·先进计算与数据处理· 文章编号:1000-3428(2018)08-0068-06 文献标志码:A 中图分类号:TP393

一种针对大规模社交网络的用户信任度预测算法

张 琼,张 勇

(江西财经大学 软件与物联网工程学院,南昌 330013)

摘 要: 在社交网络信任度研究领域中,多数模型或算法仅适用于小规模网络,或在大规模网络中效率低下。为此,综合考虑网络中的节点拓扑结构和用户信任率信息,提出一种针对大规模社交网络的信任度预测算法。将大规模社交网络约简为一个信任图,在该信任图上计算用户间的信任度。其中,节点的拓扑结构信息可在线下计算。实验结果表明,与典型的信任度预测算法 Tidal Trust、SWTrust 相比,该算法具有较高的预测精度和计算效率。

关键词: 大规模社交网络;信任度预测;拓扑结构;用户信任率;信任图

中文引用格式: 张 琼,张 勇. 一种针对大规模社交网络的用户信任度预测算法[J]. 计算机工程,2018,44(8):68-73.

英文引用格式: ZHANG Qiong, ZHANG Yong. A user trust degree prediction algorithm for large-scale social network[J]. Computer Engineering, 2018, 44(8):68-73.

A User Trust Degree Prediction Algorithm for Large-scale Social Network

ZHANG Qiong, ZHANG Yong

(School of Software and Communication Engineering, Jiangxi University of Finance and Economics, Nanchang 330013, China)

【Abstract】 At present, in the research field of social network trust degree prediction, most proposed models or algorithms just are suitable for small-scale network, or have bad efficiency in large-scale network. To solve this problem, considering the node topological structure information and the user trust rate information in the network, a trust degree prediction algorithm for large-scale social network is proposed. The large-scale social network is reduced to a trust graph, and the trust degree between users is calculated on the trust graph. The topological structure information of nodes can be calculated under line. Experimental results show that, compared with the typical trust prediction algorithm Tidal Trust and SWTrust, the proposed algorithm has higher prediction accuracy and computational efficiency.

【Key words】 large-scale social network; trust degree prediction; topological structure; user trust rate; trust graph

DOI: 10.19678/j.issn.1000-3428.0049691

0 概述

随着计算机网络的快速发展,网络社交已经成为人们进行交流的主要方式之一[1]。近年来,社交网络中的用户量越来越庞大,网络结构也越来越复杂。如何快速准确地预测大规模社交网络中的用户信任度显得尤为重要。

通常情况下,信任是人们进行一切社交活动的基础[2]。同时,在实际中,信任被认为是社交网络中的一种特殊信息,其具有可传播性、弱传递性和不确定性等性质[3]。这些性质为研究预测社交网络中 2 个用户间的信任度提供了思路。例如:社交网络中的用户 A 与用户 B 素不相识,而在某个特定的环境下,A 想要知道 B 是否可以信任。虽然 A 与 B 之前没有任何联系,但此时 A 可以通过咨询他的朋友来搜集关于 B 的信任信息,并结合相关的信任推理方法最后对 B 做出信任度预测。

目前,社交网络信任预测和个性化评估已被广泛研究[4],且被应用于多种领域,如网络安全、社交服务推荐[5]、P2P 网络[6]、电子商务[7-8]、云计算[9]等。这些研究的主要思路为:首先,从社交网络中提取出相应的信任路径(信任图);然后,在该信任路径上采用一些典型的信任整合策略计算出用户的预测信任值,其中,基于图简化的信任度预测模型是研究较多的方式之一,较为典型的有 Tidal Trust[10]、Mole Trust[11] 和 GFTrust[12] 等算法。但是,这些算法通常只适用于小规模网络或在大规模网络中效率低下,且要求该网络中每条边都具有完整的信任关系。为解决这一问题,文献[13]建立一种基于“小世界”网络理论的信任图生成框架 SWTrust。该框架主要根据用户活动域信息来计算用户的信任相关度,并结合弱连接理论和社会距离来从大规模社交网络中提

基金项目: 国家自然科学基金(61762043,61562035);江西省研究生创新专项资金(YC2017-S226)。

作者简介: 张 琼(1993—),女,硕士研究生,主研方向为大数据、网络安全;张 勇(通信作者),副教授、博士。

收稿日期: 2017-12-13 **修回日期:** 2018-03-14 **E-mail:** zhangyong@jxufe.edu.cn

取出一个信任图,然后在该信任图的基础上利用8 种基于可靠性模型的信任整合策略计算出用户的预测信任值。但是,该方法所计算的用户信任度依赖信任的发起者和被评价的信任者(如 *source* 和 *target*),导致其计算效率存在局限性。

