East West Journal of Mathematics, Vol. 27, No 1 (2026) pp. 70-80
https://doi.org/10.36853/ewjm0422

HILBERT OPERATOR–BASED TRUST COMPUTATION IN COMPLEX NETWORKS VIA TENSOR STATE REPRESENTATION

Dinh Que Tran

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

Abstract

Trust evaluation plays a central role in complex networks or network-drived systems such as social networks, collaborative systems, and online communities. Traditional trust models rely on heuristic propagation rules that do not fully exploit netwrok heterogeneous data. This paper proposes a mathematical framework for trust computation based on Hilbert space operators. The framework transforms user features of networks into tensor representations, constructs trust states from these tensors, and models trust propagation through bounded operators in Hilbert spaces. A numerical example illustrates the complete pipeline from features to trust scores.

1 Introduction

Trust evaluation is a fundamental problem in online network-drived systems such as collaborative platforms, social networks, information sharing or financial systems etc. Agents continuously interact with one another by exchanging

Key words: hilbert space, embedding, operator, complex network, trust computation.
2010 AMS Mathematics classification: 911D30, 91D10, 68U115, 68U35, 68M14, 68M115, 68T99.

information, recommending content, and forming communities. In such environments, estimating the reliability or trustworthiness of participants becomes essential for applications such as recommendation systems, misinformation filtering, fraud detection, and reputation management. As network platforms grow in size and complexity, automated trust computation mechanisms have become an important component of modern networked systems.

Early research on trust modeling focused on graph-based approaches in which a netwrok-drived system is represented as a network of agents connected by trust relationships. In these models, trust values are propagated across the network through iterative diffusion processes. Well-known examples include the EigenTrust algorithm for reputation management in peer-to-peer networks [1] and the TrustRank framework designed to combat web spam [2]. These methods demonstrate how global trust scores can emerge from local trust relationships through network propagation mechanisms. More generally, the study of trust diffusion is closely related to classical concepts in network science and spectral graph theory, where diffusion and random-walk processes are used to analyze connectivity and influence in large networks [8, 11]. Despite their success, traditional trust models typically represent trust as scalar quantities associated with nodes or edges. However, real-world social interactions are inherently multidimensional and involve heterogeneous information about agents, including behavioral patterns, communication frequency, reputation indicators, and shared interests etc. Capturing such heterogeneous attributes requires richer representations than simple weighted graphs. In many practical systems, trust relationships depend not only on direct interactions but also on complex contextual factors derived from agent features.

Recent advances in graph representation learning have significantly expanded the capability of machine learning models to capture such complex relational structures. Graph neural networks (GNNs) extend deep learning techniques to graph-structured data and enable the joint modeling of node attributes and network topology [7, 6]. These models perform iterative message passing across graph edges, allowing information from neighboring nodes to influence feature representations. More broadly, the emerging field of geometric deep learning studies learning algorithms for non-Euclidean data such as graphs and manifolds [5]. Comprehensive surveys of graph learning methods highlight their applications in recommendation systems, social network analysis, and knowledge graphs [14, 15]. Although these approaches have demonstrated strong empirical performance, the mathematical interpretation of trust propagation within such frameworks remains an active area of research.

From a theoretical perspective, *propagation processes on networks can be interpreted as operators acting on state spaces defined over the graph.* This viewpoint naturally connects graph learning models with tools from functional analysis and operator theory. In particular, Hilbert spaces provide a convenient mathematical structure for representing network states and studying the stabil-

ity and convergence properties of iterative propagation processes [12]. Viewing trust diffusion as an operator acting on a Hilbert space allows the analysis of trust dynamics using well-established tools from spectral theory.

