A Hybrid Approach to Trust Inference in Social Networks

Maryam Fayyaz[1], Hamed Vahdat-Nejad [2], and Mahdi Kherad[3]
[1] Department of and Computer Engineering, Islamic Azad University of Birjand , Birjand, Iran
[2,3] Faculty of Electrical and Computer Engineering, University of Birjand, Birjand, Iran
maryam_fayaz71@yahoo.com, vahdatnejad@birjand.ac.ir, m.kherad@birjand.ac.ir

Abstract **- The trust inference issue in a social network is defined as anticipating the trust level which a user can have to another user who is not directly connected to him in the trust network. This paper proposes a method for trust inference using soft computing. To our best knowledge, it is the first time that soft computing is used to solve the trust inference issue. One of the main advantages of the proposed method is that, unlike previous methods, it is not limited to one type of trust network, and it can also be used for trust networks with different trust values. The proposed method is applied on the standard trust network and is compared to other similar methods. Experimental results show that it is able to produce more accurate results in comparison with previous methods.**

KEYWORDS - Trust Inference, Social Network, Soft Computing

I. INTRODUCTION

Trust plays an important role in the formation of the relations between users. In fact, users share their information according to their trust on other users or make decision based on provided information by other users. We deal with a graph in social networks which its vertices are users and edges are relations between them. The main issue is how to inference trust between people who are not connected directly.

Social network is a term used for the first time in 1954 by [1] who was active in the field of Social studies [2]. He studied a research about social groups in Norway and used 'social network' term in that research to describe the relationship between humans and analyse communication mechanisms. A social network is a graph G = (V, E) in which $V = \{v_1, v_2, v_3, \cdots\}$ is set of vertices and $E = \{e_1, e_2, e_3, \cdots\}$ is set of edges and each edge interconnects a pair of vertices together.

Any computational model, which is proposed for trust inference up to now, suggests a particular representation method. [3] and [4] consider a discrete set of values and a continuous numerical range to show trust, respectively. [5] and [6] select the continuous range of [0,1] as the set of allowed values to show trust. [7] considers both continuous range of [0,10] and discrete binary values of 0 and 1.

Models that utilize social network structure are specially based on trust concepts of web or friend of friend [8]. [5] presents an algorithm to traverse trust graph and infer trust. TidalTrust model [7] reviews the value of trust using numbers at the range of 0 to 10. This model is simple and its low complexity leads to high scalability. In the current research, trust values are considered through paths, as a result only the shortest path form source to destination is checked.

Although soft computing is a powerful tool for solving similar problems, is has not been used in previous trust inference methods. One of the most important advantages of the proposed method is that unlike previous methods, it is not limited to one type of trust network, but applicable to different trust networks with various trust values.

This research aims to infer trust in a social network based on social behavior. In fact, the aim is predicting the trust that a user can have to another user who is not connected directly to. The genetic algorithm and neural network are used in the proposed method. Neural network has not been used in any of previous trust inference methods. In the proposed method, three features of the social network are exploited, which represent different aspects of trust. Therefore, a model based on neural network predicts trust values regarding these features. Finally, genetic algorithm is utilized to set the weights and balance the neural network. The experimental results show higher precision for the proposed method in comparison to BBK [9], Simple average [1], TidalTrust [7], TISoN [10] and κ-FuzzyTrust [11] methods in estimating the amount of trust.

After this introduction, the proposed method is presented in section 3. In section 4, the experimental results are discussed. Last of all, the final section deals with the conclusion and future research.

II. THE PROPOSED METHOD

We face with two problems when working with neural networks: choosing the right architecture, and choosing the right training algorithm. The architecture of neural network includes number of hidden layers, number of neurons in hidden layers and the stimulation function. Each of these parameters affects the performance of neural network, directly and significantly [12].

The most common neural training algorithm is Back propagation algorithm [13]. The problem of Back propagation algorithm is late convergence and also stopping in local optimized points. One approach in training neural networks is using innovative algorithms such as genetic that in fact, is considered as a part of soft computing [14]. Genetic algorithms are from a family of computational models inspired from Evolution theorem. They indicate a possible solution for specified problems using the data structure of chromosome and apply combined operations on this data structure to protect vital information [15]. The genetic algorithm is an optimization mechanism according to the process of selecting the best in the nature [16]. In a genetic neural system, every chromosome indicates weight values and biases. To determine the fitness value of each chromosome, neural network runs with weight and bias values of the chromosome and neural network error is calculated as the fitness function of the Genetic algorithm [14].

