Trust Rank Algorithm

Web Spam:

Web pages with many links to other websites. The pages may pretend to provide assistance or facts about a subject, but the help is often meaningless and the information shallow. Web spam pages use repetitive text in the copy or meta tags in order to achieve a higher ranking in search engine results. Though the definition of a spam page is ambiguous, a person can easily determine whether a web page is spam or not, but determining it for all the pages on the web is not possible. Trust Rank algorithm is used to solve this problem.

Algorithm:

Trust Rank is an iterative algorithm that gives the probability of a page being spam for all pages knowing the true value of only a few pages. To discover good pages without invoking the oracle function on the entire web, we rely on an important observation- that good pages rarely point to bad ones. However, the creators of good pages can sometimes be “tricked,” so we do find some good-to-bad links on the web; but it is a fairly good enough assumption. This notion is fairly intuitive as bad pages are built to mislead search engines, not to provide useful information. Therefore, people creating good pages have little reason to point to bad pages.

Using this idea, we can say that all the nodes a good node points towards should most probably be good. And more the number of good nodes pointing towards a node, more the probability of it also being a good node. If a good node points towards too many other nodes, the probability of all the nodes it is pointing towards to be good decreases proportionally. In practice, even if the algorithm does not accurately measure the likelihood

that a page is good, it would still be useful if the function could at least help us order pages by their likelihood of being good.

Indegree and Outdegree:
The algorithm first computes two parameters for each node - indegree and outdegree (i.e. The number of nodes pointing towards it and the number of nodes it is pointing towards respectively).

Seed set:
An initial set of nodes is selected from all the nodes, and it is calculated whether these nodes are good or not using an Oracle. The trust score of the good nodes is set to 1 initially.

Iteration Step:
For each iteration, the trust score of any node is propagated to all the nodes it is pointing towards. Instead of simply propagating the score, following are the two major methods used to do it.

Trust Attenuation:
There can be cases when a good node can point to a bad node. So, not all nodes pointed to by good nodes are necessarily good. Trust attenuation is used to make sure bad nodes don't get high scores. Two major techniques used are:

Trust Dampening:A parameter β is used to propagate a fraction of the score of nodes. This ensures that the further a node is from a good node, the less impact the good node has on it.

Trust Splitting: When a node points towards many other nodes, it's impact on each individual node's score is reduced. Eg: If a node points to 5 other nodes, each node receives ⅕ of the part of our node's score.

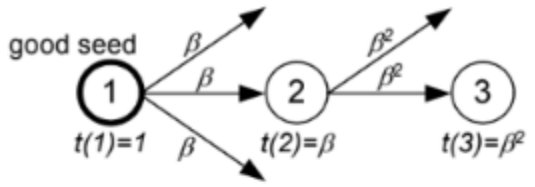


Figure 3: Trust dampening.

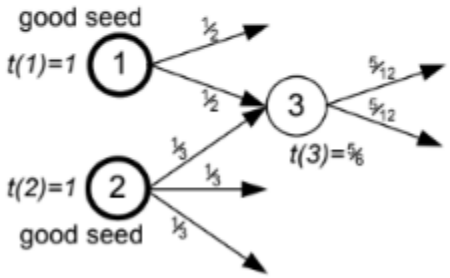


Figure 4: Trust splitting.

Trust Matrix:

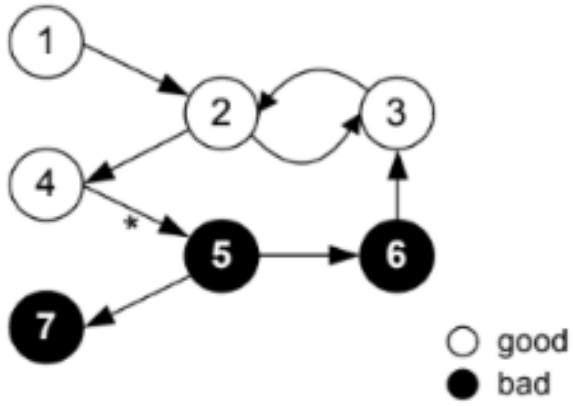
To compute the iteration step quicker and efficiently, a trust matrix is used.

T[i, j] = {0 if no edge from j to i

1/outdegree(q) if edge from j to i

So, the sum of all elements of any column is either 0 or 1.

Evaluation of final trust values: For the given set of nodes in the paper, the final trust scores are as follows:



Final Trust scores = [0.08, 0.13, 0.08, 0.10, 0.09, 0.06, 0.02]
Even though 1 is a good node, it was scored lower than node 5 because node 4 directly points towards it, and no other node points towards node 1. So, it can be said that this algorithm doesn't always work well for nodes with no indegree. And nodes with no outdegree cannot propagate their score to other nodes. Out of nodes 2 and 4, 2 has a higher score because it's score is reinforced by node 3 continuously. Other than the outliers, the algorithm works well in sorting nodes using the scores.