针对以上方法的不足,本文提出一种针对大规模社交网络的用户信任度预测算法 TTDTrust。综合考虑网络中节点的拓扑特征和用户信任率信息,并设计一种用户拓扑信任度指标来衡量用户信任信息。此外,该算法所计算的用户拓扑信任度独立于被评价的用户对,即计算网络中用户拓扑信任度的过程可在线下进行。最后,本文在真实的社交网络分析数据集上进行多次实验来验证该算法的有效性。

1　算法设计

1.1　相关定义

定义 1(信任)　信任在不同领域有不同定义。本文主要采用文献[10]对信任的定义,即信任为"一个用户的行为将带来好的结果"。此外,信任分为推荐信任和功能信任[4]。推荐信任是指一个用户为某个用户推荐另一个用户的信任值,功能信任是一个用户对另一个用户的直接信任值。

通常情况下,2 个用户间的信任有 2 种表示方式:1)二进制表示,即"1"表示信任,"0"表示不信任;2)将信任表示为[0,1]内的某一个数值[14],数值越大表示越信任。本文采用后者作为信任表示方法。

定义 2(信任网络)　对于一个给定的社交网络,将其建模表示为对应的信任网络 $G=(N,E,W)$。其中,N 代表社交网络中的所有用户,E 代表信任网络中的有向边集合,每一条有向边 $e(n_i,n_j)$ 表示用户 $n_i(n_i\in N,0<i<|N|)$ 对用户 $n_j(n_j\in N,0<j<|N|)$ 有信任关系,W 代表边上的权值集合,每一个权值 $t_{ij}(t_{ij}\in W)$ 表示 2 个用户的直接信任值。在本文中,t_{ij} 范围为[0,1],该值越大,表示信任关系越强。

定义 3(信任路径)　对于给定的信任网络 $G=(N,E,W)$,如果该网络中的一个源节点 *source* 到另一个目标节点 *target* 之间存在一条可达的路径 $p=(source,\cdots,n_i,n_j,\cdots,target)$,且 p 中任意边 $e(n_i,n_j)$ 上的权值 t_{ij} 大于信任阈值 θ,则定义该路径 p 为信任路径。

定义 4(信任图)　对于给定的信任网络 $G=(N,E,W)$,信任图由 *source* 到 *target* 间的所有信任路径构成。

1.2　算法框架

基于第 1.1 节的定义,本节进一步描述如何预测网络中源节点 *source* 对目标节点 *target* 的信任度。本文 TTDTrust 算法整体框架如图 1 所示。该算法旨在预测大规模社交网络中没有直接社交的 2 个用户之间的信任度,算法输入为原始大规模社交网络与 2 个待预测的用户 *source* 和 *target*。

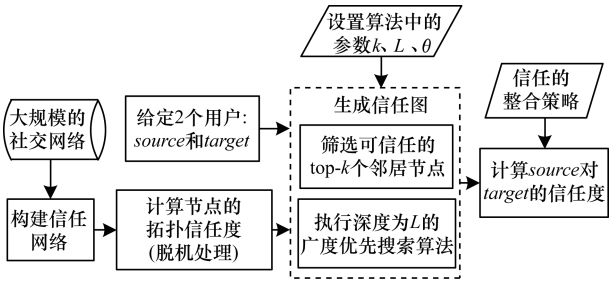


图 1　TTDTrust 算法整体框架

TTDTrust 算法步骤为:1)将大规模社交网络建模表示为信任网络;2)计算信任网络中节点的拓扑信任度(在线下进行计算);3)根据节点的拓扑信任度筛选该节点可信任的 top-k 个邻居;4)执行深度为 L 的广度优先搜索算法,生成 *source* 到 *target* 的信任图;5)在该信任图上采用信任整合策略计算 *source* 对 *target* 的信任度。