Another challenge in trust modeling is the integration of heterogeneous agent attributes into the propagation process. Tensor representations offer a natural framework for encoding interactions between nodes and multiple feature dimensions. By extending adjacency relationships into higher-order structures, tensors allow the simultaneous modeling of network connectivity and attribute interactions. Such representations provide a richer description of network systems compared with traditional matrix-based approaches. Motivated by these observations, this paper proposes a mathematical framework for trust computation based on tensor state representations and Hilbert space operators. The proposed approach introduces a structured pipeline consisting of four stages:

1. Feature representation of users,
2. Tensor-based modeling of feature interactions across the network,
3. Construction of trust states from tensor aggregates,
4. Propagation of trust via bounded linear operators in Hilbert space.

Within this framework, trust diffusion is interpreted as an operator evolution process defined on a Hilbert space associated with the network. This perspective provides a unified mathematical view that integrates feature-based representations, tensor interaction modeling, and operator-based propagation. It also reveals connections between classical trust propagation models and modern graph representation learning methods. To illustrate the proposed framework, a numerical example is presented to show the trust computation process that combines ideas from graph theory, tensor modeling, and functional analysis. The remainder of the paper is organized as follows. Section 2 introduces the trust state representation in social networks. Section 3 describes the trust propagation in Hilbert space. Section 4 presents an numerical example. Finally, Section 5 concludes the paper and outlines directions for future research.

2 Trust State Representation in ComplexNetworks

2.1 Social Network

Definition 1. *A complex network is represented by a weighted directed graph*

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, W),$$

where

- $\mathcal{V} = \{1, \ldots, n\}$ *denotes the set of agents,*
- $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ *is the set of relationships,*
- $W : \mathcal{E} \to [0, 1]$ *is a weight function assigning a trust value to each edge.*

The weighted adjacency matrix associated with $\mathcal{G}$ is defined as

$$A = (a_{ij}) \in \mathbb{R}^{n \times n},$$

where

$$a_{ij} = \begin{cases} W_{ij}, & (i, j) \in \mathcal{E}, \\ 0, & \text{otherwise.} \end{cases}$$

The diagonal degree matrix is defined by

$$D = \operatorname{diag}(d_1, \ldots, d_n), \qquad d_i = \sum_{j=1}^{n} a_{ij}.$$

The (combinatorial) graph Laplacian is defined as

$$L = D - A,$$

where A is the adjacency matrix and D is the degree matrix. The Laplacian operator plays a fundamental role in describing diffusion and propagation processes on networks [8, 10, 9].

2.2 Feature Representation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ denote the network graph with $|\mathcal{V}| = n$ agents. Each agent $i \in \mathcal{V}$ is associated with a feature vector

$$\mathbf{f}_i \in \mathbb{R}^d,$$

which encodes observable attributes describing the agent's behavior and interactions within the network. Following the trust modeling perspective introduced in [4], agent features capture both interaction characteristics and contextual information relevant to trust estimation such as in social networks. Typical components of $\mathbf{f}_i$ may include

- Interaction frequency with other agents,
- Reputation or credibility scores,
- Similarity of interests between agents,
- Behavioral activity indicators derived from agent interactions.

Stacking all node feature vectors yields the feature matrix

$$\mathbf{F} = \begin{bmatrix} \mathbf{f}_1^\top \\ \mathbf{f}_2^\top \\ \vdots \\ \mathbf{f}_n^\top \end{bmatrix} \in \mathbb{R}^{n \times d},$$

where each row represents the feature representation of an agent node in the network.

2.3 Tensor Representation

To jointly represent network connectivity and agent attributes, we introduce a third–order tensor

$$\mathcal{T} \in \mathbb{R}^{n \times n \times d}$$

defined by

$$\mathcal{T}_{ijk} = a_{ij} f_{ik}.$$

Each slice $\mathcal{T}_{::k}$ captures the interaction structure of the network weighted by the k-th feature dimension. This representation enables simultaneous modeling of structural and attribute-based dependencies.

