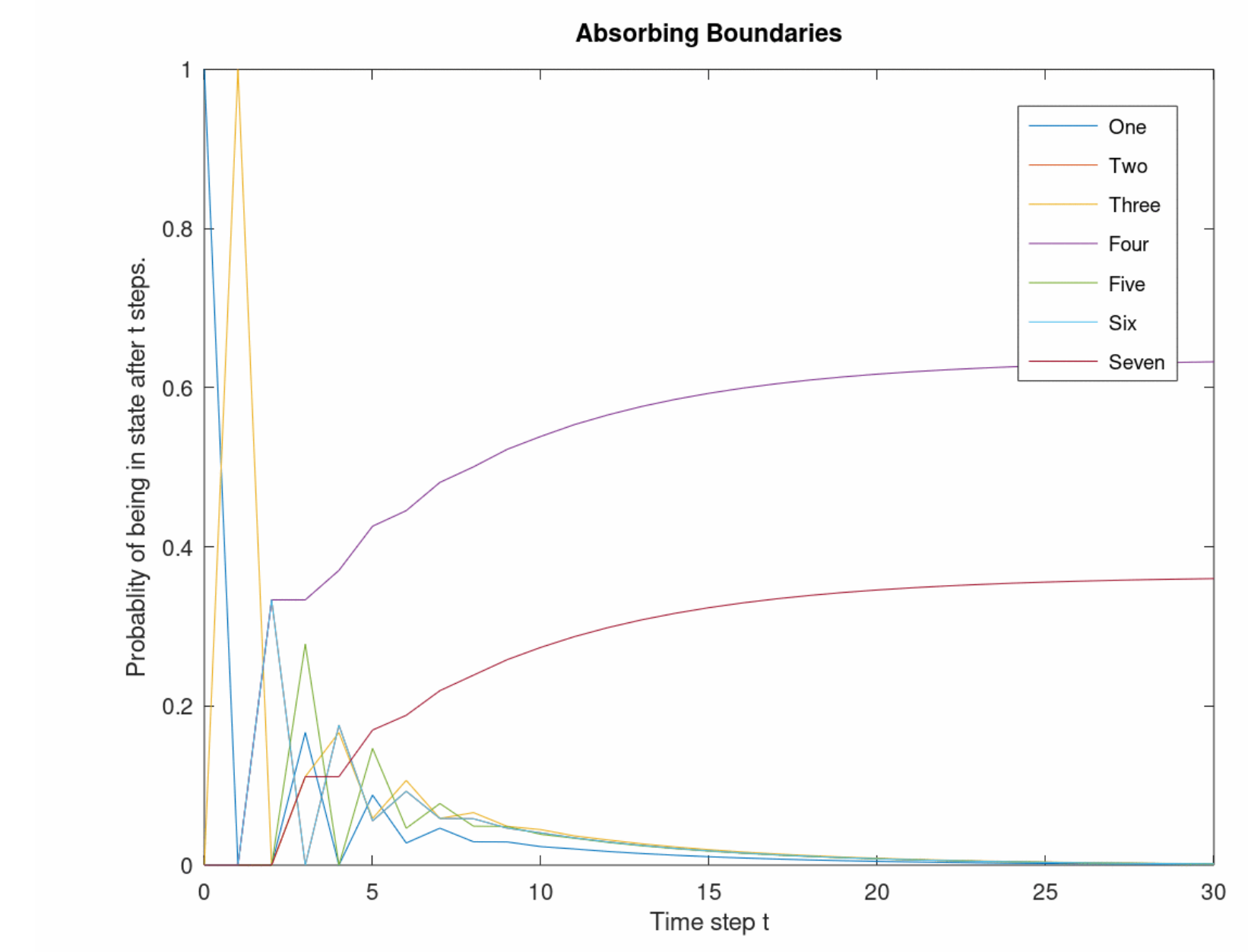
The World Wide Web can be modeled using graph theory with vertices representing webpages and edges representing the links that lead between them similarly to how we visualized the seven-page example. Google utilizes Markov chains to rank pages based on time spent on each page and tries to create a list of the most important pages for your search.

Graphical Analysis

The following matrices represent the long-term behavior of Markov chains when analyzing Google’s PageRank algorithm. The first matrix and graph depict an absorbing boundary, the second matrix depicts a circulatory or reflexive boundary, and the final matrix depicts a chain in which there is no absorbing or reflexive boundaries. The probability of landing on a page is shown by the y values over time, with time being the number of pages visited.