The main steps of the proposed method are as follows:

i. Loading network information: At first, adjacent matrix of trust social network graph is loaded.
ii. Feature extraction: In this stage, for each direct link in the network graph, four characteristics are calculated and a sample data is added to the training data. The output class corresponding to each of these data samples is the value of link or the trust between two.
iii. Setting up the neural network: In this step, the proposed.
iv. Setting up the genetic algorithm: In this stage, the genetic system is created for adjusting the parameters of the neural network. The length of a chromosome is equal to the number of weights and biases of the neural.
v. Finalization of neural network: At the end, the best obtained chromosome determines the best weights for neural network.

In each iteration, one link (u,v) of the trust graph is eliminated temporarily, and the features of the link are computed. The process is iterated for all links. These features contain following items, which are considered as input for neural network:

Mean trust of source node u (MST): This feature indicates the average of trust values that the source node u has to its neighbouring nodes.

$$MST_u = \frac{\sum_{j \in adj^+(u)} t_{uj}}{|adj^+(u)|} \quad (1)$$

Where $adj^+(u)$ is the set of neighboring nodes of u, that exists a link from u to them and t_{uj} is the trust value of node u to the node j.

Mean trust of destination node v (MDT): This feature shows the average of trust values that neighbouring nodes u have to node v.

$$MDT_v = \frac{\sum_{j \in adj^-(v)} t_{jv}}{|adj^-(v)|} \quad (2)$$

Where $\mathrm{adj}^-(v)$is the set of neighbors of v that there exist links from them to v and t_{jv} is the trust value of node j to node v.

Distance: This feature points to the value of the shortest path between a pair of source and destination nodes. The greater the distance between the two nodes of source and destination, the less influenced is the relation between source and destination user. In fact, the estimated trust value of source user to the destination user is influenced by the distance between them.

Multilayer perceptron neural network (MLP) is used for predicting trust. Since three features of MST, MDT and Distance are considered, the number of inputs of neural network is three. The proposed neural network consists of ten outputs, which are the estimated trust value. The number of neurons of the input layer with the number of features of input data and the number of neurons of output layer with the number of outputs are equal, respectively. The number of hidden layers is three, because a neural network with more than two layers is able to solve any kind of problem. Figure 1 shows the proposed neural network architecture.

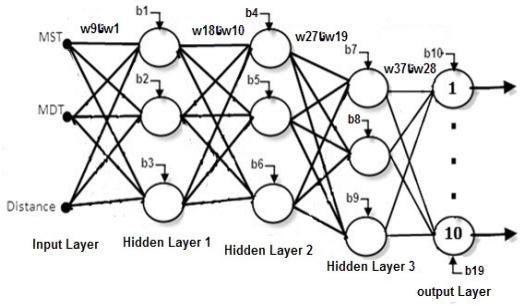

Figure 1: Architecture of the proposed neural network.

As it can be seen in Figure 1, the total number of neurons is equal to 19. Hence, the number of biases and weights to train the neural network is 19 and 37, respectively (one bias is considered for each neuron). The aim of the genetic algorithm is to determine the biases and the optimized weights of the neural network for the estimation of trust. In a chromosome, the genes of 1 to 37 indicate the weights from w1 to w37 of the neural network and the genes from 38 to 56 indicate values of neurons' biases (b1 to b19). Therefore, each chromosome has 56 genes that are able to take a value in the range of -1 to 1. Figure 2, shows the structure of a chromosome for training the neural network.

Figure 2: The structure of a chromosome for training the proposed neural network.

Weights and biases are set using the genetic algorithm so that output trust has minimum error and maximum precision. The fitness function is given in formula 3.

$$f(x) = \sum_{i=1}^{n} |t_{ri} - t_{xi}| \qquad (3)$$

Where f(x) is the fitness function of the chromosome x, n is the number of training data elements, $t_{ri}$is the value of real trust for ith data element, and $t_{xi}$is the value of output trust of neural network generated by weights of chromosome x.

III. EXPERIMENTS

The social network used in this research is a part of trust project of mindswap [17] and FilmTrust [18]. Mindswap is created of obtained data from semantic web. In this network, users give the rank of trust between 1 (minimum trust) to 10 (maximum trust). Mindswap consists of about 2000 members with more than 2500 relations. FilmTrust is a dataset of a website, in which people comment their opinions about different movies and also give a trust value between one to ten to others' opinions. This collection consists of about 900 users and 1067 links (direct trust) between them.

Matlab software is used for implementing the proposed method. 70 percent of data is considered for training, 15 percent as the test data, and 15 percent as validation data for neural network. In the genetic algorithm, initial population is 100, number of iterations is 1000, crossover rate is 0.8 and mutation rate is 0.2.

