

ML-ELM-NeuralWalk: 在线社交网络的信任评估

张硕硕

烟台大学计算机与控制工程学院, 山东 烟台
Email: 307470478@qq.com

收稿日期: 2021年1月7日; 录用日期: 2021年2月2日; 发布日期: 2021年2月8日

摘要

信任关系在网上购物、推荐系统、物联网等方面发挥着重要作用。在线社交网络(OSN)中用户之间的信任评估问题引起了广泛关注, 已成为社会计算领域的一个热点问题。但是, 在OSN中信任的传递和聚合方式以及信任计算的准确性还不清楚。首先, ML-ELM-WalkNet学习两跳信任计算规则, 计算OSN内用户之间的两跳信任。然后, ML-ELM-NeuralWalk用计算出的信任值更新OSN, 通过迭代调用ML-ELM-WalkNet实现用户间多跳信任的计算。与采用推断方式的传统解决方案不同, ML-ELM-WalkNet能够以归纳的方式学习信任计算规则, 并准确计算出用户之间的间接信任。在两个实际OSN数据集上进行的实验表明, ML-ELM-NeuralWalk的性能优于现有的解决方案。

关键词

信任评估, 在线社交网络, 多层极限学习机, 机器学习

ML-ELM-NeuralWalk: Trust Evaluation for Online Social Networks

Shuoshuo Zhang

School of Computer and Control Engineering, Yantai University, Yantai Shandong
Email: 307470478@qq.com

Received: Jan. 7th, 2021; accepted: Feb. 2nd, 2021; published: Feb. 8th, 2021

Abstract

Trust relationship plays an important role in online shopping, recommendation systems, Internet

of Things, etc. The problem of trust evaluation among users in online social network (OSN) has attracted much attention, and has become a hot issue in the domain of social computing. However, the way of trust propagation and aggregation in OSN is still not clear, as well as the accuracy of trust calculation. In order to calculate the indirect trust, an ML-ELM-NeuralWalk algorithm to implement trust propagation and aggregation is proposed. ML-ELM-WalkNet firstly learns two-hop trust calculation rules, calculates two-hop trust among users in the OSN. After that, ML-ELM-NeuralWalk updates the OSN with the calculated trust value, so as to realize the calculation of multi-hop trust among users through iterative calling ML-ELM-WalkNet. Unlike traditional solutions that use inference methods, ML-ELM-WalkNet can learn trust calculation rules in an inductive way and accurately calculate indirect trust between users. Experiments on two real OSN datasets showed that ML-ELM-NeuralWalk outperforms existing solutions.

Keywords

Trust Assessment, Online Social Networks, Multi-Layer Extreme Learning Machine, Machine Learning

Copyright © 2021 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

如今,随着通信技术的飞速发展,Facebook、Twitter、QQ、微信、微博等在线社交网络(Online social networks, OSN)已经成为人们生活中的不可缺少部分[1],用户在网上发布个人动态,订阅他们感兴趣的用户。通常在线社交网络中用户之间的互动会产生大量的信息[2] [3],例如动态和评论。但对于用户来说 OSN 中大部分的数据可能是其不感兴趣的,小部分感兴趣的信息又存在不可靠问题。在这种情况下,信任可以帮助用户得到感兴趣和可靠的信息,从而缓解信息过载和可信度问题。因此,信任在 OSN 中起着至关重要的作用,信任社交网络(Trust social networks, TSN)引起了学者的广泛关注。

TSN 中的信任评估研究可以大致分为两类:信任建模和信任计算。信任建模是研究如何将大量原始信息转换为可以计算的信任。当前的研究使用多种理论来设计信任模型,包括主观逻辑、模糊逻辑和 Dempster-Shafer (DS)证据理论等。信任计算是研究如何计算没有直接交互的两个用户通过其他用户的推荐形成的间接信任。TSN 中有一些现有的信任度计算方法,例如 TidalTrust [4]、MoleTrust [5]、Mobifuzzytrust [6]、TISoN [7]和[8] [9] [10]等方法都是基于推断的方式来获取信任传播或信任聚合的规则。但是,这种方式往往很难捕捉到信任传递和信任聚合的本质,进而使信任计算的准确性受到严重影响。因此,信任传播和聚集的规则不清楚。与用推断的方式计算信任不同,NeuralWalk [11]使用归纳的方式进行信任计算,其通过神经网络来学习信任传递和信任聚合的规则。

TSN 和神经网络在结构上有一些相似之处。此外,神经网络中神经元的激活过程与 TSN 中用户的信任传播过程非常相似。两者都处理接收到的数据并根据处理后的数据与阈值的大小关系判断是否将其向下传递,如果处理后的数据低于阈值,不会向下传递,否则就将其向下传递。因为 NeuralWalk [11]在信任聚合中使用了平均值的方法,并且在训练神经网络中使用了反向传播(Back propagation, BP)算法,所以 NeuralWalk 存在容易陷入局部最优且算法速度较低等缺点。在[12]中, Kasun 提出了多层极限学习机(Multi-layer extreme learning machine, ML-ELM)来训练多隐层前馈神经网络,与传统的 BP 算法相比,它在学习速度和泛化性能上都有很大的优势。

由于本文研究的信任评估问题也可以看作是一个分类问题，受 ML-ELM 的启发，鉴于 NeuralWalk 的缺陷，我们提出了一种名为 ML-ELM-NeuralWalk 的信任计算方法。ML-ELM-WalkNet 将从训练数据集中学习两跳信任计算规则，并计算 TSN 中用户之间未知的两跳信任。然后，ML-ELM-Neuralwalk 将使用计算出的信任值更新 TSN。最后，可以通过在 TSN 中迭代调用 ML-ELM-Walknet 来计算用户之间的多跳信任。

