Southeast Asian J. of Sciences, Vol. 11, No 1 (2024), pp. 01–10

MATHEMATICAL FOUNDATIONS OF MODELING TRUST COMPUTATION IN COMPLEX NETWORKS

Dinh Que Tran

Department of Information Technology
Posts and Telecommunications Institute of Technology (PTIT)
Hanoi, Vietnam
E-mail: quetd@ptit.edu.vn

Abstract

Modeling trust computation in complex networks has become increasingly important in various domains, including social networks, telecommunication networks, multi-agent systems, transportation and distributed computing. This paper introduces a novel approach to modeling trust based on two primary metrics: belief and trustworthiness. We explore the theoretical foundations, computational methods of these metrics with a functional approach and deep learning in modeling trust in complex networks.

1 Introduction

Various models of trust computation, which have been developed in computer science and various research areas, make use of various factors including interaction, relationship among peers, propagation, contexts and so on [4][5][6][9][7][10] [11][12][8]. However, the existing approaches to trust modeling often focus on some aspect of network structures with types of interactions or reputation forms. They fail to capture the nuanced relationships that arise in large, dynamic networks. This paper addresses this gap by proposing a trust model

Key words: complex networks, trust model, belief, trusworthiness.
2010 AMS Mathematics classification: 911D30, 91D10, 68U115, 68U35, 68M14, 68M115, 68T99.

based on two key metrics of belief and trustworthiness, which are represented by means of combining various factors from network structure. The belief is used to measure the relationship degree between a peer (trustor) and a peer (trustee), whereas trusworthiness is used to measure the trustworthy degree of trustee determined by means of network interaction. In order to exhibit the feasibility of the proposed model, we implement these algorithms in Python programming language and Tensorflow for two approaches: functional approach and deep learning[1]. The computation results are given in an illustrated example. The contributions of this work include:

- A formal definition of belief and trustworthiness metrics derived from the network structure metrics.
- A computational framework for trust modeling in complex networks.

The remainder of this paper is structured as follows. Section 2 is the problem statement. Section 3 and Section 4 are respectively devoted to presenting the formal definition of belief and trusworthiness metrics and the algorithms for computing two metrics. An inllustrated example is given in Section 5. Conclusion is presented in Section 6.

2 Problem Statement

A complex network is represented as a directed graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. An additional assumption is that the graph is connected, it means that there is at least one path connected to any couple nodes in the graph. The problem of trust modeling is to define a function to compute a trust score $trust(i, j)$ for each pair of nodes $i, j \in V$ based on:

- **Belief**: A metric $b(i, j)$ capturing the direct and indirect relationships between nodes in the network structure.
- **Trustworthiness**: A node-level metric $trustworthiness(j)$ reflecting the reliability of a node on another in the network.

3 Belief Metric for Trust Computing

Belief $belief(i, j)$ is critical for trust computation because it represents the subjective probability or confidence that a node i has in the reliability, capability, or honesty of node j in a process of interaction or relationship. Trust,

[1] What is a Convolutional Neural Network (CNN)? https://www.datacamp.com/tutorial/introduction-to-convolutional-neural-networks-cnns

at its core, is inherently probabilistic and context-dependent and belief plays a pivotal role in capturing this uncertainty. Belief is not just a metric but the cornerstone of trust computation because it captures the probabilistic nature, uncertainty, and evolving dynamics of relationships in complex systems. By integrating belief into trust models, systems can make informed, adaptive, and reliable trust decisions in uncertain and dynamic environments. Unlike direct trust or reputation scores, belief incorporates:

- **Direct Interactions:** Captures the strength of the direct connection between two nodes.
- **Indirect Relationships:** Accounts for trust propagation through shared neighbors or intermediary nodes.
- **Network Topology:** Leverages structural properties like shortest paths connecting two nodes.

By considering these factors, belief becomes essential for trust modeling in scenarios where direct interaction data is sparse or unavailable.

3.1 Computation of Edge Weight

The edge weight $w(i,j)$ between nodes i and j is computed based on two factors: *normalized frequency* and *similarity.*

Definition 1. *The normalized frequency $freq(i,j)$ is defined as follows:*

$$freq(i,j) = \frac{\mathit{frequency}(i,j)}{\max_{(k,l)}(\mathit{frequency}(k,l))} \tag{1}$$

where $\mathit{frequency}(i,j)$ represents the number of direct interactions between i and j, and $\max_{(k,l)}(\mathit{frequency}(k,l))$ is the maximum frequency observed across all node pairs (k,l) in the network.

