Final Project : Google's PageRank

MAT 167 001 / University of California, Davis / Summer Session 1 2024 / DP Varn, Instructor

20 July 2024

Introduction: For this summer session, we'll all work on Google's PageRank algorithm. I'll provide you with a small network of webpages to analyze, and you'll rank the webpages that are the most important in the network. You will use your results to determine a ranking for the most useful webpages given certain term searches.

References:

1. Please read over Moler's Chapter 2, from pages 23 - 29.
2. Please look over Prof. Saito's Lecture #25.

I'll draw heavily from each of these sources, and you'll benefit greatly from looking over each of these sources in some detail.

Input Materials: Each group will be provided, by me, with these materials to get started:

1. A hyperlink network of 12 nodes.
2. A list terms (keywords) that appear on each webpage.
3. A set of user generated queries to which you should answer the question: Which webpages are responsive to each query, and in which order should the webpages be presented to the user?

Suggested Procedure:

1. Turn your hyperlink graph into a "raw" Google Matrix, G. You should check to see if all of the rows either add to 1 or 0. If not, then you have made a mistake.
2. Do an eigenvalue decomposition of your raw Google Matrix. You should solve the equation

$$\boldsymbol{\pi}^\intercal = \boldsymbol{\pi}^\intercal G$$

   where $\boldsymbol{\pi}^\intercal$ is a *row* vector whose entries measure the importance of the the webpage that you are considering, and G is the google matrix. The eigenvalue here is explicitly $\lambda = 1$. You can use a more familiar eigenvalue decomposition by taking the transpose of the equation, ie, $G^\intercal \boldsymbol{\pi} = \boldsymbol{\pi}$.
3. Did you get an eigenvalue of 1? If not, choose the eigenvector with the largest eigenvalue. Does this eigenvector have all non-negative entries?
4. We really need to fix the original Google matrix to solve this problem. To do this, we have to fix any *dangling* webpages. That is, if we have webpages that don't have any outlinks, we artificially add hyperlinks leaving this page that go to every webpage in the network, including itself. Do this. Now your G matrix should have rows that all sum to one.
5. Now solve $G^\intercal \boldsymbol{\pi} = \boldsymbol{\pi}$. Do you get an eigenvalue of 1 now? Are all the entries in the corresponding eigenvector non-negative?
6. It is customary to normalize the eigenvector so that $\|\boldsymbol{\pi}\|_1 = 1$. This is sometimes written as $\boldsymbol{\pi}^\intercal \mathbb{1}_n = 1$ where $\mathbb{1}_n \in \mathbb{R}^{n\times 1}$ is a column vector of all 1's. Do this.
7. The entries in the $\boldsymbol{\pi}$ vector give the importance of the corresponding webpage. Order the entries in your $\boldsymbol{\pi}$ vector so that you sort the relative importance of each webpage in non-increasing order.
8. If G is a large matrix, it is difficult to solve the eigenvalue problem directly. As an alternative, one can instead use the *power method* (see, eg, Saito's Lecture #24) to find the eigenvector associated with the largest eigenvalue. Use the power method to solve for the eigenvector associated with the largest (aka *dominant*) eigenvalue. Specifically we can say

$$\boldsymbol{\pi}_j^\intercal = \boldsymbol{\pi}_{j-1}^\intercal G$$

   where $\boldsymbol{\pi}_j$ is the j^{th} iteration of $\boldsymbol{\pi}$, and one repeats the process until the $\|\boldsymbol{\pi}_j - \boldsymbol{\pi}_{j-1}\| \leq \epsilon$, where ϵ is some preset tolerance. It doesn't really matter what you choose as $\boldsymbol{\pi}_0$, as long as $\boldsymbol{\pi}_0$ is normalized. A common choice is $\boldsymbol{\pi}_0 = \frac{1}{n}\mathbb{1}_n$. (Note that this choice gives $\|\boldsymbol{\pi}_0\|_1 = 1$.) How many iterations do you need to use? Do you get the same answer as before?