我们的方法与 NeuralWalk 之间的区别如下。首先，NeuralWalk 的神经网络架构 WalkNet 更加复杂，其使用了在神经网络中嵌套神经网络的方法。而 ML-ELM-WalkNet 是通过一个简单的多层神经网络实现的。其次，NeuralWalk 中内部的神经网络权重参数需要通过外部神经网络的误差传播来更新，整个神经网络只能通过 BP 算法进行训练。而 ML-ELM-NeuralWalk 中的神经网络训练是通过 ML-ELM 实现的，具有训练速度快，可以求得唯一最优解等优点。

这篇文章的贡献在于以下三个方面。首先，针对 BP 算法在训练神经网络时存在的不足，通过使用 ELM 有效缓解了这一点，对于信任计算过程中信任传递和信任聚合的规则还不明确的问题，基于 ELM 设计了一种神经网络架构 ELM-WalkNet 来学习二跳信任计算规则。其次，通过迭代使用 ELM-WalkNet 提出了一种多跳信任计算方法 ELM-NeuralWalk。然后，在两个著名的真实 TSN 数据集上进行的实验表明，ELM-WalkNet 能快速有效的学习二跳信任计算规则，而且 ELM-NeuralWalk 算法能够准确的计算用户之间的多跳信任，即，ELM-NeuralWalk 性能优于现有的解决方案。

2. 形式化描述

2.1. 问题公式化

用一个有向图 $G(V, E, W)$ 来表示一个信任社交网络(TSN)，其中 V , E 和 W 分别是顶点、边和边上的权重的集合。图的一个顶点 $i(i \in V)$ 代表社交网络中的一个实体，图的一个有向边 $e_{ij}(e_{ij} \in E)$ 表示实体 i 对实体 j 有一个信任值，这里的 $w_{ij}(w_{ij} \in W)$ 就表示实体 i 对实体 j 的信任值。我们的信任评估问题可以表述为:给出一个信任社交网络，对于 $\forall i$ 和 $\forall j$, $s.t.i, j \in V$ 且 $i \neq j$, i 和 j 之间, \exists 至少一条路径, 且 i 和 j 不直接相连, 计算 i 对 j 的信任值。

2.2. 信任传递和信任聚合

由于 TSN 中的直接信任关系过于稀疏，而使用间接信任关系可以有效的缓解这一问题。间接信任是指在 TSN 中没有直接交互关系的两个用户之间通过他人推荐形成的信任关系。信任传递和信任聚合是 TSN 中进行信任计算的两种基本操作。如图 1 所示，对于两个用户 i 和 j ，假设 i 对 j 存在间接信任关系，从 i 到 j ，其二跳信任路径的个数为 $m \in [1, 2, \dots, n-2]$ 条(这里 n 表示 TSN 中用户的个数)，二跳信任路径上的信任值用标量 $r_{i1}, r_{1j}, \dots, r_{is}, r_{sj}$ 表示。对于用户 i 到用户 j 之间的一条信任路径 i 经过 s 到 j ，经过信任传递产生一个信任值 $\overline{r_{isj}}$ ， $\overline{r_{isj}}$ 代表了 i 基于 s 对 j 的推荐而得到的对 j 的间接信任。为了区别于原始的直接信任值，用 $\overline{r_{ij}}$ 来表示 i 对 j 的间接信任。用 $\overline{r_{i1j}}, \dots, \overline{r_{isj}}$ 表示从 i 到 j 的 m 个二跳信任路径经过信任传递得到的 m 个间接信任， $\overline{r_{i1j}}, \dots, \overline{r_{isj}}$ 经过信任聚合产生一个新的信任值 $\overline{r_{ij}}$ ， $\overline{r_{ij}}$ 表示基于 i 到 j 之间的 m 个二跳信任路径得到的 i 对 j 的间接信任。

3. ML-ELM-WalkNet

WalkNet [11]是通过神经网络嵌套神经网络来实现的。采用 BP 算法来更新整个神经网络中的参数。但是，它存在一些缺陷，例如复杂的模型结构，在信任聚合操作中直接使用平均值的方法以及在神经网络的训练中使用相对基础的 BP 算法。本文基于归纳法学习信任传播和聚集规则的思想，提出了

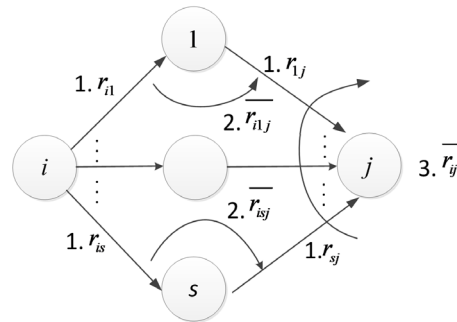


Figure 1. The process of trust calculation between two users

图 1. 两个用户之间的信任计算过程

ML-ELM-WalkNet 概述

使用 TSN 中的原始信任关系作为训练集的标签来训练 ML-ELM-WalkNet, 以学习两跳信任计算规则, 然后通过训练后的 ML-ELM-WalkNet 计算 TSN 中用户之间未知的两跳信任。在 ML-ELM-WalkNet 中, 将 TSN 中两个用户之间的两跳信任路径上的信任值用作神经网络的输入, 然后通过多隐层前馈神经网络获得神经网络的输出来模拟 TSN 中通过信任传递和聚合两个操作来计算两个用户之间的两跳信任的过程。

