CS6890- Fraud Analytics Using Predictive and Social Network Techniques

Implementation of Trust Rank Algorithm

Submitted by :

Naitik Malav - CS19BTECH11026

Jai Goyal - CS19BTECH11027

Objective:

The objective of the assignment is to implement the Trust Rank Algorithm.

The algorithm takes the graph, adjacency matrix, damping factor beta as input and returns the no. of iterations required with and without considering the trust rank algorithm.

Overview of Dataset :

We have given a graph.txt file as input. In which each row has 2 nodes represnting the linkage between them. It's an unweighted directed graph. With the help of the getNodes function I have calculated the

number of unique nodes and with the help of nodes_dictionary, I have made a dictionary of the form {NODE: [Set of nodes being pointed to by the "NODE"]}.

Dataset Summary:

- total edges: 88234
- Unique Nodes: 4039

Page Rank Algorithm:

Modern search engines use more complex methods of ranking results than just conventional text ranking to offer the "best" results first. The Page Rank algorithm used by the Google search engine is one of the most well-known and influential methods for determining the relevancy of web pages. The concept behind Page Rank was that the importance of a web page may be determined by looking at the pages that link to it. If we make a web page I and link to a web page j, we're indicating that we think j is essential and relevant to our issue. If a lot of pages link to j, it suggests that the general consensus is that page j is important. If, on the other hand, j only has one backlink, but it originates from an authoritative site k, we can say that k delegated its authority to j; in other words, k asserted that j is significant. We may iteratively assign a rank to each web page based on the ranks of the pages that link to it, whether we're talking about popularity or authority.
In short, the intuition behind PageRank is that a web page is important if several other important web pages point to it.

The PageRank score, r(p) of a page p is defined as: (where alpha is a decay factor)

$$\mathbf{r}(p) = \alpha \cdot \sum_{q:(q,p)\in\mathcal{E}} \frac{\mathbf{r}(q)}{\omega(q)} + (1-\alpha)\cdot\frac{1}{N}$$

while the equivalent matrix equation form is:

$$\mathbf{r} = \alpha \cdot \mathbf{T}\cdot\mathbf{r} + (1-\alpha)\cdot\frac{1}{N}\cdot\mathbf{1}_N.$$

As a result, the score of a page p is made up of two parts: one component comes from pages that link to p, while the other (static) part is the same for all web sites. Page rank score can be computed using the defined function

It's worth noting that, whereas the conventional PageRank algorithm assigns each page the same static score, a biased PageRank version may deviate from this guideline. In the matrix equation

r = α·T· r+ (1−α)·d

vector d is a static score distribution vector of arbitrary, non-negative entries summing up to one. Vector d can be used to assign a non-zero static score to a set of special pages only; the score of such special pages is then spread during the iterations to the pages they point to.

Trust Rank Algorithm:

TrustRank is a link analysis approach that helps search engines rank pages in SERPs (Search Engine Results Pages) by distinguishing between relevant and irrelevant webpages. It's a semi-automated process, which implies some human intervention is required for it to perform properly.

```
function TrustRank
input
        T       transition matrix
        N       number of pages
        L       limit of oracle invocations
        α_B     decay factor for biased PageRank
        M_B     number of biased PageRank iterations
output
        t*      TrustRank scores
begin
        // evaluate seed-desirability of pages
(1)     s = SelectSeed(...)
        // generate corresponding ordering
(2)     σ = Rank({1,...,N}, s)
        // select good seeds
(3)     d = 0_N
        for i = 1 to L do
                if O(σ(i)) == 1 then
                        d(σ(i)) = 1
        // normalize static score distribution vector
(4)     d = d/|d|
        // compute TrustRank scores
(5)     t* = d
        for i = 1 to M_B do
                t* = α_B · T · t* + (1 − α_B) · d
        return t*
end
```

The above function computes trust scores for a web graph. The input to the algorithm is the graph (the transition matrix T and the number N of web pages) and parameters that control execution (L, MB, αB, see below).

Steps:

1. The algorithm calls function SelectSeed, which returns a vector s. The entry s(p) in this vector gives the “desirability” of page p as a seed page.

2. Function Rank(x,s) generates a permutation x' of the vector x, with elements x'(i) in decreasing order of s(x'(i)). That is, Rank(x, s) reorders the elements of x in decreasing order of their s-scores.

3. On the L most desirable seed pages, the oracle function is called. The static score distribution vector d's entries corresponding to good seed pages are set to 1.

4. Normalizing vector d so that its entries sum up to 1.

5. Evaluating TrustRank scores using a biased PageRank computation with d replacing the uniform distribution.

Results:

Transition Matrix is printed by running the submitted python code.

No. of iterations required without considering Trust Rank: 240
No. of iterations required after considering Trust Rank: 376