Definition 2. *The similarity $sim(i,j)$ is based on the set of common neighbors between i and j and defined as follows:*

$$sim(i,j) = \frac{|N(i) \cap N(j)|}{|N(i) \cup N(j)|} \tag{2}$$

where:

- *$N(i), N(j)$ are the set of neighbors of node i and j, respectively.*
- *$|N(i) \cap N(j)|$ is the size of the intersection of $N(i)$ and $N(j)$.*
- *$|N(i) \cup N(j)|$ is the size of the union of $N(i)$ and $N(j)$.*

Definition 3. *The edge weight $edgeWeight(i,j)$ is computed as a weighted sum of the normalized frequency and similarity:*

$$edgeWeight(i,j) = w_1 \cdot freq(i,j) + w_2 \cdot sim(i,j) \tag{3}$$

where w_1 and w_2 are weight parameters such that $w_1 + w_2 = 1$.

3.2 Belief Metric Formula

Definition 4. *The belief metric is calculated as:*

$$bel(i,j) = \alpha \cdot \frac{1}{shortestPath(i,j)} + \beta \cdot edgeWeight(i,j) \tag{4}$$

where:

- *$shortestPath(i,j)$: Measures the shortest path distance between nodes i and j, with inverse distance contributing more to belief.*
- *$edgeWeight(i,j)$: Represents the strength or reliability of the direct connection between i and j.*
- *α, β: Weights for balancing the contribution of each component.*

The algorithm for computing belief $bel(i,j)$ for all node pairs is outlined in **Algorithm 1**.

Algorithm 1 Belief Computation Algorithm

```
Require: Graph $G = (V, E)$, weight parameters $\alpha, \beta$
Ensure: Belief matrix $B$ containing $bel(i,j)$ for all node pairs
Initialize $bel(i,j) = 0$ for all $i, j \in V$
Compute shortestPath$(i,j)$ for all pairs $(i,j)$ using Dijkstra's or Floyd-Warshall algorithm
for all edges $(i,j) \in E$ do
    Extract edgeWeight$(i,j)$ from the graph
end for
for all node pairs $(i,j)$ do
    Compute $bel(i,j)$ as follows:
        $bel(i,j) = \alpha \cdot \frac{1}{\text{shortestPath}(i,j)} + \beta \cdot \text{edgeWeight}(i,j)$
end for
return Belief matrix $B$
```

4 Trustworthiness Metric and Overall Trust

While the belief is utilised to measure reliability via interaction of two peers, trustworthiness is used to exhibit the importance or influence degree of a peer in the network. It $trustWorth(j)$ is constructed from centrality measures *degree centrality*, *closeness centrality* and *eigenvector centrality*, which are studied widely in complex network analysis [1][2]. We formalise these measures here for being suitable to our paper.

Definition 5. *Degree centrality measures the number of direct connections of a node j in the network and is defined as follows:*

$$degCent(j) = \frac{deg(j)}{|V| - 1}, \tag{5}$$

where:

- *$deg(j)$: The number of edges connected to node j.*
- *$|V|$: The total number of nodes in the graph.*

Definition 6. *Closeness centrality measures how close a node is to all other nodes in the graph and is defined as follows:*

$$closeCent(j) = \frac{|V| - 1}{\sum_{i \in V, i \neq j} shortestPath(j, i)}, \tag{6}$$

where:

- *$shortestPath(j, i)$: The shortest path distance between nodes j and i.*
- *$|V|$: The total number of nodes in the graph.*

Definition 7. *Eigenvector centrality evaluates the influence of a node based on the importance of its neighbors and is defines as follows:*

$$eigenCent(j) = \frac{1}{\lambda} \sum_{i \in N(j)} w(i, j) \cdot eigenCent(i), \tag{7}$$

where:

- *$N(j)$: The set of neighbors of node j.*
- *$w(i, j)$: The weight of the edge between nodes i and j.*
- *λ: The largest eigenvalue of the adjacency matrix of the graph.*

Eigenvector centrality is solved iteratively until convergence, often using matrix-based methods [3].