2.4 Trust State Construction

Using the tensor representation, we define a trust state vector

$$x \in \mathbb{R}^n$$

with components

$$x_i = \sum_{j=1}^{n} \sum_{k=1}^{d} \mathcal{T}_{ijk}.$$

The quantity x_i represents the aggregated trust potential of agent i, combining both relational influence and attribute contributions.

Proposition 1. *The trust state vector x can be written in matrix form as*

$$x = A(F\mathbf{1}_d),$$

where $\mathbf{1}_d \in \mathbb{R}^d$ denotes the vector of ones.

Proof. From the definition of the tensor,

$$x_i = \sum_{j=1}^{n} \sum_{k=1}^{d} a_{ij} f_{ik}.$$

Rearranging the sums gives

$$x_i = \sum_{j=1}^{n} a_{ij} \left(\sum_{k=1}^{d} f_{ik} \right).$$

Define

$$s_i = \sum_{k=1}^{d} f_{ik},$$

which implies $s = F\mathbf{1}_d$. Hence

$$x_i = \sum_{j=1}^{n} a_{ij} s_i$$

which in vector form yields

$$x = A(F\mathbf{1}_d).$$

□

3 Trust Propagation in Hilbert Space

3.1 Hilbert Space Formulation

Denote

$$\mathcal{H} = \ell^2(\mathcal{V})$$

the Hilbert space of square–summable real-valued functions defined on the vertex set $\mathcal{V}$, where $|\mathcal{V}| = n$. The space $\mathcal{H}$ is equipped with the inner product

$$\langle x, y \rangle = \sum_{i=1}^{n} x_i y_i, \qquad x, y \in \mathcal{H},$$

which induces the norm

$$\|x\|_2 = \sqrt{\langle x, x \rangle}.$$

A trust configuration over the network is represented by a vector

$$x = (x_1, \ldots, x_n)^\top \in \mathcal{H},$$

where x_i denotes the trust score associated with agent i. Within this functional framework, trust propagation can be modeled as the action of a linear operator

$$T : \mathcal{H} \to \mathcal{H},$$

which transforms one trust state into another through the structural interactions of the network.

3.2 Trust Propagation Operator

Definition 2 (Trust Propagation Operator)**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ be a weighted directed graph with $|\mathcal{V}| = n$. Let $A \in \mathbb{R}^{n\times n}$ denote the adjacency matrix and let $D = \mathrm{diag}(d_1, \ldots, d_n)$ be the degree matrix, where*

$$d_i = \sum_{j=1}^{n} A_{ij}, \qquad i = 1, \ldots, n.$$

The trust propagation operator *is defined as the linear operator*

$$T = D^{-1}A.$$

The operator T is the row–normalized adjacency matrix of the network and can therefore be interpreted as the transition operator of a random walk on $\mathcal{G}$. In particular,

$$T_{ij} = \frac{A_{ij}}{d_i}.$$

Thus $T_{ij} \geq 0$ and

$$\sum_{j=1}^{n} T_{ij} = \frac{1}{d_i} \sum_{j=1}^{n} A_{ij} = 1.$$

Hence T is a row–stochastic matrix. Consequently, each row of T defines a probability distribution over the neighbors of node i. In this sense, the trust value of a node at the next iteration is obtained as a weighted average of the trust values of its neighbors. Denote $x \in \mathcal{H}$ a trust state vector on the network. The propagation of trust is modeled as the discrete-time dynamical system

$$x^{(t+1)} = Tx^{(t)}, \qquad t = 0, 1, 2, \ldots,$$

where $x^{(t)}$ represents the trust state at iteration t.

Proposition 2. *The operator $T = D^{-1}A$ defines a bounded linear operator on the Hilbert space $\ell^2(\mathcal{V})$.*

Proof. The operator T is linear by construction. Since T is a row–stochastic matrix with nonnegative entries, its induced operator norm on $\ell^2(\mathcal{V})$ is bounded. In particular,

$$\|T\| \leq 1.$$

Hence T is a bounded linear operator on $\ell^2(\mathcal{V})$. □

Proposition 3 (Convergence of Trust Propagation)**.** *Suppose the graph $\mathcal{G}$ is strongly connected and let $T = D^{-1}A$ be the trust propagation operator. Then the sequence $\{x^{(t)}\}_{t\geq 0}$ generated by*

$$x^{(t+1)} = Tx^{(t)}$$

converges to a stationary trust distribution.