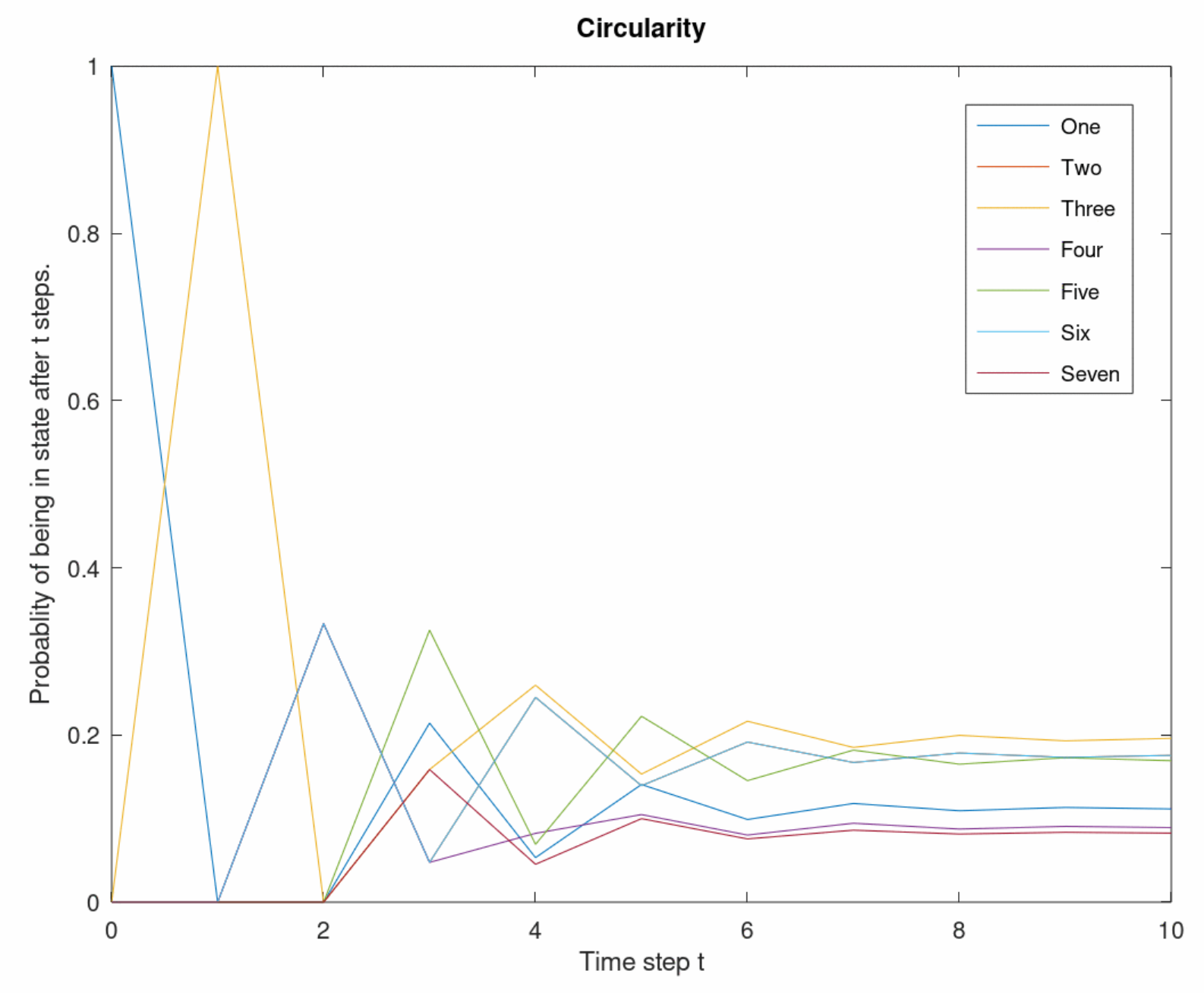
Matrix 1: Absorbing Boundaries

$$P = \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{bmatrix} 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 1/2 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 1/3 & 1 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 1/3 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/3 & 1 \end{bmatrix} \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array}$$