对于两个用户 i 和 j , 假设 i 对 j 存在间接信任关系, 从 i 到 j , 其二跳信任路径的个数为 $m \in [1, 2, \dots, n-2]$ 条, 二跳信任路径上的信任值用标量 $r_{i1}, r_{1j}, \dots, r_{is}, r_{sj}$ 表示。例如, 在 Advogato 数据集[13]中, 用户 i 和 j 之间的信任关系由四个等级表示: 1, 2, 3, 4, 即用户 i 到 j 的信任评级 $r_{ij} \in [1, 2, 3, 4]$ 。计算用户 i 对 j 间接信任的过程可以分解为 3 个步骤。下面, 将根据图 2 描述 ML-ELM-WalkNet 每个步骤。

步骤 1. 从数据集上检索出从 i 到 j 二跳信任路径上的信任 $r_{i1}, r_{1j}, \dots, r_{is}, r_{sj}$ 并组合构成向量 $\mathbf{x}_{ij} \in \mathbb{R}^{2m}$, 然后将向量 \mathbf{x}_{ij} 处理成维数固定的向量 $\mathbf{x}_{ij} \in \mathbb{R}^d$ (这里 d 作为模型的超参数)再作为神经网络的一个输入数据。

步骤 2. 使用 ML-ELM 来构建多隐层前馈神经网络, 并确定神经网络的隐藏层的数量 f , 以及每个隐藏层节点的个数 L_p 和激活函数 $g_p(\cdot)$, 然后将 \mathbf{x}_{ij} 输入到多隐层前馈神经网络中, 得到输出层的输出向量 $\mathbf{t}_{ij} \in \mathbb{R}^{d_s}$ (这里 d_s 表示数据集中可能的信任级别的数量)。

步骤 3. 使用 softmax 函数将获得的输出层的输出向量 \mathbf{t}_{ij} 转换为信任评级 \bar{r}_{ij} 。

由于不同的两个用户之间的存在的二跳信任路径的个数不一定相同, 这就使得不同的两个用户在步骤 1 中生成的 \mathbf{x}_{ij} 的维数不一定相同, 由于神经网络中的每一条输入数据的维数需要是相同的, 所以把向量 \mathbf{x}_{ij} 通过填充 0 的方式处理成维数相同的向量 $\mathbf{x}_{ij} \in \mathbb{R}^d$ 。

在第 3 步中, ML-ELM-WalkNet 将神经网络的输出向量 \mathbf{t}_{ij} 变换为标量信任值 \bar{r}_{ij} (为了区别原始信任值, 这里将计算得出的信任值用 \bar{r}_{ij} 表示)。该变换通过一个 softmax 函数实现, softmax 函数是逻辑函数的推广, 它将多维向量压扁到特定标量(信任评级)上。给定一个神经网络的输出向量 \mathbf{t}_{ij} , 它对应于评级 k 的概率可以表示为:

$$P(\bar{r}_{ij} = k | \mathbf{t}_{ij}) = \frac{(\mathbf{t}_{ij})_k}{\sum_1^{d_s} \exp(\mathbf{t}_{ij})} \quad (1)$$

其中 \bar{r}_{ij} 是推断出的信任评级, $(\mathbf{t}_{ij})_k$ 表示向量 \mathbf{t}_{ij} 的第 k 个分量的值。这一步的输出将所有可能的评级中概

率最大的评级认为是 i 对 j 的推断信任。

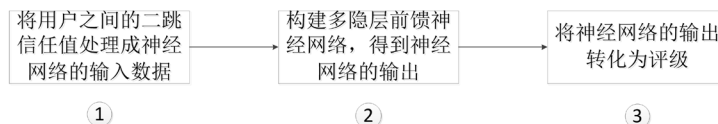


Figure 2. ML-ELM-WalkNet architecture
图 2. ML-ELM-WalkNet 的体系结构

4. ML-ELM-NeuralWalk

采用 ML-ELM-NeuralWalk 算法遍历 TSN，并迭代调用 ML-ELM-WalkNet 来计算用户之间的间接信任。在本节中，将详细描述 ML-ELM-NeuralWalk 算法。首先，对该算法如何迭代调用 ML-ELM-WalkNet 进行了概述。其次，给出了 ML-ELM-NeuralWalk 算法的伪代码，并进行了详细的分析。

4.1. ML-ELM-NeuralWalk 算法

ML-ELM-NeuralWalk 通过迭代的使用 ML-ELM-WalkNet 计算用户之间的间接信任，每次迭代都会更新 TSN，把第 q 次迭代更新所得到的 TSN 表示为 $G^{(q)}$ ，使用 $G^{(0)}$ 表示原始 TSN。为了方便起见，使用 $N_{in}(i)$ 和 $N_{out}(i)$ 分别表示节点 i 的内邻居和外邻居。 i 的内邻居是对节点 i 有直接信任的节点， i 的外邻居是节点 i 直接信任的节点。