Proof. Since T is row–stochastic, the vector $e = (1, \dots, 1)^\top$ satisfies $Te = e$, so that 1 is an eigenvalue of T and the spectral radius satisfies

$$\rho(T) \leq 1.$$

If the graph $\mathcal{G}$ is strongly connected, the matrix T is irreducible and defines the transition matrix of an irreducible Markov chain [16, 17]. By the Perron–Frobenius theorem [18, 19], the spectral radius satisfies

$$\rho(T) = 1,$$

and there exists a unique stationary distribution π such that

$$T\pi = \pi.$$

Therefore, for any initial trust vector $x^{(0)}$,

$$x^{(t)} = T^t x^{(0)}$$

converges to a scalar multiple of π as $t \to \infty$. The vector π represents the stationary trust distribution of the network. □

4 A Numerical Example

Consider a small network with three agents. The trust relations are represented by the adjacency matrix

$$A = \begin{bmatrix} 0 & 0.6 & 0.4 \\ 0.5 & 0 & 0.5 \\ 0.2 & 0.8 & 0 \end{bmatrix}.$$

Assume that each agent is described by four features: activity, reputation, expertise, and reliability. The feature matrix is

$$F = \begin{bmatrix} 0.8 & 0.7 & 0.6 & 0.9 \\ 0.6 & 0.9 & 0.7 & 0.8 \\ 0.5 & 0.6 & 0.8 & 0.7 \end{bmatrix}.$$

4.1 Tensor Construction

To combine network structure and agent features, we construct a third-order tensor

$$\mathcal{T} \in \mathbb{R}^{n \times n \times m},$$

where $n = 3$ is the number of agents and $m = 4$ is the number of features. The tensor entries are defined by

$$\mathcal{T}_{ijk} = A_{ij} F_{ik}.$$

Thus the tensor captures the interaction between trust links and agent attributes. For example,

$$\mathcal{T}_{1,2,1} = 0.6 \times 0.8 = 0.48,$$

$$\mathcal{T}_{1,3,2} = 0.4 \times 0.7 = 0.28.$$

4.2 Trust State in Hilbert Space

Aggregating over neighbors and features yields a trust state vector

$$x \in \mathcal{H} = \ell^2(\mathcal{V}).$$

For this example we obtain

$$x = (1.36,\ 1.50,\ 0.84).$$

The vector x represents the global trust state of the network in the Hilbert space.

4.3 Operator Propagation

Let

$$T = D^{-1}A$$

be the trust propagation operator. The evolution of the trust state is governed by

$$x^{(t+1)} = Tx^{(t)}.$$

For illustration, applying the adjacency matrix gives

$$x^{(1)} = Ax = \begin{bmatrix} 1.18 \\ 1.10 \\ 1.47 \end{bmatrix}.$$

This example illustrates how agent features are lifted to a tensor representation, aggregated into a Hilbert-space trust state, and propagated across the network via the trust operator. It also shows the complete modeling pipeline proposed in this paper. Agent attributes are first represented by the feature matrix F. These features are then integrated with the network structure through the tensor construction $\mathcal{T}$, which captures interactions between trust relations and agent characteristics. Aggregating the tensor along the structural and feature dimensions produces a trust state vector $x \in \ell^2(\mathcal{V})$, which represents the global trust configuration of the network in a Hilbert space framework. Finally, the evolution of trust is governed by the propagation operator $T = D^{-1}A$, leading to the dynamical system

$$F \ \rightarrow \ \mathcal{T} \ \rightarrow \ x \in \ell^2(\mathcal{V}) \ \rightarrow \ x^{(t+1)} = Tx^{(t)}.$$

This pipeline provides a unified representation linking feature information, tensor interactions, Hilbert-space states, and operator-based trust diffusion on the network.