1.3　拓扑信任度

在社交网络分析中,通常利用节点度[15]、紧密度[16]和介数中心性[17]等度量来评估节点的信息传播能力。其中,节点度因设计简单而被广泛运用,但其难以充分度量节点的信息传播能力。相比之下,紧密度和介数中心性因考虑到节点与节点之间的信息,可以较好地体现节点的信息传播能力,但是其计算复杂度较高。因此,这 3 种度量方法均难以直接应用于社交网络信任度预测中。文献[18]通过考虑节点邻居度信息,提出用一种局部中心性指标来评估节点的信息传播能力。通常,在该指标考虑节点三步内邻居的入出度情况时,就可以达到较好的效果,且其计算过程只需在一步节点度的基础上进行即可。因此,该指标不仅具有较好的信息传播能力,而且具备节点度计算简单的特点。基于此,本文提出一种新的度量节点信任能力的指标:拓扑信任度,该指标综合考虑网络节点的拓扑特征和用户的信任率信息。节点拓扑信任度如图 2 所示。

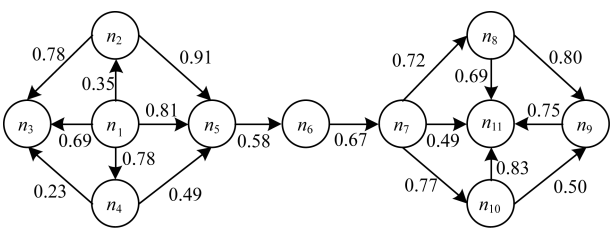


图 2　节点拓扑信任度示意图

定义 5(节点拓扑信任度)　图 2 中的一个信任网络 $G=(N,E,W)$ 共有 11 个节点,边上权值为 2 个用户的信任率,则定义节点 n_1 的拓扑信任度

$T_{\text{degree}}(n_1)$的计算公式如下：

$$T_{\text{degree}}(n_1)=\sum_{u\in\Gamma(n_1)}T_{\text{temp}}(u) \tag{1}$$

其中，$T_{\text{temp}}(u)=\sum_{\omega\in\Gamma(u)}R(\omega)$，$\Gamma(n_1)$和$\Gamma(u)$分别是节点$n_1$和$u$的一步邻居集合，$R(\omega)$是节点$\omega$的两步内所有邻居的信任率之和。则根据式(1)可计算出$R(n_1)=3.194\ 8$，$T_{\text{temp}}(n_1)=R(n_2)+R(n_3)+R(n_4)+R(n_5)=4.82$，$T_{\text{degree}}(n_1)=T_{\text{temp}}(n_2)+T_{\text{temp}}(n_3)+T_{\text{temp}}(n_4)+T_{\text{temp}}(n_5)=13.73$。

综上，对于一个给定的信任网络$G=(N,E,W)$和2个用户*source*、*target*，通过式(1)可以事先(脱机处理)计算出所有节点的拓扑信任度。

1.4 信任整合策略

通常情况下，信任整合包括2个步骤：

步骤1 针对信任图中的每条路径整合出一个信任值。

步骤2 聚合多条路径的信任值，即将步骤1中的所有信任值聚合成一个整体的信任度。

本文在实验过程中实现了基于可靠性模型[19]的4种信任整合策略，具体信息如表1所示。

表1 信任整合策略信息

序号	策略	信任传播	信任聚合	说明
1	Min-Max	Minimum	Maximum	最小值传播、最大值聚合策略
2	Min-WAve	Minimum	Weighted Average	最小值传播、加权平均聚合策略
3	Multi-Max	Multiplication	Maximum	乘积传播、最大值聚合策略
4	Multi-WAve	Multiplication	Weighted Average	乘积传播、加权平均聚合策略

1.5 TTDTrust 算法描述

TTDTrust算法主要针对一个给定的大规模社交网络和用户*source*、*target*，预测*source*对*target*的信任度。其伪代码如算法1所示。

