TrustRank Algorithm Using Pregel Framework

Priyanshu Goyal MA22BTECH11015
Pranay Jain AI22BTECH11020
Varun Gupta CS21BTECH11060

Indian Institute of Technology, Hyderabad

1 Problem Statement

Search engines use PageRank to rank web pages based on link structures, but it is vulnerable to link spam, where fraudulent pages manipulate rankings. TrustRank improves this by propagating trust from human-verified pages to detect spam.

However, computing TrustRank on large-scale web graphs is computationally expensive. Pregel, a distributed graph processing framework, efficiently parallelizes trust propagation, treating pages as vertices and links as directed edges. We implement TrustRank using Pregel, iteratively propagating trust scores from known bad traders.

2 Data Input Formats

The algorithm takes two input files:

- **Payments File**: Contains records of financial transactions in the format (Sender, Receiver, Amount). It contains 130535 row of transactions.
- **Bad Traders File**: Contains a list of known fraudulent traders. It contains ID of 20 traders which are known to be bad.

3 Approach

The TrustRank algorithm works as follows:

1. A set of fraudulent traders (bad seeds) is identified.
2. Trust scores are initialized as 1/N for fraudulent accounts and 0 for others (N is total known bad seeds).
3. Trust is propagated iteratively using the Pregel framework:
    - Each vertex updates its trust score based on incoming messages.

- Trust is weighted based on transaction values.
- The process continues until scores converge.

4. The final trust scores indicate the likelihood of fraud.

4 Implementation

First, we identify the IDs of the all the traders involved in any kind of transaction and the traders with no outgoing transaction.

4.1 Graph Construction

- Directed edges are added to graph for a Sender-Receiver pair, with total amount from Sender to Receiver as weight of the edge.
- Since pregel can not handle nodes with no outgoing edge, we map the nodes with outgoing degree 0 to each bad node with an equal weight of 1.

4.2 TrustRankVertex class

The `TrustRankVertex` class extends the base `Vertex` class and implements the TrustRank algorithm using the Pregel framework. The trust score is initialized based on prior knowledge of good or bad nodes, and trust propagates iteratively. Arguments for the class objects:

- `id`: Unique identifier for the vertex (e.g., trader ID).
- `value`: The TrustRank score of the vertex, updated in each superstep.
- `out_vertices`: List of outgoing connected vertices.
- `outgoing_weights`: Dictionary storing edge weights, indicating the importance of outgoing links.
- `initial_value`: The initial trust score; 1/N for bad nodes and 0 for others.
- `dampingFactor`: The damping factor (α), controlling how much trust propagates ($\alpha = 0.85$).
- `num_supersteps`: The maximum number of iterations for trust propagation.

4.3 TrustRank Propagation

The TrustRank algorithm propagates trust iteratively through a network of connected nodes. Each node (vertex) is assigned an initial trust score, and trust is redistributed based on the structure of outgoing links.

4.3.1 Initializing Trust Scores

Each node starts with a score of $\frac{1}{N}$ if it is a known bad node, otherwise, it starts with 0:

$$T_i^{(0)} = \begin{cases} \frac{1}{|B|}, & \text{if } i \in B \text{ (bad nodes)} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where B represents the set of known bad nodes.

4.3.2 Trust Propagation Through Edges

At each superstep, trust is propagated based on outgoing link weights. The outgoing weight from node i to node j is given by:

$$w_{ij} = \frac{T(i,j)}{\sum_k T(i,k)} \tag{2}$$

where $T(i,j)$ is the transaction weight from i to j and $\sum_k T(i,k)$ is the total outgoing transaction from i, ensuring that the sum of outgoing trust proportions is normalized.

4.3.3 Trust Update Rule

The trust score of each node is updated using the formula:

$$T_i^{(t+1)} = (1-\alpha)T_i^{(0)} + \alpha \sum_{j \in I(i)} w_{ji} T_j^{(t)} \tag{3}$$

where:

- $\alpha = 0.85$ is the damping factor, ensuring stability in propagation.
- $I(i)$ represents the set of nodes with an edge to i.
- w_{ji} is the normalized outgoing weight from node j to i.

The process continues iteratively until trust scores converge or the maximum number of iterations is reached.

5 Results and Analysis

The TrustRank algorithm effectively assigns scores to traders based on their likelihood of being fraudulent. The results are analyzed using two key visualizations: a scatter plot of TrustRank scores and a histogram of the bad score distribution.

5.1 Bad Score Distribution

Figure 1 illustrates the distribution of bad scores among traders. The x-axis represents the TrustRank score, while the y-axis (log scale) represents the frequency of traders with a given score. The histogram reveals that most traders have very low scores, while a small fraction exhibits high fraud likelihood.

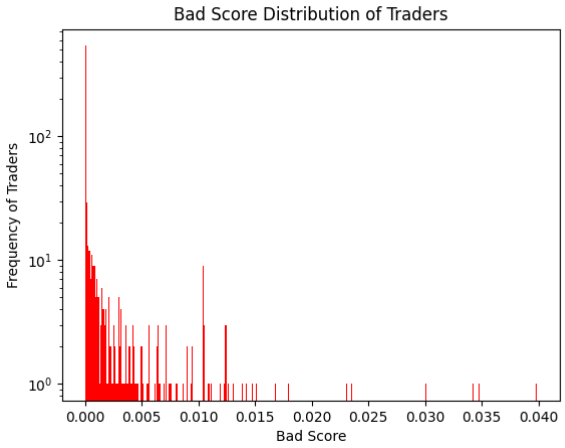

Figure 1: Bad Score Distribution.

5.2 TrustRank Score Distribution

Figure 2 presents a scatter plot of trader IDs against their corresponding TrustRank scores. The plot shows that a few traders have significantly high TrustRank scores, indicating a higher likelihood of fraud. The red markers represent known fraudulent traders, while blue markers denote other traders.

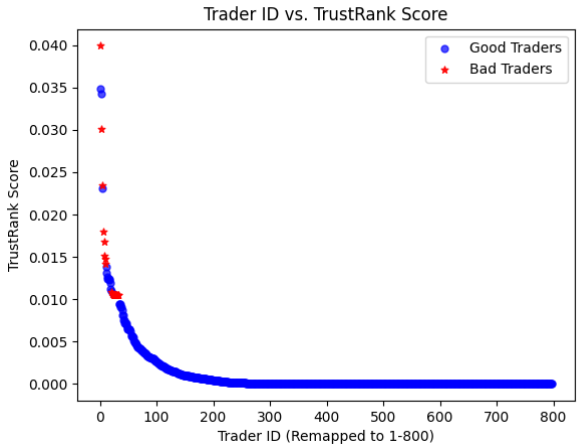

Figure 2: Trader ID vs. TrustRank Score.

5.3 Observations

- The scatter plot confirms that only a few traders have high TrustRank scores, highlighting potential fraudulent accounts.
- The histogram demonstrates that fraud scores are heavily skewed towards lower values, indicating that the majority of traders have low fraud risk.
- The use of the Pregel framework allows efficient computation of TrustRank on large transaction graphs.

The results validate the effectiveness of the TrustRank algorithm in detecting fraudulent traders by propagating trust scores iteratively.

6 Conclusion

In this report, we implemented the TrustRank algorithm using the Pregel framework to identify and rank potentially fraudulent traders based on transaction data. By propagating trust scores iteratively, the method effectively distinguishes between trustworthy and suspicious entities. The results demonstrate that fraudulent traders receive higher TrustRank scores, allowing for efficient fraud detection..