在第 q 次迭代中，对于每个用户 i ，算法从 i 开始，进行广度优先搜索，找到用户 i 所有的外邻居 U ，即，用户 i 的 1 跳邻居。如果 $q=0$ ，则 i 对 $u \in U$ 的信任评级 $r_{iu}^{(0)}$ ，仅为原始信任评级，因为 $r_{iu}^{(0)}$ 来自 $G^{(0)}$ 。如果 $q \neq 0$ ，则 i 对 $u \in U$ 的信任评级 $r_{iu}^{(q)}$ 可能是原始的信任评级，也可能是 ML-ELM-NeuralWalk 算法迭代使用 ML-ELM-WalkNet 计算出的间接的信任评级。基于用户 i 所有的外邻居 U ，ML-ELM-NeuralWalk 在每个 $u \in U$ 上再进行一级广度优先搜索，到达距离 $G^{(q)}$ 中用户 i 最多 2 跳的节点。将这些节点的集合表示为 J 。ML-ELM-NeuralWalk，从 i, U, J 中收集 ML-ELM-WalkNet 的训练和推断样本。具体来说，ML-ELM-NeuralWalk 根据用户 i 到用户集合 J 的最短距离，分成 3 种情况处理 J 中的节点：

情况 1. 如果节点 $j \in J \cap i$ (即用户 i 到用户 j 的最短距离为 0 跳)，如图 3(a)所示，这时 ML-ELM-NeuralWalk 什么都不做。

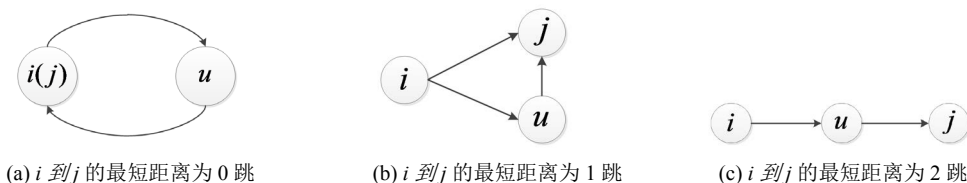


Figure 3. Three cases of the shortest distance from user i to user set J
图 3. 用户 i 到用户集合 J 的最短距离的 3 种情况

情况 2. 如果节点 $j \in J \cap U$ (即用户 i 到用户 j 的最短距离为 1 跳)且 $r_{ij}^{(q)} = r_{ij}^{(0)} \in G^{(0)}$ ， $r_{ij}^{(q)}$ 是原始图中的原始信任关系，如图 3(b)所示。这时 ML-ELM-NeuralWalk 收集从 i 到 j 二跳路径上的信任，即 $(r_{iu}^{(q)}, r_{uj}^{(q)})$ ，把 $r_{ij}^{(0)}$ 作为标签，形成训练样本：

$$\{(r_{iu}^{(q)}, r_{uj}^{(q)}) \forall u \in N_{out}(i) \cap N_{in}(j), r_{ij}^{(0)}\} \tag{2}$$

情况 3. 如果节点 $j \in J$ 且 $j \notin U$ (即用户 i 到用户 j 的最短距离为 2 跳)，如图 3(c)所示。这时

ML-ELM-NeuralWalk 收集从 i 到 j 二跳路径上的信任, 即 $(r_{iu}^{(q)}, r_{uj}^{(q)})$, 去形成推断样本:

$$\left\{ \left(r_{iu}^{(q)}, r_{uj}^{(q)} \right) \forall u \in N_{out}(i) \cap N_{in}(j) \right\} \quad (3)$$

ML-ELM-NeuralWalk 通过遍历所有节点 $i \in G^{(q)}$, 将得到当前迭代的训练集 $T^{(q)}$ 和推断集 $I^{(q)}$ 。使用 $T^{(q)}$ 去训练当前迭代的 ML-ELM-WalkNet, 然后通过 ML-ELM-WalkNet 计算出每个推断样本 $\left\{ \left(r_{iu}^{(q)}, r_{uj}^{(q)} \right) \forall u \in N_{out}(i) \cap N_{in}(j) \right\} \in I^{(q)}$ 的间接信任评级 $\overline{r_{ij}^{(q)}}$ 。然后, 将计算出的所有间接信任评级添加到 $G^{(q)}$ 中, 得到 $G^{(q+1)}$, 算法进入下一个迭代, 当 $I^{(q+1)}$ 为空时, 表明 TSN 中的所有间接信任评级都已经计算出来, 算法结束。

4.2. ML-ELM-NeuralWalk 算法

算法 1 给出了 ML-ELM-NeuralWalk 的伪代码。在第 1 行, 对迭代次数 q 初始化为 0。在第 2 行, 判断是否超出最大迭代次数。在第 3 行, 每次迭代初始化训练集 $T^{(q)}$ 和推断集 $I^{(q)}$ 为空。从第 5 行到第 7 行, 通过收集每个节点 $i \in V$ 的训练样本 $T_i^{(q)}$ 和推断样本 $I_i^{(q)}$ 产生训练集 $T^{(q)}$ 和推断集 $I^{(q)}$ 。在第 9 行, 判断推断集 $I^{(q)}$ 是否为空, 如果推断集为空, 表示所有的间接信任评级都已经被计算出来, 此时算法结束。在第 10 行中, 用 $T^{(q)}$ 训练 ML-ELM-WalkNet。在第 11 行中, 用训练好的 ML-ELM-WalkNet 去推断 $I^{(q)}$, 得到推断集中的间接信任评级。在第 12 行中, 用计算出的信任评级去更新 $G^{(q)}$ 得到 $G^{(q+1)}$ 。在第 16 行中, 表示完成了 1 次迭代, 迭代次数加 1。在第 18 行中, 表示到达了最大迭代次数或者所有的间接信任评级都已经被计算出来, 得到了最终的信任社交网络图 $G^{(q)}$ 。