Definition 8. *The trustworthiness metric $trusWorth(j)$ evaluates the reliability of a node j based on centrality measures and is defined as follows:*

$$trustWorth(j) = \lambda_1 \cdot degCent(j) + \lambda_2 \cdot closeCent(j) + \lambda_3 \cdot eigenCent(j), \quad (8)$$

where $\lambda_1, \lambda_2, \lambda_3$ are the weight parameters such that $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

Definition 9. *The overall trust of a peer trustor on a peer trustee is defined as follows:*

$$trust(i, j) = \alpha \cdot bel(i, j) + \beta \cdot trustWorth(j) \quad (9)$$

where $\alpha + \beta = 1$

The algorithm for computing trust is given in **Algorithm 2**

5 Illustrated Example

In order to illustrate the proposed trust computing method. We proceed to implement algorithms in the Python programming language a network with the number of nodes $|V| = 85$ and the number of edges $|E| = 235$ as in **Figure 1**. We utilize the functional approach to compute trust values via belief and trustworthiness then build a structure of deep learning CNN for enhancing estimation of trust values.

The results are given in Table 1 including values computed by belief function, trustworthiness function, overall composition function of belief and trustworthiness and values estimated by a simple CNN model.

Table 1: Top 10 Node Pairs with Their Values

(i, j)	**belief(i, j)**	**trustworthiness(j)**	**trust_function(i, j)**	**trust_cnn(i, j)**
(13, 14)	0.754433690124282	0.492678281904111	0.6497315268362136	0.6564876437187195
(5, 14)	0.7393110535879477	0.492678281904111	0.640657944914413	0.6471788883209229
(23, 5)	0.759603199387425	0.4380647031753637	0.6309878009026004	0.6383023858070374
(10, 5)	0.750055407387242	0.4380647031753637	0.6252591257024906	0.6324253082275391
(14, 5)	0.7393110535879477	0.4380647031753637	0.6188125134229141	0.6258114576339722
(14, 13)	0.754433690124282	0.40109720645255476	0.613099096655591	0.6206569075584412
(1, 2)	0.7800783675764401	0.35873681979877114	0.6115417484653725	0.6200504899024963
(18, 13)	0.7490748817264209	0.40109720645255476	0.6098838116168744	0.6173582673072815
(18, 14)	0.6826854556236025	0.492678281904111	0.6066825861358058	0.6123228073120117
(7, 0)	0.7017294592627983	0.4640110869317423	0.6066421103303758	0.6128294467926025

6 Conclusions

This paper has introduced a functional approach for constructing a trust model in complex network based on measures belief and trustworthiness. These metrics reflex characteristics of network structure and the importance of a node

Algorithm 2 Trustworthiness and Overall Trust Computation Algorithm

```
Require: Graph $G = (V, E)$, weight parameters $\lambda_1, \lambda_2, \lambda_3, \alpha, \beta$
Ensure: Trust matrix $T$ containing $trust(i, j)$ for all node pairs
Initialize $trust(i, j) = 0$ for all $i, j \in V$
Compute centrality measures for all nodes $j \in V$:
for all nodes $j \in V$ do
    Compute degree centrality:
        $$\text{degCent}(j) = \frac{\deg(j)}{|V| - 1}$$
    Compute closeness centrality:
        $$\text{closeCent}(j) = \frac{|V| - 1}{\sum_{i \in V, i \neq j} \text{shortestPath}(j, i)}$$
    Compute eigenvector centrality iteratively until convergence:
        $$\text{eigenCent}(j) = \frac{1}{\lambda} \sum_{k \in N(j)} w(k, j) \cdot \text{eigenCent}(k)$$
end for
Compute trustworthiness metric for each node $j$:
for all nodes $j \in V$ do
    Compute $trustWorth(j)$:
        $$trustWorth(j) = \lambda_1 \cdot \text{degCent}(j) + \lambda_2 \cdot \text{closeCent}(j) + \lambda_3 \cdot \text{eigenCent}(j)$$
end for
Compute overall trust for all node pairs $(i, j)$:
for all node pairs $(i, j)$ do
    Compute $trust(i, j)$:
        $$trust(i, j) = \alpha \cdot bef(i, j) + \beta \cdot trustWorth(j)$$
end for
return Trust matrix $T$
```