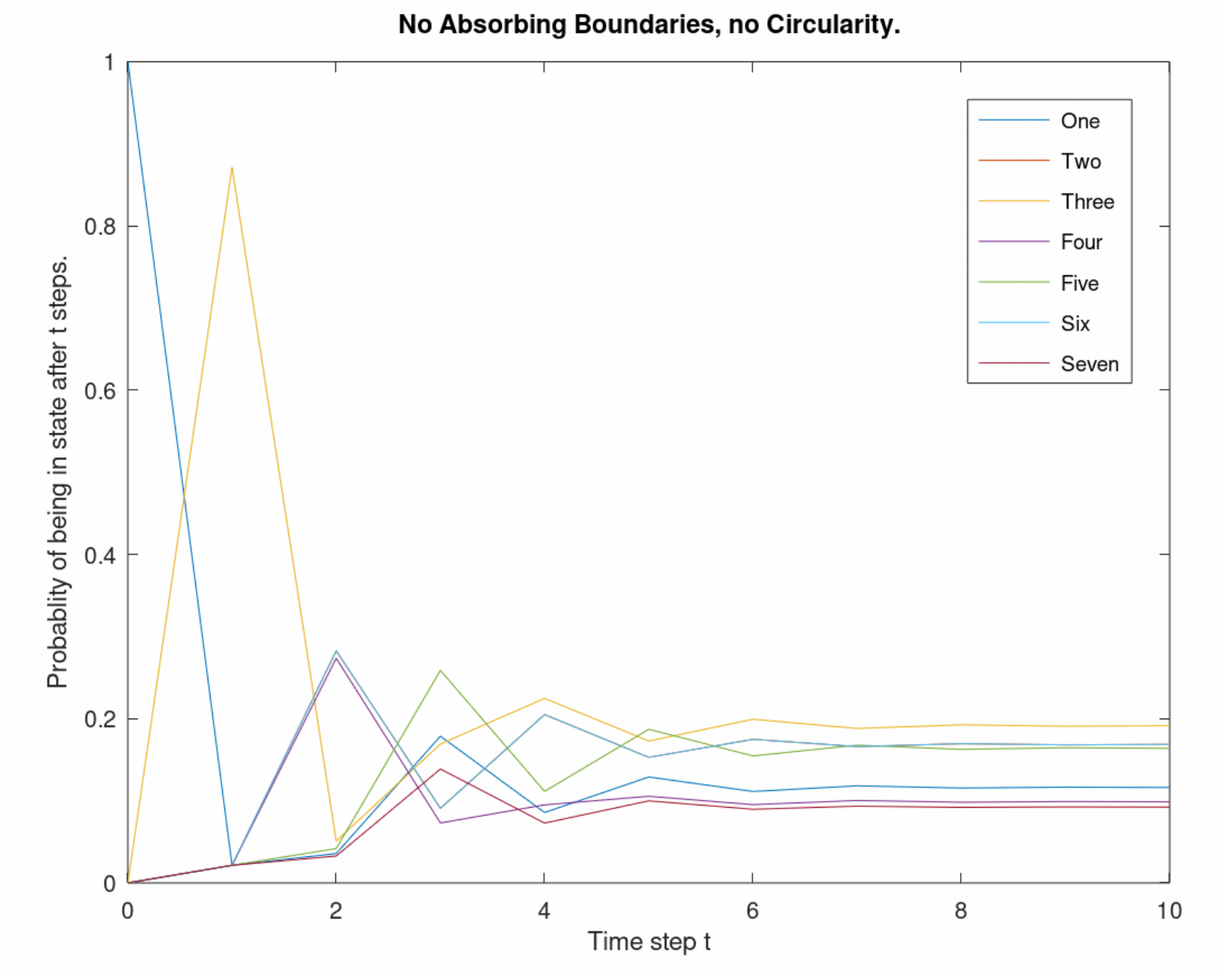
Matrix 2: Circulatory Boundaries

$$P_* = \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{bmatrix} 0 & 1/2 & 0 & 1/7 & 0 & 0 & 1/7 \\ 0 & 0 & 1/3 & 1/7 & 1/2 & 0 & 1/7 \\ 1 & 0 & 0 & 1/7 & 0 & 1/3 & 1/7 \\ 0 & 0 & 1/3 & 1/7 & 0 & 0 & 1/7 \\ 0 & 1/2 & 0 & 1/7 & 0 & 1/3 & 1/7 \\ 0 & 0 & 1/3 & 1/7 & 1/2 & 0 & 1/7 \\ 0 & 0 & 0 & 1/7 & 0 & 1/3 & 1/7 \end{bmatrix} \end{array}$$



