# CSE 3504 Probabilistic Performance Analysis
# Project 2

Drew Monroe and Sailesh Simhadri

Tuesday 25th April, 2017

## Practice 1

### Question 3

Changing the damping factor does change the ranks, however, some of the pages remain the same. For example, the first page, node 2, and the last page, node 51, for a damping factor of 0.85 remain the first and last page for a damping factor of 0.95. However, for a damping factor of 0.85, the fourth ranked page is node 61, while it is ranked fifth for a damping factor of 0.85.

### Question 4

Switching the damping factor to 0.5 does change some of the ranks, but not all of them. Again, the first page is still node 2, and the last page is still 51. For a damping factor of 0.5, the fourth and fifth nodes are switched from a damping factor of 0.85.

## Practice 2

The algorithm that we decided to target is the "synonyms" algorithm. The goal of this algorithm is to search for words closely related to words contained in the search. For example, if a user searches for "humor", the algorithm would include results that would have the word "funny" or "commical" in them. We feel that a basic form of such an algorithm would work as follows:

1. Assume we have a hash map, $h$ that maps words to a list of synonyms, each synonym with a weight specifying how close of a synonym it is to the given word, and that we have a list of basic words, $b$, to ignore (for example "a", "the", "for", etc.)

2. For each word, $w$, in the search

   (a) If $w$ is in $b$, continue

    (b) Otherwise, put $w$ through the hashmap, returning all related synonyms to that word, and their relative weightings.

3. Now that we have a larger list of related words to search from, search through pages that have words related to these, and return results ranked upon the similarity of words in the page with words searched. This part of the algorithm will utilize the weights from $h$. Words that are more closely related as synonyms will have a higher rank than words less closely related.

However, this algorithm gets more interesting when you start to consider phrases that are similar. For example, if some one were to search “office supplies”, it would be nice if things like “staples”, “pens”, and “clipboards” were to show up as well. This is where Google can start using its own users to build up its synonym hashmap. Whenever someone searches a phrase, Google can record what links were clicked on, and use the generate the hashmap. Then, one sufficient data has been gathered, it can start to target phrases of words as opposed to individual words. This will generate better search results, because the phrase “water gun” has very different implications than the words “water” and “gun” taken individually.

Unlike the pagerank algorithm, which uses the links on pages to generate weights, this algorithm could use actual user-driven data. By initially starting to sprinkle in random results to users’ searches, Google can determine how successful their guess was based upon if a user clicks on a link. For example, assume that Google start with a primitive hashmap of individual words to a list of individual words, as described above. When a user searches for “office supplies”, Google breaks up the search by word. It looks for synonyms for the word “office”, and synonyms for the word “supplies”, and adds in the top results from these synonyms into the search. If a user clicks on one of these inserted links, then Google now creates a mapping between the phrase “office supplies” and the key words on that page. So, for example, if the page was an amazon link to purchase staples, Google would now start to associate the entire phrase “office supplies“ with staples. The next time someone searched this phrase, Google would also take into account its weighting of the synonyms for the phrase “office supplies”, potentially ranking those results higher than the synonyms for only the word “office”, or only the word “supplies”.

From a mathematically and statistical approach, this algorithm can also be modelled using Markov chains. Unlike Pagerank though, this algorithm would use the relevance of synonyms, and potentially the unique number of synonyms, to create its edge weights. By doing this, a transition matrix can be created that describes the likelihood of going from one web page to another. As users interact with the algorithm by clicking links, the transition matrix can be updated to weight edges higher if a user actually clicks on them, and lower if they are not clicked on. This creates a much more complicated system, due to the fact that the matrix is constantly changing. However, Google could limit how often they update their live transition matrix, perhaps to once a week, while keeping track of the changes on another server in the background.

Knowing that Google uses this algorithm, and that we know how this algorithm works, website owners can use this to help their page show up higher on the front page. One such way that this can be done is by including synonyms for your website on your page. This will increase the likelihood that the initial seeding of the dictionary will create a synonym mapping between your site and other commonly searched terms. The more terms someone can search and find your site, the more likely they are to click on it. This is especially useful if Google values having many different synonyms on a page. Although your webpage may sound silly, it may be beneficial to include many different ways to describe your product or service, using different words to do so. Another way that a website owner could use this algorithm to their advantage is by utilizing the phrase matching part of the algorithm. Going back to the office supplies example, a site selling staples may want to start also selling paper, and other office supplies, so that someone searching "office supplies" is more likely to see their site, since it will have more of the key terms on it. Another, albeit somewhat silly way, to force your page to show up is the manipulate that fact that Google uses user input to influence its algorithms. Even if you cannot afford to pay for Google ad space, you may be able to afford to pay your employees to search for terms related to your site for 30 minutes a week and only click on your site. This could potentially trick the algorithm into thinking that your site is really what people want, and not necessarily competitors. Depending on how much Google uses the user to influence its rankings, this could potentially be very effective. For example, lets say that the Google Chrome browser keeps track of how long people spend on a page before hitting the back button. Your company could have its employees use the Chrome browser, go to competitor sites, and then quickly go back to Google. This could give Google the impression that your competitor's site was not what the user was looking for, and it would decrease the relevant ranking for that site. Furthermore, you could have your employees go to your site and browse it for some time, giving Google the impression that your site was truly what they were looking for and was worth browsing.

## References

http://searchengineland.com/is-googles-synonym-matching-increasing-how-searchers-and-brands-can-be-both-helped-and-hurt-131504

https://www.techwyse.com/blog/search-engine-optimization/googles-new-way-of-finding-search-synonyms/

http://www.makeuseof.com/tag/google-show-fix/