Assignment 2: Trust Rank Assignment

Kummitha Jhanavi CS21BTECH11032
Potta Vennela CS21BTECH11046
Vinjam Aswitha MA21BTECH11018
Pundi Bindusree CS21BTECH11048
K Vivek Kumar CS21BTECH11026

May 2, 2024

Abstract

Keywords: Trust Rank analysis, Bad Senders Analysis, Data analysis, Fraud Analytics

This paper presents a solution for conducting trust rank data analysis on a dataset provided for Assignment 2 in the Fraud Analytics course.

---

1 Introduction to Trust Rank Algorithm

The Trust Rank Algorithm is a graph-based ranking algorithm used to assess the trustworthiness or authority of nodes within a network. Similar to PageRank, Trust Rank assigns scores to nodes based on the principle of transitivity of trust, where the trustworthiness of a node is influenced by the trustworthiness of its neighbors. Nodes with higher trust scores are considered more trustworthy or authoritative within the network. The Trust Rank Algorithm plays a crucial role in various applications such as web search, social network analysis, and recommendation systems.

The Trust Rank Algorithm can be mathematically represented as follows:

$$TR(u) = (1 - \alpha) + \alpha \cdot \sum_{v \in B(u)} \frac{TR(v)}{C(v)}$$

Where:

- $TR(u)$ is the Trust Rank score of node u,
- α is the damping factor, typically set between 0 and 1,
- $B(u)$ represents the set of nodes that link to node u,
- $C(v)$ is the out-degree of node v, i.e., the number of outgoing edges from node v.

2 Problem Statement

Given a dataset `costsensitive.csv`, we have to do the cost-sensitive analysis using two approaches, Bahnsen's Approach and the Nikou and Gunnemann's Approach. Provide proper results by also testing the models on test datasets.

3 Datasets

1. **Payments.csv**: This dataset likely contains information about financial transactions, such as sender, receiver, and the amount transferred.
2. **bad_sender.csv**: This dataset likely contains a list of individuals or entities identified as "bad senders" or untrustworthy actors within the financial transaction network.

4 Approach (Algorithm Used)

1. **Data Preparation**:
   - Load the payment data from the 'Payments.csv' file and the list of bad senders from the 'bad_sender.csv' file into pandas DataFrames.
2. **Network Construction**:
   - Create a directed graph using NetworkX library to represent the financial transaction network. Each node in the graph represents an entity involved in the transactions (e.g., sender or receiver), and each edge represents a transaction between two entities, with the weight of the edge indicating the amount transferred.
3. **Trust Initialization**:
   - Initialize trust scores for each node in the network. Nodes identified as bad senders are given a low trust score (e.g., 0.3), while all other nodes are initially assigned a high trust score (e.g., 1.0).
4. **Trust Propagation Algorithm**:
   - Iterate through the network for a maximum number of iterations (e.g., 100 times) or until a convergence criterion is met.
   - For each node in the network:
     - Calculate the incoming trust from neighboring nodes by multiplying the trust score of each neighbor by the weight of the corresponding edge.
     - Update the trust score of the node based on the incoming trust and damping factor accordingly.
   - Normalize the trust scores to ensure they sum up to 1.
5. **Convergence Check**:

- Monitor the change in trust scores between iterations. If the change falls below a predefined tolerance level (e.g., 1e-12), stop the iteration.

6. **Outputs**:
   - Sort the nodes based on their final trust scores in descending order.
   - Print the top N trustworthy nodes along with their corresponding trust scores.

5 Results

We have obtained the Trust Rank scores for the nodes in the network. Below are the top 10 nodes along with their corresponding Trust Scores:

1. Node 1007 - Trust Score: 0.01969992369304464
2. Node 1088 - Trust Score: 0.01581179102421544
3. Node 1076 - Trust Score: 0.015011554846393273
4. Node 1194 - Trust Score: 0.014854357502297917
5. Node 1034 - Trust Score: 0.01460683461173565
6. Node 1199 - Trust Score: 0.014423528065757758
7. Node 1016 - Trust Score: 0.013858808130365951
8. Node 1426 - Trust Score: 0.013509189698339068
9. Node 1138 - Trust Score: 0.013410737052019906
10. Node 1037 - Trust Score: 0.013086995314874455

The first image (Figure 1) illustrates the Trust Rank scores of the top 100 nodes after looping. It provides a visual representation of how the trust scores of these nodes have evolved over the course of the algorithm's iterations. The second image (Figure 2) compares the Trust Rank scores of the top 100 nodes before and after looping. This comparison allows us to observe the changes in trust scores for these nodes, providing insights into the effectiveness of the trust propagation algorithm.

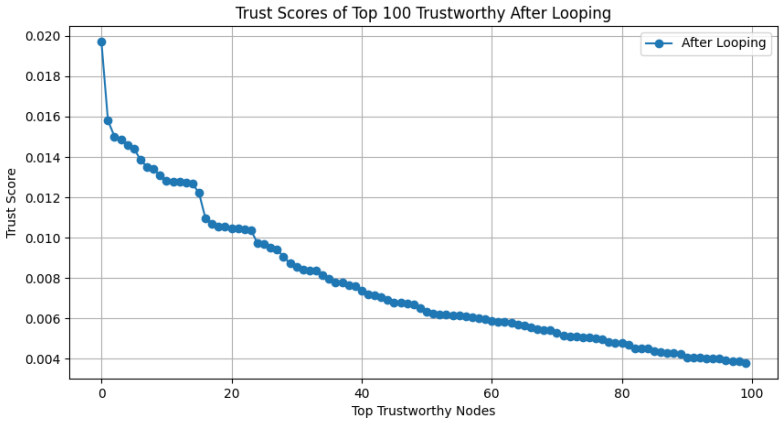

Figure 1: Trust Rank Scores of Top 100 Nodes After Looping

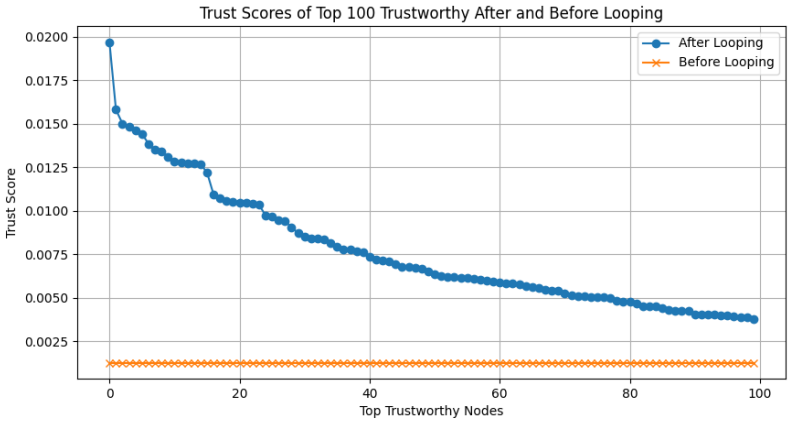

Figure 2: Trust Rank Scores of Top 100 Nodes Before and After Looping