算法 1 ML-ELM-NeuralWalk (G, V, Q)

输入: 有向图 $G^{(0)}$, 节点集 V , 最大迭代次数 Q
输出: i 对 j 的信任评级, $\forall i, j \in V, i \neq j$ (i, j 之间至少要有一条路径)

```

1:  $q \leftarrow 0$ 
2: while  $q < Q$  do
3:  $T^{(q)} \leftarrow \{ \}, I^{(q)} \leftarrow \{ \}$ 
4: for all  $i \in V$  do
5:     从  $G^{(q)}$  中生成  $T_i^{(q)}$  和  $I_i^{(q)}$ 
6:      $T^{(q)} \leftarrow T^{(q)} \cup T_i^{(q)}$ 
7:      $I^{(q)} \leftarrow I^{(q)} \cup I_i^{(q)}$ 
8: end for
9: if  $I^{(q)} = \{ \}$  do
10:    用  $T^{(q)}$  训练 ML-ELM-WalkNet
11:    用训练好的 ML-ELM-WalkNet 去推断  $I^{(q)}$ 
12:    用推断结果更新  $G^{(q)}$  得到  $G^{(q+1)}$ 
13: end if
14: else do
15:    break
16:     $q \leftarrow q + 1$ 
17: end while
18: return  $G^{(q)}$ 

```

5. 实验和结果分析

为了验证本文提出的 ML-ELM-NeuralWalk 在信任评估中的优越表现, 对算法的准确性和速度进行了一系列实验, 并与现有的最先进的信任评估算法 NeuralWalk [11]、OpinionWalk [8]、MoleTrust [5] 和 TidalTrust [4] 进行了比较。

5.1. 数据集

本文采用了两种常用的信任评估真实数据集 Advogato [13]和 Pretty Good Privacy (PGP) [14]。数据集 Advogato 是从一个在线软件开发社区收集的，其中一个用户对另一个用户的信任程度代表了用户对另一个用户的软件开发实力的态度，用户之间的信任程度分为四个级别。数据集 PGP 来自公钥认证网络。在该数据集中，一个用户对另一个用户的信任程度表示该用户认证了另一个用户的可信度，用户之间的信任程度也被划分为四个级别。表 1 汇总了两个数据集的统计数据。

Table 1. Statistical overview of Advogato and PGP datasets

表 1. Advogato 和 PGP 数据集的统计汇总

数据集	边的数量	节点的数量
Advogato	51372	5420
PGP	100000	26812

5.2. 实验设置

ML-ELM-NeuralWalk 的超参数包括：3.1 节中神经网络输入数据 \mathbf{x}_i 的维度 d 、神经网络的隐藏层的数量 f 以及每个隐藏层节点的个数 L_p 和激活函数 $g_p(\cdot)$ ，4.2 节中算法的最大迭代次数 Q 。

当算法的迭代次数大于等于 1 时，为了使形成的训练集更准确，将 $G^{(0)}$ 中的用户 i 到用户 j 之间的二跳信任路径作为构成 \mathbf{x}_i 的优先选择。因为 Advogato 和 PGP 中两个节点间的二跳信任路径的条数大多数小于 32 因此，设 $d = 64$ 。在进行信任计算时，如果 $G^{(0)}$ 中用户 i 到用户 j 之间的二跳信任路径的条数小于 32，则随机选择计算出的间接信任组成的路径作为构成 \mathbf{x}_i 的补充，如果此时 \mathbf{x}_i 的维数仍然小于 64，则剩余位置填充为零。如果 $G^{(0)}$ 中用户 i 到用户 j 之间的二跳信任路径的条数大于 32，则随机选择 32 条信任路径来组成 \mathbf{x}_i 。设置每个隐藏层节点的个数相同，即 $L_p = L_{p+1}$ ，每个隐藏层激活函数 $g_p(\cdot)$ 都选择 $\text{ReLU}(x) = \max(0, x)$ 。

从图 4 中可以发现，在 Advogato 数据集中，当神经网络的隐藏层的数量 $f = 2$ 时，ML-ELM-NeuralWalk 的 F1 得分最高，因此设 $f = 2$ 。在隐藏层的数量 $f = 2$ 时，当隐藏层节点的个数 L_p 的大小从 100 增长到 1800 时，ML-ELM-NeuralWalk 算法的 F1 分数在逐步提升，当 $L_p \geq 1800$ 时，ML-ELM-NeuralWalk 算法的 F1 得分没有明显的变化，且随着隐藏层节点数目的增加，算法的速度逐渐下降，在综合考虑算法的准确性和算法的速度之后，设 Advogato 数据集中的 $L_p = 1800$ 。

从图 5 中可以看出，在 PGP 数据集中，当神经网络的隐藏层的数量 $f = 2$ 时，ML-ELM-NeuralWalk 的 F1 得分最高，因此设 $f = 2$ 。在隐藏层的数量 $f = 2$ 时，当隐藏层节点的个数 L_p 从 50 增长到 800 时，ML-ELM-NeuralWalk 的 F1 得分有较大提升，当 $L_p \geq 800$ 时，EW 的 F1 分数趋于稳定，在同样的综合考虑算法的准确性和算法的速度之后，设 PGP 数据集中的 $L_p = 800$ 。根据六度分离理论，设 EW 算法的最大搜索深度为 6，又因为当 ML-ELM-NeuralWalk 算法的迭代次数为 3 时算法的最大搜索深度为 6，因此设 $Q = 3$ 。