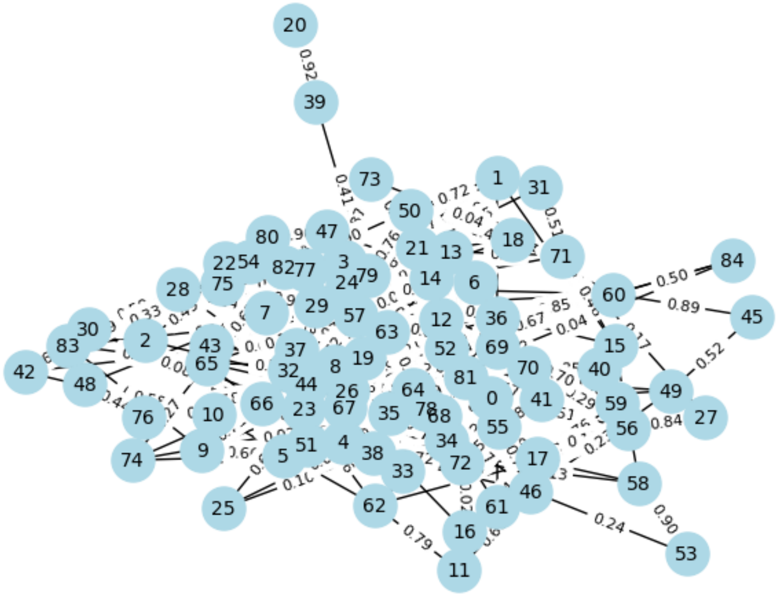

Figure 1: The Network with node_number = 85 and edge_number = 235

in network. We propose algorithms for computing trust values and illustrate the results in the generated network. These issues need to be studied furthermore to compare and evaluate various models on larger datasets. The research results will be presented in our future work.

References

[1] Matheus R. F. Mendonca, Andre M. S. Barreto, and Artur Ziviani, Ap andproximating Network Centrality Measures Using Node Embedding and Machine Learning, 2020 *https://arxiv.org/pdf/2006.16392*

[2] Cristian Riveros and Jorge Salas How do centrality measures choose the root of trees? 2022, *https://arxiv.org/pdf/2112.13736*

[3] Dal Taylor et al., Eigenvector based central measures for temporal Networks, Multiscale Model Simul. 2017 ; 15(1): 537–574, *https://pmc.ncbi.nlm.nih.gov/articles/PMC5643020/pdf/nihms908451.pdf*

[4] J. Tang et alt., Social Influence Analysis in Large-scale Networks,*Conference KDD'09, June 28, 2009*

[5] Bingoi et al., Topic-Based Influence Computation in Social Networks under Resource Constraints, *IEEE Transactions on service computing, Vol. PP, No. 99, 2018.*

[6] Ju Fan et al., OCTOPUS: An Online Topic-Aware Influence Analysis System for Social Networks, IEEE 34th International Conference on Data Engineering (ICDE), 2018. Available at: https://ieeexplore.ieee.org/document/8509399

[7] Kan Li et l., Social Influence Analysis: Models, Methods, and Evaluation, Engineering 4 (2018) 40–46

[8] Dinh Que Tran and Phuong Thanh Pham, TreeXTrust: Topic-aware Computational Trust based on Interaction Experience, Reputation of Users with Similarity and Path Algebra of Graph in Social Networks, *Computer Science Journal, AGH, Poland*, to appear.

[9] David Crandall, Dan Cosley et al. Feedback effects between similarity and social influence in online communities, KDD'08, 2008, USA.

[10] Vedran Podobnik et al. How to calculate trust between social network users? In *Software, Telecommunications and Computer Networks (SoftCOM), 20th International Conference on*, p.1–6. IEEE, 2012.

[11] Chung-Wei Hang et al., Operators for Propagating Trust and their Evaluation in Social Networks, Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS), 2009

[12] Wanita Sherchan, Surya Nepal, and Cecile Paris. A survey of trust in social networks. *ACM Comput. Surv.*, 45(4):47:1–47:33, August 2013.

[13] Phuong Thanh Pham, Dinh Que Tran, Incorporation of Experience and Reference-Based Topic Trust with Interests in Social Network, Advances in Intelligent Systems and Computing 538, Springer, 2017, M. Akagi et al. (eds.).