算法1 TTDTrust

输入 信任网络G，源节点*source*，目标节点*target*，搜索广度上界k，搜索深度上界L，信任阈值θ

输出 *source*对*target*的信任度$t(source,target)$

1. Queue = {source}，tempQ = { }//初始化队列Queue为//source，tempQ为空
2. depth = 0//初始化搜索深度为0
3. S_{path} = { }//初始化信任路径集为空
4. while (Queue非空，且depth小于等于L) do
5. 从Queue中取出最前面的元素，并赋给v节点
6. 从v所有邻居中筛选出拓扑信任度最高的前top-k个邻居
7. for (v节点的top-k邻居中每一个节点u) do
8. 查询获取u的拓扑信任度T_{degree}(u)//事先通过式(1)//线下计算并存储T_{degree}(u)
9. if(u未访问，且T_{degree}(u)大于θ) then
10. if(u是target节点) then
11. 从target回溯source得到一条信任路径p
12. 将p加入到S_{path}中
13. else
14. 将u加入到tempQ中
15. 标记u已经被访问
16. end if
17. end if
18. end for
19. if (Queue为空) then
20. 将tempQ中的元素赋给Queue
21. depth增加1
22. 清空tempQ
23. end if
24. end while
25. 在S_{path}上分别利用4种信任整合策略计算出t(source，target)
26. return t(source，target)

TTDTrust算法具体流程如下：

步骤1 初始化数据结构与参数(第1行～第3行)。队列*Queue*和*tempQ*分别用于存储当前层访问的节点和下一层访问的节点，*depth*用于记录每次遍历的深度，$S_{\text{path}}=\{\}$用于存储最强信任路径集。

步骤2 执行具有限制条件的广度优先搜索算法(第4行～第24行)：1)只要队列*Queue*不为空，且搜索深度$depth\leqslant L$，就取出*Queue*中最前面的节点并赋给节点v；2)根据之前线下计算存储的节点拓扑信任度，从v的所有邻居中筛选出拓扑信任度最高的前top-k个邻居；3)遍历该top-k个邻居中的每个节点u，查询u的拓扑信任度$T_{\text{degree}}(u)$；4)对u进行判断，如果u没有被访问且$T_{\text{degree}}(u)$大于给定的信任阈值θ，则继续判断u是否为目标节点*target*，如果是目标节点，即从*target*回溯到*source*生成一条信任路径p，并把p添加到信任路径集S_{path}中，否则，标记u已经被访问，同时将u放入*tempQ*队列中；5)如果*Queue*为空，则将*tempQ*队列中的元素全部赋给*Queue*队列，同时将搜索深度*depth*加1，并清空*tempQ*队列。

步骤3 分别采用表1中的4种信任整合策略计算出*source*对*target*的预测信任度$t(source,target)$(第25行)。

1.6 算法复杂度分析

TTDTrust算法的计算时间主要包括线下计算时间和线上计算时间2个部分。

1)线下计算时间。线下计算过程主要为计算网络中所有用户的拓扑信任度。该过程遍历每个用户三步邻居的边，即需要扫描$3|E|$条边，因此，该过程时间复杂度为$O(|E|)$。

2)线上计算时间。线上计算包括2个部分：

(1)执行搜索宽度为 k、深度为 L 的广度优先搜索算法;(2)根据信任图(信任路径集)计算 *source* 对 *target* 的预测信任度。首先,为限制广度优先搜索算法的广度,采用查找排序算法筛选每个用户的 top-k 个邻居,其时间复杂度为 $O(|N|\cdot\log_a k)$;然后,执行广度优先搜索算法,该过程最差情况的时间复杂度为 $O(|V|+|E|)$。此外,在计算 *source* 对 *target* 预测信任度的过程中,需要扫描信任图中的每条路径和路径上的边权(信任率),又因为信任图中只有 $|S_{\text{path}}|$ 条信任路径,每条路径的边数最多不超过 L,且 $|S_{\text{path}}|$ 和 L 通常是一个较小的常数,所以该过程的时间复杂度为 $O(1)$。因此,线上计算过程的时间复杂度为 $O(|N|\cdot\log_a k)$。

综上,TTDTrust 算法的总时间复杂度为 $O(|E|+|n|\cdot\log_a k)$,线上计算的时间复杂度为 $O(|N|\cdot\log_a k)$,远小于传统的全遍历信任度预测算法的复杂度 $O(|V|\cdot|E|)$,尤其是在大型网络中,TTDTrust 算法的时间复杂度优势更明显。

2　实验结果与分析

2.1　数据集说明与实验设置

本文采用真实的社交网络数据集 Epinions[20] 进行实验,实验中舍弃不存在信任关系的用户和边,最后得到 7 375 个用户节点和 111 781 条边。考虑到数据集中的信任表示为二进制类型,在实验过程中,本文采用文献[14]方法,将用户二值信任关系转换为[0,1]范围内的信任表示。具体方法为:首先,为每个用户节点赋一个质量 $q_i\in[0,1]$,且 $q_i\sim N(\mu,\sigma^2)$;然后,从 $[\max(q_j-\delta_{ij},0),\min(q_j+\delta_{ij},1)]$ 范围内均匀地选取用户 n_i 对 n_j 的信任值 $t(n_i,n_j)$。其中,$\delta_{ij}=(1-q_i)/2$ 表示噪音系数,$t(n_i,n_j)$ 的范围为[0,1],其值越大表示 n_i 对 n_j 越信任。