5 Conclusion

This paper proposed a Hilbert space operator framework for trust computation in complex networks. Agent features are first integrated with network structure through a tensor representation, which is then aggregated to form trust states in the Hilbert space $\ell^2(\mathcal{V})$. Trust propagation is subsequently modeled as a linear operator acting on these states, leading to a dynamical system that describes the diffusion of trust across the network. The proposed formulation provides a unified perspective linking feature representations, tensor interactions, and operator-based trust propagation. Future work will investigate nonlinear operator learning and manifold-based representations to capture more complex trust dynamics in large-scale complex networks.

References

[1] Kamvar, S. D., Schlosser, M. T., Garcia-Molina, H. The EigenTrust algorithm for reputation management in P2P networks. In: *Proceedings of the 12th International World Wide Web Conference (WWW)*, pp. 640–651, 2003.

[2] Gyöngyi, Z., Garcia-Molina, H., Pedersen, J. Combating web spam with TrustRank. In: *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB)*, pp. 576–587, 2004.

[3] Golbeck, J. *Computing and Applying Trust in Web-Based Social Networks*. PhD Thesis, University of Maryland, USA, 2009.

[4] Tran Dinh, Q., Pham Phuong, T. Treextrust: Topic-aware computational trust based on interaction experience, reputation of users with similarity and path algebra of graphs in social networks. *Journal of Computer Science, AGH*, 26(2), 2025.

[5] Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P. Geometric deep learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34(4), 18–42, 2017.

[6] Kipf, T. N., Welling, M. Semi-supervised classification with graph convolutional networks. In: *International Conference on Learning Representations (ICLR)*, 2017.

[7] Scarselli, F., Gori, M., Tsoi, A., Hagenbuchner, M., Monfardini, G. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1), 61–80, 2009.

[8] Chung, F. *Spectral Graph Theory*. CBMS Regional Conference Series in Mathematics, American Mathematical Society, Providence, RI, 1997.

[9] Mohar, B. The Laplacian spectrum of graphs. In: Alavi, Y., Chartrand, G., Oellermann, O., Schwenk, A. (eds.) *Graph Theory, Combinatorics, and Applications*, Vol. 2, Wiley, pp. 871–898, 1991.

[10] Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., Vandergheynst, P. The emerging field of signal processing on graphs. *IEEE Signal Processing Magazine*, 30(3), 83–98, 2013.

[11] Newman, M. E. J. *Networks: An Introduction*. Oxford University Press, Oxford, 2010.

[12] Rudin, W. *Functional Analysis*. 2nd Edition, McGraw–Hill, New York, 1991.

[13] Goodfellow, I., Bengio, Y., Courville, A. *Deep Learning*. MIT Press, Cambridge, MA, 2016.

[14] Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M. Graph neural networks: A review of methods and applications. *AI Open*, 1, 57–81, 2020.

[15] Hamilton, W. L. *Graph Representation Learning*. Morgan & Claypool Publishers, 2020.

[16] Norris, J. R. *Markov Chains*. Cambridge University Press, Cambridge, 1998.

[17] Levin, D. A., Peres, Y., Wilmer, E. L. *Markov Chains and Mixing Times*. American Mathematical Society, 2009.

[18] Meyer, C. D. *Matrix Analysis and Applied Linear Algebra*. SIAM, Philadelphia, 2000.

[19] Berman, A., Plemmons, R. J. *Nonnegative Matrices in the Mathematical Sciences*. SIAM, Philadelphia, 1994.

[20] Lovász, L. Random walks on graphs: A survey. In: *Combinatorics, Paul Erdős is Eighty*, Vol. 2, pp. 1–46, 1993.