对于 ML-ELM-NeuralWalk 算法，训练集分为两部分：一个用于训练，另一个用于验证。训练数据和验证数据的比例分别为 80% 和 20%。整个 ML-ELM-NeuralWalk 算法是使用 python 3.6 实现的。

实验的计算机采用英特尔酷睿 i7-8750H、2.20 GHz CPU、8 G 内存和 128 G SSD 硬盘。

5.3. 准确性

使用机器学习算法中评价准确性常用的指标——F1 分数、平均绝对误差(Mean absolute error, MAE)来

作为本文评价信任计算准确性的指标。由于 Advogato 和 PGP 数据集中的信任值 r_{ij} 是用 1, 2, 3, 4 四个序数表示的, 而 TidalTrust、MoleTrust 和 OpinionWalk 算法需要信任值的范围在 0 到 1 之间, 因此将 Advogato 和 PGP 数据集中的信任表示方式进行转换, 如表 2 所示。TidalTrust、MoleTrust 和 OpinionWalk 算法计算的是连续信任值, 而 ML-ELM-NeuralWalk 和 NeuralWalk 算法计算的是分类信任值, 为了方便进行比较, TidalTrust、MoleTrust 和 OpinionWalk 算法计算出来信任值后, 再计算与分类信任值的距离, 选择距离最近的分类信任值作为其分类信任值。

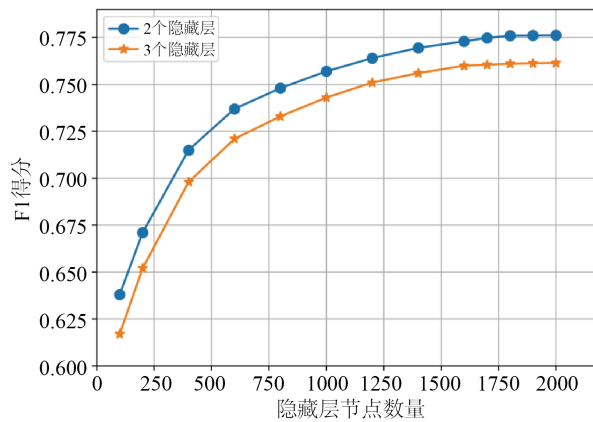


Figure 4. Trust calculation accuracy of ML-ELM-NeuralWalk algorithm on Advogato

图 4. ML-ELM-NeuralWalk 算法在 Advogato 上的信任计算准确性

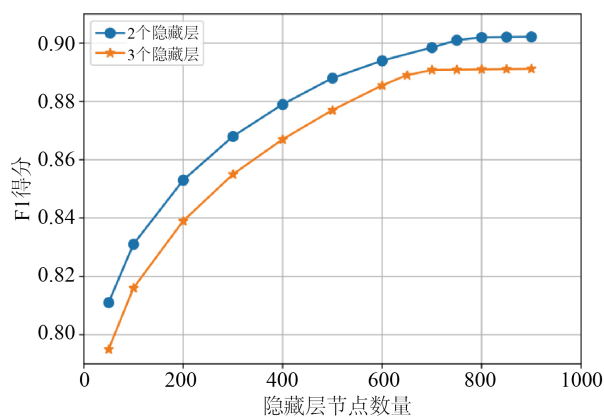


Figure 5. Trust calculation accuracy of ML-ELM-NeuralWalk algorithm on PGP

图 5. ML-ELM-NeuralWalk 算法在 PGP 上的信任计算准确性

Table 2. Transformation of the representation of trust values in a dataset

表 2. 数据集中信任值表示方式的转换

数据集	信任评级	分类信任值
Advogato/PGP	1	0.1
	2	0.4
	3	0.7
	4	0.9

在 Advogato 数据集中，对 TidalTrust、MoleTrust、OpinionWalk、NeuralWalk 和 ML-ELM-NeuralWalk 算法的信任计算准确性进行了比较。从图 6 中可以发现，ML-ELM-NeuralWalk 有最高的 F1 得分和最低的 MAE，体现了 ML-ELM-NeuralWalk 算法在信任计算准确性方面的优越性。