本次实验使用的个人计算机配置为 64 位 win7 操作系统,处理器为 Intel(R) Core(TM) i5-2400 CPU@3.10 GHz RAM 8 G。实验参数如表 2 所示。

表 2　实验参数设置

参数名称	描述	参数值
k	限制搜索的宽度(top-k 邻居)	9
L	限制搜索的深度	6
θ	信任阈值	0.5
μ	生成用户质量中正态分布的均值	0.75
σ	生成用户质量中正态分布的标准差	0.25

实验过程采用留一法进行测试。实验次数 $\tau=10\,000$,每次实验随机地抽取一条边进行测试,将第 i 次实验中 2 个用户之间的实际信任度值记为 $t_{\text{real}}(i)$,预测信任度值记为 $t_{\text{pred}}(i)$。相应的信任度评估指标为平均绝对误差(*MAE*)、准确度(*Precision*)、召回率(*Recall*)和 *F* 值。各值计算公式为:

$$MAE=\frac{1}{\tau}\sum_{i=1}|t_{\text{real}}(i)-t_{\text{pred}}(i)| \tag{2}$$

$$Precision=|S_A\cap S_B|/S_B \tag{3}$$

$$Recall=|S_A\cap S_B|/S_A \tag{4}$$

$$F=2\times Precision\times Recall/(Precision+Recall) \tag{5}$$

其中,S_A、S_B 分别为实验随机抽取的边中 $t_{\text{real}}(i)$ 大于 θ 和 $t_{\text{pred}}(i)$ 大于 θ 的边集合。

2.2　算法性能分析

2.2.1　TTDTrust 算法精度

为验证 TTDTust 算法的有效性,分别运用 4 种典型信任整合策略测试 TTDTust 算法的信任预测精度,实验结果如表 3 所示。

表 3　TTDTrust 算法的信任预测精度

策略	*MAE*	*Precision*	*Recall*	*F*
Min-Max	0.148 2	0.921 4	0.848 7	0.883 5
Min-WAve	0.132 1	0.925 6	0.870 0	0.897 0
Multi-Max	0.147 6	0.919 0	0.850 0	0.883 1
Multi-WAve	0.136 4	0.924 5	0.865 0	0.893 8

由表 3 可以看出,在 4 种信任整合策略中,Min-WAve 的 *MAE*、*Precision*、*Recall* 和 *F* 值都优于其他 3 种策略,即 Min-WAve 预测效果最好。此外,从表 3 还可以看出,4 种策略的综合指标 *F* 最小值为 0.883 1,最大值为 0.897 0,该结果验证了 TTDTust 算法具有较高的信任预测精度。

2.2.2　TTDTrust 算法与典型预测算法的对比分析

在社交网络信任度预测研究领域中,Tidal Trust 算法[10] 较典型,且其具有很强的代表性。此外,文献[13]提出的 SWTrust 算法也是针对大规模社交网络进行用户信任度预测研究,且是近几年社交网络信任度预测模型中效果较好的算法之一。因此,本节实验通过 Min-WAve 策略,进行 TTDTrust 算法与 Tidal Trust 算法、SWTrust 算法的性能对比分析,实验结果如表 4所示。其中,*T* 表示算法运行时间。