9. Lastly, although your hyperlink network does not have this issue, it is possible that one might have a network that is not *strongly connected*. This can be fixed by the following adjustment to G:

$$\tilde{G} = \alpha G + (1 - \alpha)E$$

where $\alpha \in [0, 1]$ and $E := \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^\intercal$. This is equivalent to a *teleportation*, where a user may just type in a webpage instead of following a hyperlink. Usually one selects $\alpha = 0.85$. Construct $\tilde{G}$ for your network.
10. Now perform a eigendecomposition of $\tilde{G}$ for your network using whatever method you think is best. That is, solve $\tilde{G}^\intercal\boldsymbol{\pi} = \boldsymbol{\pi}$.
11. Compare your results for the rankings of the webpages from G and $\tilde{G}$. Are the rankings the same?
12. Your group will have been provided a list of keywords associated with each webpage. Use this list to construct a term-document matrix for your network. Call this matrix T.
13. Further, your group will have been provided with a user query consisting of a few keywords. From this user query, create a query vector. Call this query vector $\mathbf{q}$.
14. Compute $\mathbf{d}^\intercal = \mathbf{q}^\intercal\, T$. From your $\mathbf{d}$ vector, determine which webpages are responsive to the query.
15. Now, use your webpage ranking to rank the responsive webpages in order of importance.

You'll see that as far as the programming is concerned, much of this can be done quite simply with relatively few commands in MatLab. While the Moler's Chapter 2 does give a good amount of code that you can use, don't feel bound to use it. You might find it easier (much easier!) to just see what needs to done and write the code yourself.

Suggestions:

1. Code is always easier to read when it has been commented. Please make amply use of comment statements in your code to let me see what you are doing.
2. For putting your project together, you might want to imagine that I'm in charge of hiring for a medium sized tech company, and I'm looking for ambitious, talented folks to join our development team. You might suppose that I don't know all that much about the Google PageRank algorithm, and you want to simultaneously tell me what it is and show how it works for a simple case.
3. As such, devote a few slides to explaining the mathematics behind PageRank, and the logic behind it. Some historical context is always nice.
4. Be sure to clearly state the problem that you are trying to solve. Then show how you solve it. Then tell me the results.
5. Nothing is more tedious in a talk than having someone walk through code, unless there are subtleties that need explaining. I'll have your actual code in a separate file, so don't put it in your presentation unless you are trying to make a specific point. Don't be tedious.
6. Presentations are a little like stories. They need an introduction where you tell me about the cast of characters. Then you tell me what you are going to do. Then you do it. Then you tell me what you found out. Then you sum it all up for me. For this latter statement, that means that at the end of your presentation you have a slide that overviews the most important concepts/results from the previous slides.
7. No man is an island, and no work comes completely from scratch. Feel free to use sources, but be absolutely sure to give your sources. Otherwise, you might run the risk of plagiarism. Often the difference between *research* and *plagiarism* is whether you adequately cited your sources. Never take credit for something that isn't yours. It is common to have a slide at the end that gives the sources that you used.
8. Everything on a slide should have an informational purpose. Putting cute graphics on a slide just fill a blank spot is amateurish, and makes you look unserious. Don't use fonts, graphics, garish colors or anything else that distracts from your message. However, carefully chosen graphical representations of data, or flow graphs, or perhaps a graph of the network that you are analyzing would be entirely appropriate. These convey information in a concise and digestible form, and your audience (ie yours truly) will appreciate that.
9. Punctuation, spelling, and organization really do matter. As does flow. Your presentation should have a logical flow to it, one that seems almost inevitable. I should never wonder what you are doing or where you are going.
10. Presentations should have title pages/slides. There should be a title with a date and who it's from and why it is being given. Often, presenters give their email addresses. You never want a prospective employer to wonder how to get in contact with you.
11. Number your slides! Please! It'll make it easier for me to refer to them in my comments. And, it's just good practice.
12. Although I've given you a numbered "roadmap" for what you should calculate for you project, these numbers should never appear in the actual presentation.