Matrix 3: No Absorbing or Circulatory Boundaries

$$\begin{bmatrix} .021429 & .446429 & .021429 & .142857 & .021429 & .021429 & .142857 \\ .021429 & .021429 & .304762 & .142857 & .446429 & .021429 & .142857 \\ .871429 & .021429 & .021429 & .142857 & .021429 & .304762 & .142857 \\ .021429 & .021429 & .304762 & .142857 & .021429 & .021429 & .142857 \\ .021429 & .446429 & .021429 & .142857 & .021429 & .304762 & .142857 \\ .021429 & .021429 & .304762 & .142857 & .446429 & .021429 & .142857 \\ .021429 & .021429 & .021429 & .142857 & .021429 & .304762 & .142857 \end{bmatrix}$$



Examples

Example 1: Consider a Markov Chain on {1,2} with the given transition matrix P. Use two methods to find the probability that, in the long run, the chain is in state 1.

$$P = \begin{bmatrix} .2 & .4 \\ .8 & .6 \end{bmatrix}$$

Method 1. Find a vector q such that Pq = q

$$P = \begin{bmatrix} .2 & .4 \\ .8 & .6 \end{bmatrix} P^{100} = \begin{bmatrix} .333 & .667 \\ .333 & .6667 \end{bmatrix} q = \begin{bmatrix} .333 \\ .667 \end{bmatrix}$$

Method 2.

$$\lambda P = q$$

$$\lambda = 1$$

$$q = -\frac{1}{1.341} \begin{bmatrix} -.44721 \\ -.89443 \end{bmatrix} = \begin{bmatrix} .333 \\ .667 \end{bmatrix}$$

Example 2. Consider an unbiased random walk with reflecting boundaries on {1,2,3,4}.

a. Find the transition matrix for the Markov chain and show that this matrix is not regular. By following the definition of a matrix with reflexive boundaries, we find that

$$P = \begin{bmatrix} 0 & 1/2 & 0 & 0 \\ 1 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1 \\ 0 & 0 & 1/2 & 0 \end{bmatrix}$$

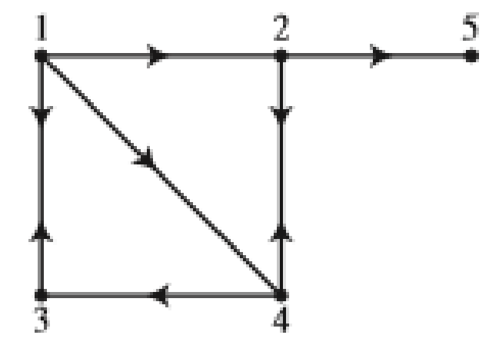
We know that this matrix is not regular because odd and even integers for P^k give alternating values at the edges.

b. Assuming that the steady-state vector may be interpreted as occupation times for this Markov chain, in what state will this chain spend the most steps?
Using the equation pq = q we can find the steady state vector as followed

$$P^{50} = \begin{bmatrix} 0 & 1/2 & 0 & 0 \\ 1 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1 \\ 0 & 0 & 1/2 & 0 \end{bmatrix}^{50} = \begin{bmatrix} 1/6 \\ 1/3 \\ 1/3 \\ 1/6 \end{bmatrix}$$

By the result of this matrix, we know that it will spend the most time at state 2 & 3 which we should expect since the matrix is reflexive.

Example 3. Consider a set of webpages hyperlinked by the given directed graph. Find the Google matrix and compute the PageRank matrix.



Adjacency Matrix:

$$P = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

PageRank Matrix:

$$S_{ij} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1/3 & 0 & 0 & 1/2 & 0 \\ 1/3 & 0 & 0 & 1/2 & 0 \\ 1/3 & 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 \end{bmatrix}$$

[G_{ij}] = p*S_{ij} + (1-p) (1/N)
Google uses a known value of p around 0.85 and there are N = 5 terms so

[G_{ij}] = 0.85*S_{ij} + 0.15 *(1/5)

References

Lay, David C., et al. *Linear Algebra and Its Applications*. Pearson, 2016.