表 4　3 种算法在 Min-WAve 策略下的性能对比分析

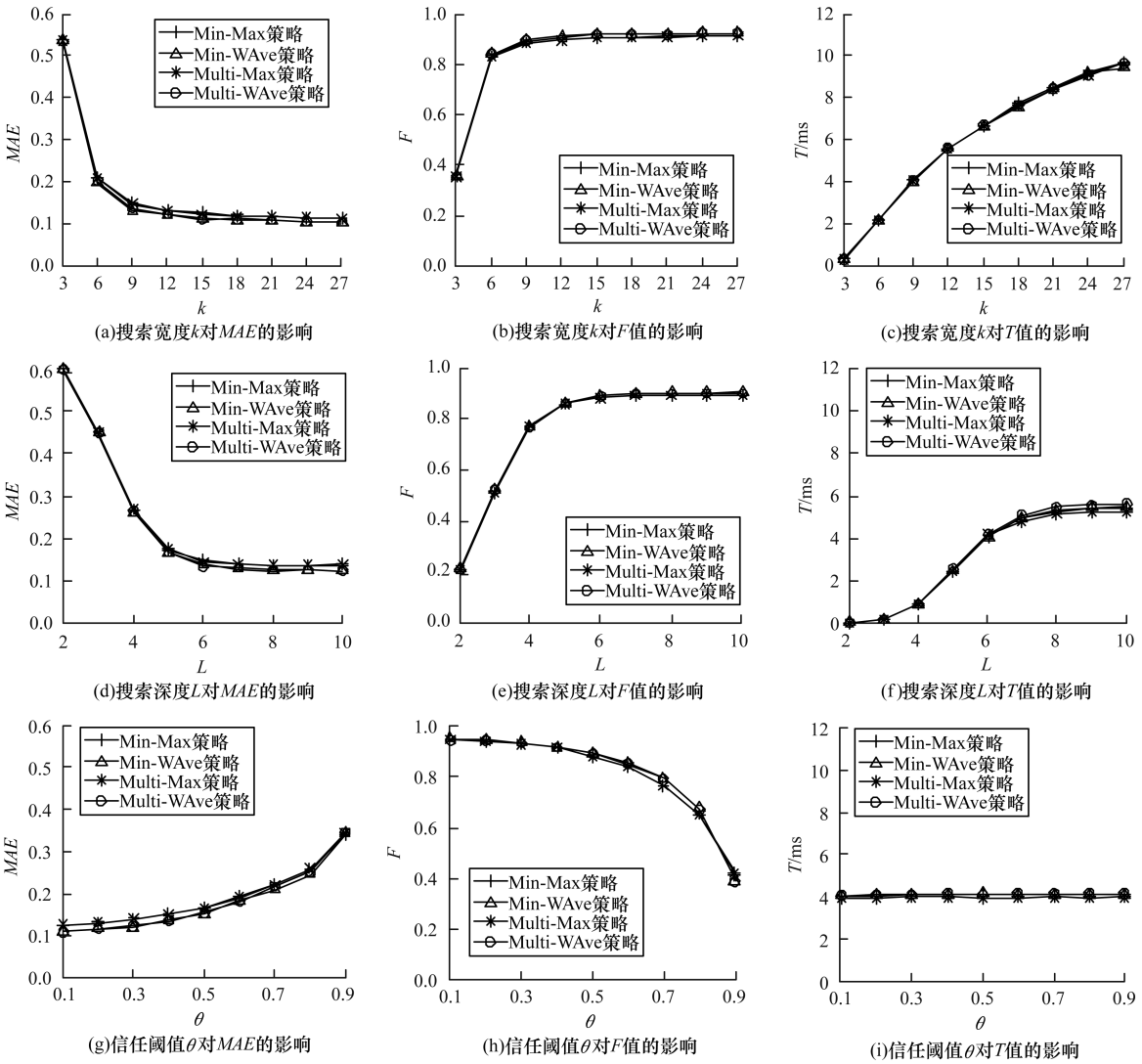
算法	预测精度评价指标				计算效率指标
	MAE	*Precision*	*Recall*	*F*	*T*/ms
Tidal Trust	0.187 7	0.911 4	0.793 4	0.848 3	26.603 9
SWTrust	0.172 3	0.922 5	0.803 5	0.858 9	24.212 0
TTDTrust	0.132 1	0.925 6	0.870 0	0.897 0	3.999 9

由表 4 可以看出,与 Tidal Trust、SWTrust 算法相比,TTDTrust 算法 *MAE* 值分别降低了 29.62% 和 23.33%,*Precision* 值分别提高了 1.55% 和 0.33%,*Recall* 值分别提高了 9.65% 和 8.27%,*F* 值分别提高了 5.74% 和 4.43%。此外,在计算效率上,因为 TTDTrust 算法的线上计算时间较少,所以其总体计

算时间大幅减少,Tidal Trust 算法的计算时间是 TTDTrust 算法的 6.65 倍,SWTrust 算法的计算时间是 TTDTrust 算法的 6.05 倍。这些结果均能说明 TTDTrust 算法针对大规模社交网络用户信任度预测时具有较好的效率。