The proposed method is compared with five other methods of trust inference including BBK [9], simple average [1], TidalTrust [7], TISoN [10] and κ-FuzzyTrust [11]. These methods take two trust nodes in a trust network and calculate how much trust one node has to the other node. To determine the precision, Δ is calculated, which is the difference between actual value of trust between two nodes and the trust value inferred using the algorithm. In Table I, the average value of Δ is given for each of the methods over the dataset.

Table 1: The Average of accuracy for different methods of trust inferencing

Results on mindswap dataset					
Proposed method	κ-Fuzzy Trust	TISoN	Simple Average	BBK	Tidal Trust
1.07	1.33	1.24	1.43	1.59	1.09
Results on FilmTrust dataset					
Proposed method	κ-Fuzzy Trust	TISoN	Simple Average	BBK	Tidal Trust
1.41	1.52	1.49	1.72	1.64	2.38

As Table 1 shows, the proposed method achieves more accurate results in comparison with previous methods.

IV. CONCLUSION

In this paper, a hybrid model for trust inference in social networks using genetic algorithm and neural network has been proposed. In fact, the proposed neural network system is constituted based on the genetic algorithm. To evaluate the proposed method, the model has been coded in Matlab and implemented on validated social networks. Due to the obtained results, the proposed algorithm is an appropriate method in solving trust inference. The results confirm that this

method is able to produce trust values close to the actual ones.

V. REFERENCES

[1] J. Golbeck, "Trust on the World Wide Web: A survey," *Found. Trends Web Sci.* , vol. 1, no. 2, pp. 131–197, 2006.

[2] J. Scott, *Social network analysis*. Sage, 2017.

[3] E. Elsalamouny, V. Sassone, and M. Nielsen, "HMM-based trust model," in *6th International Workshop on Formal Aspects on Security and Trust (FAST)* vol. 5983, pp. 21–35, 2010.

[4] R. Xiang, J. Neville, and M. Rogati, "Modeling relationship strength in online social networks," in *19th International Conference on World Wide Web (WWW'10)*, New York, 2010, pp. 981–990: ACM Press.

[5] A. Josang, "Probabilistic logic under uncertainty," in *the thirteenth Australasian symposium on Theory of computing*, Darlinghurst, Australia, 2007, vol. 65, pp. 101-110: Computer Society.

[6] J. Tang, Y. Chang, C. Aggarwal, and H. Liu, "A survey of signed network mining in social media," *ACM Computing Surveys (CSUR)*, vol. 49, no. 3, p. 42, 2016.

[7] J. A. Golbeck, "computing and a applying trust in web-based social networks," PhD thesis, Department of Computer Science, University of Maryland, Maryland, College Park, MD, USA, 2005.

[8] W. Sherchan, S. Nepal, and C. Paris, "A Survey of Trust in Social Networks," *ACM Computing Surveys*, vol. 45, no. 4, 2013.

[9] Y. Wang, Z. Cai, G. Yin, Y. Gao, and Q. Pan, "A trust measurement in social networks based on game theory," in *International Conference on Computational Social Networks*, 2015, pp. 236-247: Springer.

[10] S. Hamdi, A. L. Gancarski, A. Bouzeghoub, and S. B. Yahia, "Tison: Trust inference in trust-oriented social networks," *ACM Transactions on Information Systems (TOIS)*, vol. 34, no. 3, p. 17, 2016.

[11] S. Chen, G. Wang, and W. Jia, "κ-FuzzyTrust: efficient trust computation for large-scale mobile social networks using a fuzzy implicit social graph," *Information Sciences*, vol. 318, pp. 123-143, 2015.

[12] B. D. Ripley, *Pattern recognition and neural networks*. Cambridge university press, 2007.

[13] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[14] X.-S. Yang and M. Karamanoglu, "Swarm intelligence and bio-inspired computation: an overview," in *Swarm Intelligence and Bio-Inspired Computation*: Elsevier, 2013, pp. 3-23.

[15] S. Karakatič and V. Podgorelec, "A survey of genetic algorithms for solving multi depot vehicle routing problem," *Applied Soft Computing*, vol. 27, pp. 519-532, 2015.

[16] K. Sastry, D. E. Goldberg, and G. Kendall, "Genetic algorithms," in *Search methodologies*: Springer, pp. 93-117, 2014,.

[17] *2017, Trust Project Network*. Available: http://trust.mindswap.org .

[18] 2018, *FilmTrust*. Available: https://www.librec.net/datasets/filmtrust.zip .