Moiez Qamar
CS 100 Roadmap Project
12/07/2021

Search engines are software systems that help us make millions of web searches for information every day. Examples of common search engines we use include Yahoo, Bing, and of course, Google. While we know that search engines can give us billions of links and resources in just seconds, many of us don't look into (1) how search engines pick the best sites to display for their users (2) what algorithms are incorporated to have these search engines do their ultimate job.

Two concepts from linear algebra that are necessary to understand this topic are Markov chains and PageRank. By definition, Markov chains are a sequence of probability vectors together with a stochastic matrix. A probability vector is essentially non negative entries that add up to 1 and a matrix made up of such probability vectors is what creates a stochastic matrix. Learning the applications to Markov's Chains and expanding that to the Google PageRank concept has made me realize that this mathematical concept is what is used "behind the scenes" in the technology world. More specifically, computing PageRank and the formulas associated with it are what help search engines prioritize web pages to display in a single search (Varagouli, 2020).

So, what exactly is PageRank? When you search up something in Google, multiple links are provided. And each link that is listed is put in a certain order. PageRank essentially assigns a 'score' to each link that has relevance to what you are looking for. Putting it bluntly, the score is created by accessing the link and then going through every link on that page to find out how many links and sub-links are present. The higher the number of links, the higher relevance it has when the topic is searched for.

Taking Google's search engine as an example, while PageRank plays a major role in ranking web pages based on their rank scores, algorithms for information search are still needed. This is where technical concepts like recursion and Quicksort come in. In simple terms, Google's PageRank is a recursion intensive algorithm that delves deeper through the webpages and when it reaches the very last one, it begins returning values to create the actual page rank. Similarly, the Quicksort algorithm focuses on taking a set of data and doesn't look at it as a whole— rather it breaks down the data in halves until it has a base value it can return to build up the final sorted data set (NetworkX guide).

After doing some research, the actual mathematical formula for PageRank is: PR(A) = (1-d) + d (PR(T1)/C(T1) + ... + PR(Tn)/C(Tn)) (Collins, 2018). This continues until it finds the base value. Likewise for Quicksort, it follows a similar pattern but in a logarithmic manner. It uses a pivot point of A[q] and rearranges the data so the data is split into two subset, one that is less than A[q] : A[p,…,q−1] and the other which is greater than A[q]: A[q+1,…,r]. The same process is done repetitive till a base value can be returned to ultimately sort the data.

The complexity for Quicksort is O(logN * N). The complexity for PageRank is O(k * N) [k = no. of iterations]. These two algorithms' recursive nature make them have a similar complexity time but it varies in the number of iterations. Overall, I find technical and mathematical concepts coming together when you look into and research their overall purpose and complexity, or the resources (time and memory requirements) to make these algorithms work for search engines.

When you look at PageRank in a larger picture, you then can see its powerfulness and complexity. If you simply search "What is python?" on Google, you are hit with 445,000,000 results and the time it takes is 0.71 seconds. At first glance, that looks cool and simple but in those 0.71 seconds, Google uses its PageRank to scan each of those 445,000,000 results, assigns it a PageRank score and then organizes all the results based on the highest score to the lowest. Looking at the result of the first link that pops up when you search “what is python,” you get the link to Python.org’s executive summary for ‘What is Python?’ In a rough counting of the number of external links, I counted approximately 170 links excluding the links for Twitter Tweets (which numbered to over 100 themselves). Trying to go through one of these links was a massive effort due to continuous redirections. I would approximate that if I were to attempt to go through every single link through Python.org in order to find the base case, I would have to go through over 1.353395e+156 links. The fact that it took a huge amount of time for me to get through just 170 links, Google goes through this in a tiny fraction of .71 seconds, it takes Google actually 1.59550562e-9 seconds to simply go through every one of the 445,000,000 results from when you search up ‘what is python’. Overall, this topic of PageRank along with the sorting algorithm of QuickSort show the complexity behind how powerful these search engines are.

References:

https://www.python.org/doc/essays/blurb/

https://www.semrush.com/blog/pagerank/

https://networkx.guide/algorithms/link-analysis/pagerank/

https://www.searchenginewatch.com/2018/10/25/googles-pagerank-algorithm-explained/