2.2.3 参数 k、L、θ 对 TTDTrust 算法的影响

TTDTrust 算法涉及 3 个关键参数:k,L,θ。因此,本节实验分别测试参数 k、L、θ 对 TTDTrust 算法的影响,算法 *MAE* 值、*F* 值、*T* 值 3 个指标的实验结果如图 3 所示。其中,图 3(a) ~ 图 3(c)的实验参数为 $k \in [3,27]$、$L = 6$、$\theta = 0.5$;图 3(d) ~ 图 3(f)的实验参数为 $k = 9$、$L \in [2,10]$、$\theta = 0.5$;图 3(g) ~ 图 3(i)的实验参数为 $k = 9$、$L = 6$、$\theta \in [0.1,0.9]$。



(a)搜索宽度k对*MAE*的影响 (b)搜索宽度k对*F*值的影响 (c)搜索宽度k对*T*值的影响

(d)搜索深度L对*MAE*的影响 (e)搜索深度L对*F*值的影响 (f)搜索深度L对*T*值的影响

(g)信任阈值θ对*MAE*的影响 (h)信任阈值θ对*F*值的影响 (i)信任阈值θ对*T*值的影响

图 3 参数 k、L、θ 对 TTDTrust 算法的影响

由图 3(a) ~ 图 3(f)可以看出,在 $k \in [3,9]$、$L \in [2,6]$范围内,随着 k、L 的增大,TTDTrust 算法的 *MAE* 值逐渐减小,*F* 值、*T* 值均逐渐增加。随后,*MAE* 值和 *F* 值开始收敛,*T* 值增长放缓。此外,在图 3(c)中,*T* 值增长曲线的趋势与 log 函数基本相同。该结果与前文分析的 TTDTrust 算法时间复杂度为 $O(|N| \cdot \log_a k)$ 相吻合。

由图 3(g) ~ 图 3(i)可以看出,在 $\theta \in [0.1,0.5]$范围内,随着 θ 的增大,TTDTrust 算法的 *MAE* 值、*F* 值变化较小,而在 $\theta = 0.5$ 后,*MAE* 值显著增大、*F* 值显著减小。这说明信任阈值 θ 不能设置过大,原因是信任阈值太大会导致太多的路径被剔除,即信任阈值过大将导致 *source* 获得过少关于 *target* 的信息,以至于降低了 TTDTrust 算法的精度。此外,从图 3(i)可以看出,随着 θ 的增大,TTDTrust 算法的运行时间变化甚微。

综上可知,k、L 和 θ 对 TTDTrust 算法均有较大影响,尤其在 $k \in [3,9]$、$L \in [2,6]$、$\theta \in [0.5,0.9]$范围内影响较显著。从实验结果可以看出,在实际应用中取 $k = 9$、$L = 6$ 和 $\theta = 0.5$ 时,算法效果较好。

3　结束语

本文综合考虑网络中的节点拓扑结构信息和用户信任率信息,提出一种针对大规模社交网络的用户信任度预测算法 TTDTrust,并设计一种用户拓扑信任度指标来衡量用户信任信息。在真实的社交网络分析数据集上进行多方面的实验测试,结果表明,与典型算法 Tidal Trust、SWTrust 相比,TTDTrust 算法具有较高的信任预测精度和计算效率。由于实际社交网络往往具有较复杂的网络环境,且用户信任的影响因素较多,因此下一步将考虑多因素的信任建模并构建可应用于复杂网络环境的信任模型。

参考文献

[1] MEEKER M. Internet trends 2017 reports[EB/OL]. [2017-12-10]. http://www.kpcb.com/internet-trends.

[2] GOOD D. Individuals, interpersonal relations, and trust[EB/OL]. [2017-12-10]. http://pdfs.semanticscholar.org/cb13/6769a21cb774a960cba162039fc1368215da.pdf.

[3] LIU G, WANG Y, ORGUN M A. Trust transitivity in complex social networks[C]//Proceedings of AAAI Conference on Artificial Intelligence. [S.l.]: AAAI Press, 2011: 1222-1229.

[4] SHERCHAN W, NEPAL S, PARIS C. A survey of trust in social networks[J]. ACM Computing Surveys, 2013, 45(4): 1-33.