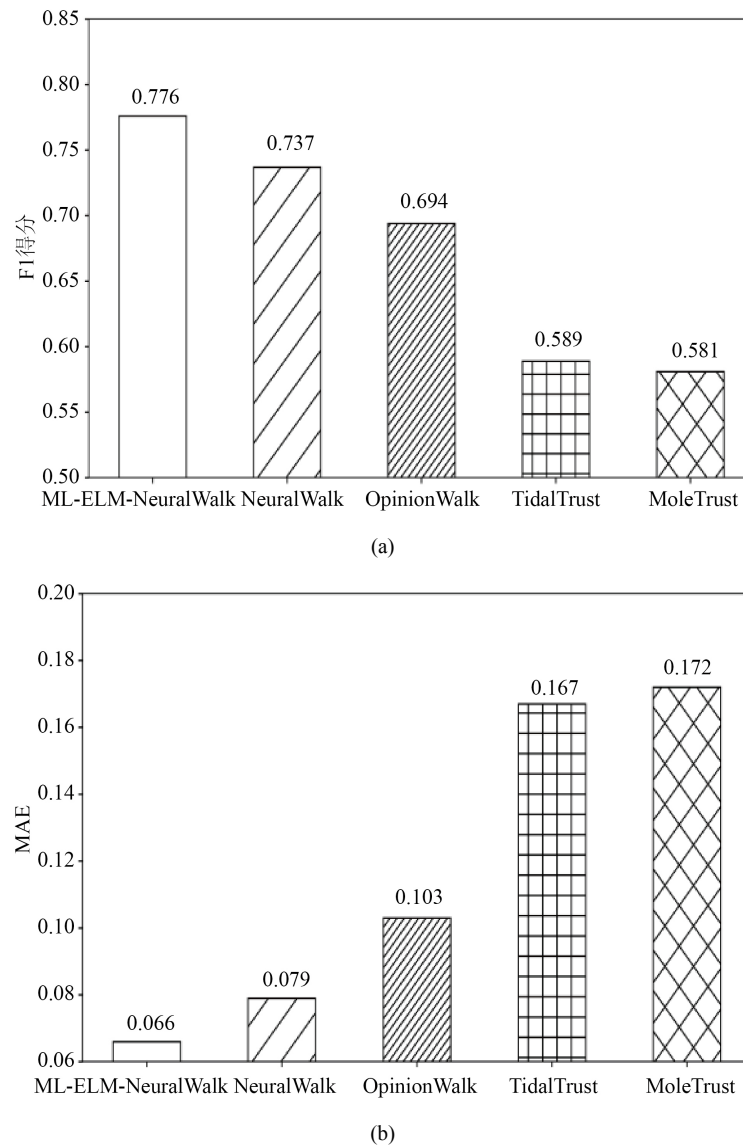


Figure 6. The trust calculation accuracy of different algorithms on Advogato
图 6. 不同算法在 Advogato 上信任计算的准确性

为了验证 ML-ELM-NeuralWalk 算法在信任计算准确性方面的优越表现不依赖于数据集，使用 PGP 数据集分别对 ML-ELM-NeuralWalk、NeuralWalk、OpinionWalk、MoleTrust 和 TidalTrust 算法的信任计算准确性进行评估。ML-ELM-NeuralWalk 的 F1 得分最高为 0.902，比第二优的算法 NeuralWalk 的 F1 分数 0.866 高出 0.036，MoleTrust 的 F1 分数最低为 0.607。通过观察图 7(b)可以发现 ML-ELM-NeuralWalk 算法在 MAE 上的表现仍然最好为 0.063，NeuralWalk 算法的 MAE 为 0.077，MoleTrust 算法的 MAE 最高为 0.209。

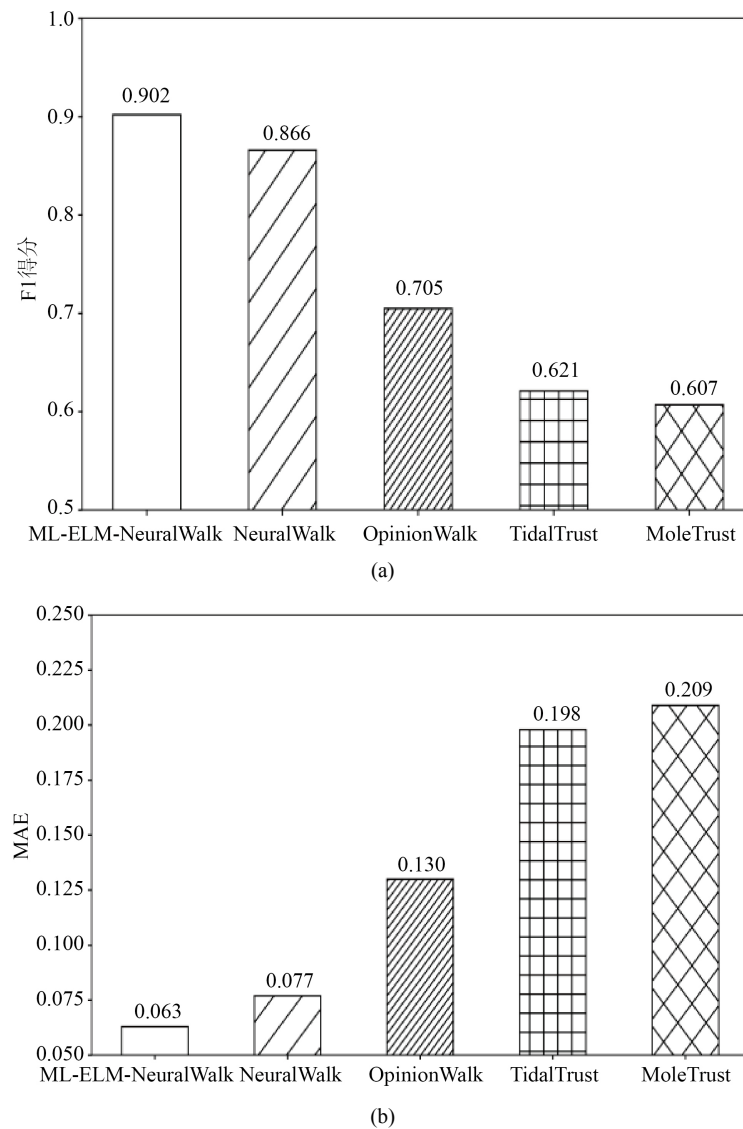


Figure 7. The trust calculation accuracy of different algorithms on PGP
图 7. 不同算法在 PGP 上信任计算的准确性

