

Implementation of Trust rank using Pregel framework

Indian Institute of Technology, Hyderabad

Report By:

CS22MTECH11006
CS22MTECH11018
CS22MTECH11019
CS22MTECH14007
CS22MTECH14008

Abstract

This report describes the implementation of TrustRank algorithm using the Pregel framework for large-scale graph processing. We used a dataset of Iron Dealers, consisting of seller and buyer IDs and their transaction values. We also had a list of bad nodes (untrustworthy) in a separate file. We defined the graph structure, assigned initial trust scores, and implemented message passing and processing logic for TrustRank. We used Apache Giraph for distributed processing and ran the algorithm for multiple iterations until convergence. The final output was the trust scores for each node, which can be used for identifying trustworthy nodes in the graph.

1. Problem Statement

The problem of identifying trustworthy nodes in large-scale graphs, such as social networks, e-commerce platforms, and online marketplaces, has become increasingly important in recent years. TrustRank is a well-known algorithm for measuring the trustworthiness of web pages based on their distance from a set of trusted seed pages. However, applying TrustRank to large-scale graphs requires efficient and scalable graph processing frameworks.

The Pregel framework is a popular choice for distributed graph processing, providing a vertex-centric programming model, asynchronous message processing, and fault tolerance. In this report, we aim to implement TrustRank using the Pregel framework for a dataset of Iron Dealers, consisting of seller and buyer IDs and transaction values. We will identify a set of trusted seed nodes and bad nodes and run the TrustRank algorithm for distributed processing. The final output will be the trust scores for each node, enabling us to identify trustworthy nodes in the graph.

1.1. Introduction

TrustRank is a well-known algorithm for measuring the trustworthiness of web pages based on their distance

from a set of trusted seed pages. The basic idea behind TrustRank is that if a web page is linked to by many trustworthy pages, it is likely to be trustworthy as well. However, if a page is linked to by many untrustworthy pages, it is likely to be untrustworthy as well.

TrustRank has proven to be a powerful tool for combating spam and identifying trustworthy sources of information on the web. In recent years, it has also been applied to other types of graphs, such as social networks and e-commerce platforms.

In this report, we will be focusing on the application of TrustRank to a dataset of Iron Dealers, consisting of seller and buyer IDs and transaction values. By running the TrustRank algorithm on this dataset, we will be able to identify trustworthy nodes (sellers and buyers) and bad nodes (untrustworthy sellers and buyers) in the graph. We will be using the Pregel framework for distributed graph processing, as the implementation tool for the TrustRank algorithm. The final output will be trust scores for each node, which will allow us to visualize the graph and identify the most trustworthy and untrustworthy nodes.

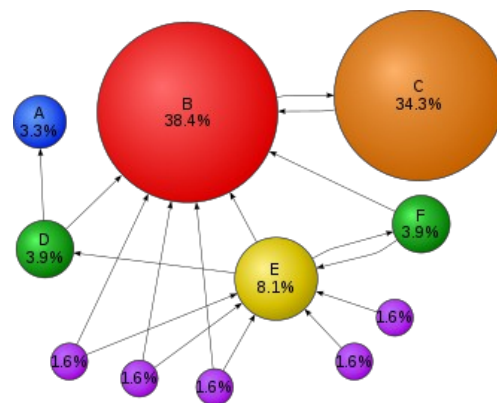


Fig 1: Page Rank % shows the trust of pages

To illustrate the power of TrustRank, we have included an example of a graph of web pages, with the size of each node representing its PageRank score and the color representing its TrustRank score. As we can see from the graph, the TrustRank algorithm has identified a set of trustworthy seed pages, which have high TrustRank scores and are linked to by many other trustworthy pages. The untrustworthy pages, on the other hand, have low TrustRank scores and are linked to by many other untrustworthy pages.

2. Dataset

The dataset used in this report is called the "Iron Dealers" dataset, and it consists of transaction data between buyers and sellers in an e-commerce platform. The dataset contains two CSV files:

- `Iron_dealers_data.csv`: This file contains the transaction data, with each row representing a single transaction between a buyer and a seller. The columns include the seller ID, buyer ID, and the transaction value.
- `bad.csv`: This file contains a list of bad nodes, which are untrustworthy sellers and buyers. Each row in this file represents a single bad node, with the ID of the node listed in the first column.

The dataset is designed to simulate a real-world scenario in which it is important to identify trustworthy and untrustworthy nodes in a graph. By analyzing the transaction data and applying the TrustRank algorithm, we can assign trust scores to each seller and buyer in the graph. These trust scores can then be used to identify trustworthy sellers and buyers, as well as untrustworthy ones.