[5] 张　波,向　阳,黄震华. 一种社交网络中的个体间推荐信任度计算方法[J]. 南京航空航天大学学报, 2013, 45(4): 563-569.

[6] 林　军,姜文君,王国军. P2P 环境中基于信誉与云理论的信任模型[J]. 计算机工程, 2012, 38(2): 141-143.

[7] 甘早斌,丁　倩,李　开,等. 基于声誉的多维度信任计算算法[J]. 软件学报, 2011, 22(10): 2401-2411.

[8] 王　刚,桂小林. 社会网络中交易节点的选取及其信任关系计算[J]. 计算机学报, 2013, 36(2): 368-383.

[9] 谢晓兰,刘　亮,赵　鹏. 面向云计算基于双层激励和欺骗检测的信任模型[J]. 电子与信息学报, 2012, 34(4): 812-817.

[10] GOLBECK J. Computing and applying trust in Web-based social networks[D]. Maryland, USA: University of Maryland, 2005.

[11] MASSA P, AVESANI P. Trust metrics on controversial users: balancing between tytanny of the majority and echo chambers[J]. International Journal on Semantic Web and Information Systems, 2007, 3(1): 39-64.

[12] JIANG W, WU J, LI F, et al. Trust evaluation in online social networks using generalized network flow[J]. IEEE Transactions on Computers, 2016, 65(3): 952-963.

[13] JIANG W, WANG G, WU J. Generating trusted graphs for trust evaluation in online social networks[J]. Future Generation Computer Systems, 2014, 31(1): 48-58.

[14] RICHARDSON M, AGRAWAL R, DOMINGOS P. Trust management for the semantic Web[C]// Proceedings of the 2nd International Semantic Web Conference. Berlin, Germany: Springer, 2003: 351-368.

[15] STEPHENSON K, ZELEN M. Rethinking centrality: methods and examples[J]. Social Networks, 1989, 11(1): 1-37.

[16] BORGATTI S. Centrality and network flow[J]. Social Networks, 2005, 27(1): 55-71.

[17] POULIN R, BOILY M, MASSE B. Dynamical systems to define centrality in social networks[J]. Social Networks, 2000, 22(3): 187-220.

[18] CHEN D, LÜ L, SHANG M, et al. Identifying influential nodes in complex networks[J]. Physical A: Statistical Mechanics and Its Applications, 2012, 391(4): 1777-1787.

[19] 张　宇. 在线社会网络信任计算与挖掘分析中若干模型与算法研究[D]. 杭州:浙江大学, 2009.

[20] TANG J L. Trust/Distrust computing[EB/OL]. [2017-12-10]. http://www.cse.msu.edu/~tangjili//trust.html.

编辑　吴云芳

(上接第 67 页)

[14] SRIKANT R, AGRAWAL R. Mining sequential patterns: generalizations and performance improvements[C]// Proceedings of International Conference on Extending Database Technology. Berlin, Germany: Springer, 1996: 1-17.

[15] WANG Y, ZHENG Y, XUE Y. Travel time estimation of a path using sparse trajectories[C]//Proceedingds of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2014: 25-34.

[16] 郭　莎. 移动目标活动规律挖掘方法研究与设计[D]. 北京:北方工业大学, 2017.

[17] 陈　鹏. 基于用户移动数据的可视化用户行为分析[D]. 广州:华南理工大学, 2016.

[18] 张　巍,刘　峰,滕少华. 改进的 PrefixSpan 算法及其在序列模式挖掘中的应用[J]. 广东工业大学学报, 2013, 30(4): 49-54.

[19] AGRAWAL R, SRIKANT R. Fast algorithms for mining association rules in large databases[C]//Proceedings of IEEE International Conference on Software Engineering and Service Science. Washington D. C., USA: IEEE Press, 2014: 487-499.

[20] SRIKANT R, AGRAWAL R. Mining sequential patterns: generalizations and performance improvements[C]// Proceedings of International Conference on Extending Database Technology. Berlin, Germany: Springer, 1996: 3-17.

[21] ZAKI M J. SPADE: An efficient algorithm for mining frequent sequences[J]. Machine Learning, 2001, 42(1-2): 31-60.

[22] 吕　锋,张炜玮. 4 种序列模式挖掘算法的特性研究[J]. 武汉理工大学学报, 2006, 28(2): 57-60.

编辑　顾逸斐