以上比较结果表明，在 Advogato 和 PGP 数据集中，基于归纳的信任评估方法在信任计算准确性方面优于基于推断的信任评估方法，且 ML-ELM-NeuralWalk 算法在信任计算的准确性方面则明显优于现有的信任计算方法。主要原因是基于机器学习的归纳信任评估方法可以实现更准确的信任计算且 ML-ELM-NeuralWalk 算法在信任计算准确性方面继承了 ML-ELM 对于分类问题有较高准确性的特性。

5.4. 速度

在 TSN 中，由于存在大量的用户，这就使得用户之间的信任关系变得非常的庞大，所以信任计算的算法必须具有高效性。由于神经网络的训练过程是比较耗时的，而 ML-ELM-NeuralWalk 和 NeuralWalk 算法都需要进行训练，因此观察这两种算法进行训练时的速度。在 Advogato 数据集中，ML-ELM-NeuralWalk 和 NeuralWalk 的训练时间分别为 32716.4 s 和 114921.6 s。在 PGP 数据集中，ML-ELM-NeuralWalk 和 NeuralWalk 的训练时间分别为 36617.1s 和 197632.4 s。从以上实验可以得出结论，

在 Advogato 和 PGP 数据集中, ML-ELM-NeuralWalk 的训练速度都比 NeuralWalk 的训练速度快得多。

6. 总结

本文将在线社交网络中的信任评估问题转化为分类问题, 利用 ML-ELM 在分类问题上的优越表现提出了一种使用神经网络的信任评估算法 ML-ELM-NeuralWalk。ML-ELM-NeuralWalk 是一种迭代的方法, 采用 ML-ELM-WalkNet 的神经网络架构来学习当前迭代中在线社交网络中用户之间的二跳信任传递和信任聚合的规则, 并用 ML-ELM-WalkNet 学到的规则来计算当前迭代中用户之间的二跳信任, 将计算的结果添加到在线社交网络中, 随着迭代地进行, ML-ELM-NeuralWalk 通过迭代的使用 ML-ELM-WalkNet 就实现了用户之间的多跳信任计算。通过两个真实的 OSN 数据集上进行的实验表明, ML-ELM-NeuralWalk 的性能优于现有的解决方案。在未来的研究中, 我们计划将 ML-ELM-NeuralWalk 应用于大型的社交商务平台中, 以测试实际的应用效果。

参考文献

- [1] Xu, J., Huang, F., Zhang, X., *et al.* (2019) Visual-Textual Sentiment Classification with Bi-Directional Multi-Level Attention Networks. *Knowledge-Based Systems*, **178**, 61-73. <https://doi.org/10.1016/j.knosys.2019.04.018>
- [2] Xu, K., Zheng, X., Cai, Y., *et al.* (2018) Improving User Recommendation by Extracting Social Topics and Interest Topics of Users in Uni-Directional Social Networks. *Knowledge-Based Systems*, **140**, 120-133. <https://doi.org/10.1016/j.knosys.2017.10.031>
- [3] Forsati, R., Mahdavi, M., Shamsfard, M., *et al.* Matrix factorization with explicit trust and distrust side information for improved social recommendation. *ACM Transactions on Information Systems (TOIS)*, 2014, 32(4): 17. <https://doi.org/10.1145/2641564>
- [4] Jennifer, G., James, H., *et al.* (2006) Filmtrust: Movie Recommendations Using Trust in Web-Based Social Networks. *IEEE CCNC*, 282-286.
- [5] Paolo, M. and Paolo, A. (2005) Controversial Users Demand Local Trust Metrics: An Experimental Study on epinions.com Community. *Proceedings of the National Conference on Artificial Intelligence*, **20**, 121.
- [6] Hao, F., Min, G., Lin, M., *et al.* (2013) MobiFuzzyTrust: An Efficient Fuzzy Trust Inference Mechanism in Mobile Social Networks. *IEEE Transactions on Parallel and Distributed Systems*, **25**, 2944-2955. <https://doi.org/10.1109/TPDS.2013.309>
- [7] Hamdi, S., Gancarski, A.L., Bouzeghoub A., *et al.* (2016) TISoN: Trust Inference in Trust-Oriented Social Networks. *ACM Transactions on Information Systems (TOIS)*, **34**, 17. <https://doi.org/10.1145/2858791>
- [8] Liu, G., Chen, Q., Yang, Q., *et al.* (2017) Opinionwalk: An Efficient Solution to Massive Trust Assessment in Online Social Networks. *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, 1-9. <https://doi.org/10.1109/INFOCOM.2017.8057106>
- [9] Mao, C., Xu, C. and He, Q. (2019) A Cost-Effective Algorithm for Inferring the Trust between Two Individuals in Social Networks. *Knowledge-Based Systems*, **164**, 122-138. <https://doi.org/10.1016/j.knosys.2018.10.027>
- [10] 童向荣, 张伟, 龙宇. Agent 主观信任的传递性[J]. 软件学报, 2012(11): 2862-2870.
- [11] Liu, G., Li, C. and Yang, Q. (2019) NeuralWalk: Trust Assessment in Online Social Networks with Neural Networks. *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, Paris, 29 April-2 May 2019. <https://doi.org/10.1109/INFOCOM.2019.8737469>
- [12] Kasun, L.L.C., Zhou, H., Huang, G.B., *et al.* (2013) Representational Learning with ELMs for Big Data. *Intelligent Systems, IEEE*, **28**, 31-34.
- [13] Raph, L. (2014) Advogato. <http://www.advogato.org/>
- [14] Hang, C.W., Wang, Y. and Munindar, P.S. (2009) Operators for Propagating Trust and Their Evaluation in Social Networks. *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*, 1025-1032.