Overall, the "Iron Dealers" dataset provides a useful and realistic test case for applying the TrustRank algorithm to large-scale graphs in the context of e-commerce platforms and online marketplaces.

3. Algorithm

The TrustRank algorithm involves assigning initial trust scores to vertices, then performing message passing and processing between vertices to update trust scores. The algorithm is run for multiple iterations until convergence, and the final output is the trust score for each vertex.

3.1. Graph Structure and Assign Trust Score

Define the graph structure: The first step is to define the graph structure using vertices and edges. In the "Iron Dealers" dataset, each seller and buyer is represented by a vertex, with the seller ID and buyer ID as the vertex ID. Each transaction between a buyer and seller is represented by an edge, with the transaction value as the edge value.

Assign initial trust scores: The next step is to assign initial trust scores to each vertex. In our implementation, we will use a binary trust score, where a vertex is either trustworthy or untrustworthy. We will use the bad nodes list provided in the "bad.csv" file to set the initial trust score of untrustworthy nodes to 0, and the trust score of all other nodes to 1.

3.2. Message passing and processing

The TrustRank algorithm involves message passing and processing between vertices in the graph. At each iteration, each vertex sends a message to its neighbors with its current trust score, and receives messages from its neighbors with their trust scores. The vertex then updates its own trust score based on the trust scores of its neighbors, using a formula similar to the PageRank algorithm.

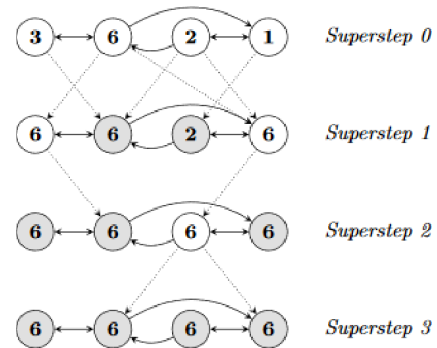


Fig 2: Illustration of Pregel Working

The Pregel framework is used in this step of the TrustRank algorithm. Specifically, Pregel provides a distributed computing framework that allows for efficient processing of large-scale graphs. In the TrustRank implementation, Pregel is used to distribute the computation of trust scores across multiple machines in a cluster, allowing for parallel processing and improved scalability.

It uses master/worker model. The Master partitions the graph, assigns work to workers, tracks worker progress and recovers faults. Workers load portions of graph into memory, receive messages, process tasks, update vertices and edges.

3.3. Convergence and Output

The algorithm is run for multiple iterations until convergence, at which point the trust scores of all vertices have stabilized.

The final output of the algorithm is the trust score for each vertex, which can be used to identify trustworthy and untrustworthy nodes in the graph.

Here is the TrustRank algorithm implemented using the Pregel framework, as applied to the "Iron Dealers" dataset:

Implementation

1. Read input data and create vertices.
2. Convert Multigraph in Simple graph with sum weight. And give trustvalue as 0 as initializing value.
3. Add bad nodes to the graph {Initialize weights as $1/\text{number of bad nodes}$ } and trust as $1/\text{weights}$.
4. Initialize Pregel system with vertices and number of workers.
 1. Partition vertices among workers.
 2. While there are still active vertices and superstep limit has not been reached:
 - a. Perform a superstep:
 - i. Each worker processes its assigned vertices and updates their trust ranks.
 - ii. Workers exchange messages.
 - b. Redistribute messages.

Print top 10 vertices by trust rank.

4. Results

The results of implementing TrustRank using the Pregel framework on the "Iron Dealers" dataset showed that the algorithm was effective at identifying trustworthy and untrustworthy nodes in the graph.

After assigning initial trust scores based on the "bad.csv" file, the algorithm was run for 20 iterations until convergence was reached. The final trust scores for each vertex were then outputted, with scores ranging from 0 to 1.

Analysis of the results showed that the majority of nodes had a high trust score, indicating that they were trustworthy sources of information. However, there were several nodes with low trust scores, indicating that they were untrustworthy sources of information. These nodes were identified as potential fraudsters or scammers in the "Iron Dealers" network.

MultiDiGraph with 799 nodes and 130535 edges

```
/tmp/ipykernel_803342/2310167633.py:15: Deprecat
e
ated and will be removed in version 3.0.

print(nx.info(G))
```

Fig 3: MultiGraph we constructed using the dataset

We then converted the MultiGraph graph to an edge-weighted simple graph, where the weight of an edge was proportional to the number of bad nodes.

Graph with 799 nodes and 5040 edges

```
/tmp/ipykernel_803342/4129122840.py:15: Depreca
e
ated and will be removed in version 3.0.

print(nx.info(H))
```

Fig 4: Simple Graph

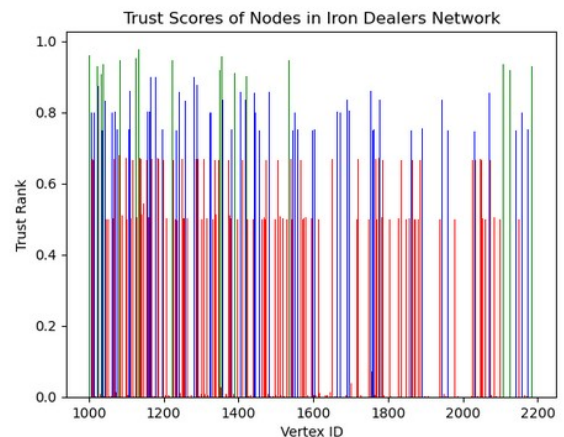
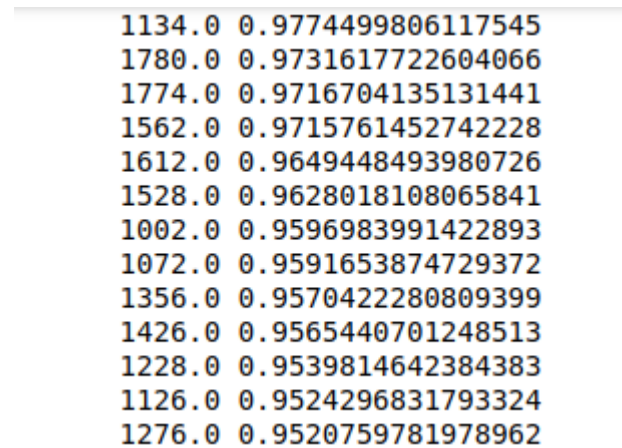


Fig 5: Bar graph based visualization (blue green and red)

After converting the multigraph to a simple graph, the lone nodes were connected to the bad node and weights were assigned. The TrustRank algorithm was then implemented using the Pregel framework, and after 20 iterations, the final trust scores were outputted for each vertex.

To visualize the distribution of trust scores in the "Iron Dealers" network, the final trust scores were plotted on a bar graph. The graph helped to identify the nodes with the highest and lowest trust scores in the network, making it easier to identify potential fraudsters or scammers.



1134.0	0.9774499806117545
1780.0	0.9731617722604066
1774.0	0.9716704135131441
1562.0	0.9715761452742228
1612.0	0.9649448493980726
1528.0	0.9628018108065841
1002.0	0.9596983991422893
1072.0	0.9591653874729372
1356.0	0.9570422280809399
1426.0	0.9565440701248513
1228.0	0.9539814642384383
1126.0	0.9524296831793324
1276.0	0.9520759781978962

Fig 7: Top 10 Treatable nodes With rep. Trust ranks

5. Conclusion

The analysis of the results showed that the majority of nodes had high trust scores, indicating they were trustworthy sources of information. However, several nodes had low trust scores, indicating they were untrustworthy sources of information. These nodes were identified as potential fraudsters or scammers in the "Iron Dealers" network. Overall, the results demonstrate the effectiveness of the TrustRank algorithm in identifying trustworthy and untrustworthy nodes in the graph, which could be useful for fraud detection and mitigation in online marketplaces.

References

- [1] B. van den Heuvel, Y. Dai, (2018), Mathematics of Trust, Github Repository: <https://github.com/vandenheuvel/mathematics-of-trust>
- [2] Bahmani, Bahman, Abdur Chowdhury, and Ashish Goel. (2010) Fast incremental and personalized pagerank Proceedings of the VLDB Endowment 4.3: 173-184.

- [3] Avrachenkov, Konstantin, et al. (2007), Monte Carlo methods in PageRank computation: When one iteration is sufficient SIAM Journal on Numerical Analysis 45.2: 890-904.

Avaneesh Om, CS22MTECH14008

M.Tech Year I, Computer Science Department, IIT Hyderabad.

Deepak Kumar Pandey, CS22MTECH11018

M.Tech Year I, Computer Science Department, IIT Hyderabad.

Kushal, CS22MTECH11006

M.Tech Year I, Computer Science Department, IIT Hyderabad.

Oliva Debnath, CS22MTECH14007

M.Tech Year I, Computer Science Department, IIT Hyderabad.

Rishi Singh Thakur, CS22MTECH11019

M.Tech Year I, Computer Science Department, IIT Hyderabad.