软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2018,29(3):772−785 [doi: 10.13328/j.cnki.jos.005436]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

社交网络高效高精度去匿名化算法*

刘家霖, 史舒扬, 张悦眉, 邵蓥侠, 崔 斌

(高可信软件技术教育部重点实验室(北京大学),北京 100871)

通讯作者: 邵蓥侠, E-mail: shao.yingxia@pku.edu.cn

摘 要: 自从社交网络成为重要的研究课题,社交网络隐私保护也成为了重要的研究内容,尤其是关于公开发布以供研究的大规模社交网络图数据的隐私保护.为了评估用户的隐私风险,研究者们设计了不同的方法对图进行去匿名化,在不同的图网络中识别个体的身份.但是,当前的去匿名化算法或者需要高质量的种子匹配,或者在精确度和效率上颇有不足.提出一种高效高精度的无种子去匿名化算法 RoleMatch,基于社交网络的拓扑结构识别个体身份.该算法包括:(1) 可以快速计算的两图结点间相似度度量方法 RoleSim++;(2) 一种有效的结点匹配算法,此法同时考虑了结点间的相似度和中间匹配结果的反馈.在实验部分,利用 LiveJournal 的数据,用 RoleMatch 对比了多种流行的匿名化算法,并根据实际应用情景,在传统实验的基础上增加了局部去匿名化的实验,实验结果验证了所提出的去匿名化算法的优秀性能.

关键词: 社交网络;去匿名化;匿名化;隐私保护;结点相似度

中图法分类号: TP311

中文引用格式: 刘家霖,史舒扬,张悦眉,邵蓥侠,崔斌.社交网络高效高精度去匿名化算法.软件学报,2018,29(3):772−785. http://www.jos.org.cn/1000-9825/5436.htm

英文引用格式: Liu JL, Shi SY, Zhang YM, Shao YX, Cui B. Effective and efficient approach for graph de-anonymization. Ruan Jian Xue Bao/Journal of Software, 2018,29(3):772−785 (in Chinese). http://www.jos.org.cn/1000-9825/5436.htm

Effective and Efficient Approach for Graph De-Anonymization

LIU Jia-Lin, SHI Shu-Yang, ZHANG Yue-Mei, SHAO Ying-Xia, CUI Bin

(Key Laboratory of High Confidence Software Technologies, Ministry of Education (Peking University), Beijing 100871, China)

Abstract: Ever since social networks became the focus of a great number of researches, the privacy risks of published network data have also raised considerable concerns. To evaluate users' privacy risks, researchers have developed methods to de-anonymize graphs and identify same person in different graphs, yet the existing algorithms either requires high-quality seed mappings, or have low accuracy and high expense. In this paper, an effective and efficient seedless de-anonymization algorithm, "RoleMatch" is proposed. This algorithm is based on the network topology and consists of (1) a new cross-graph node similarity measurement "RoleSim++" with fast computation method, and (2) an effective node matching algorithm considering both similarities and feedbacks. In experiments, the algorithm is tested with graphs anonymized in several popular anonymization ways, using the data from LiveJournal. In addition to the traditional symmetric experiments, an asymmetric experiment setting is proposed to mimic closer to real-world application. The results from those experiment

* 基金项目: 国家自然科学基金(61572039); 中国博士后科学基金(2017M610020); 中国青年自然科学基金(61702015); 深圳市政府研究项目(JCYJ20151014093505032)

Foundation item: National Natural Science Foundation of China (61572039); China Postdoctoral Science Foundation (2017M610020); National Natural Science Foundation of China for Young Scholar (61702015); Shenzhen Goverment Research Project (JCYJ20151014093505032)

本文由基于图结构的大数据分析与管理技术专刊特约编辑林学民教授、杜小勇教授、李翠平教授推荐.

收稿时间: 2017-07-22; 修改时间: 2017-09-05; 采用时间: 2017-11-07; jos 在线出版时间: 2017-12-05

CNKI 网络优先出版: 2017-12-06 15:22:59, http://kns.cnki.net/kcms/detail/11.2560.TP.20171206.1522.001.html

show that with the proposed algorithm the de-anonymization work achieves superior performance compared with existing de-anonymization algorithms.

Key words: social network; de-anonymization; anonymization; privacy risk; node similarity

随着计算机领域的软硬件技术不断发展,社交网络逐渐地成为了线上产物.脸谱、推特、腾讯等公司都维护着庞大的社交网络图,并且这些图还在不断扩张.在这些社交网络图中,用户被表示为带有标签(例如姓名、性别、兴趣、地址等)的结点,而用户之间的交互活动可以被抽象为点之间的边(有向边或者无向边,具体根据交互活动的性质而定),通过这样的方式,社交网络就被表示为了带有必要的用户关系信息的图.

由于社交网络自身的真实性、规模性以及边的分布特点,吸引了很多的学术研究者和广告商的研究兴趣.然而,在公司、组织不断开放更多的信息过程中,用户的隐私很可能被迫揭露出来,导致用户身份泄漏.而随着社交网络图中结点的真实身份被揭露,会造成许多后果,例如接收到无穷无尽的垃圾广告信息,甚至因此错信他人而被诈骗.因此,发布的社交网络数据需要经过一定的匿名化处理,其中最简单的方法就是打乱图中点的标号.Backstrom 等人[1]提出了一种方法,利用少量手动创建的用户,建立一种特殊的模式,然后在发布的图中寻找该模式,就可以在发布的图中找到目标用户的对应身份.Wang 等人[2]提出了一种防止指纹攻击的匿名化方法,既能有效地阻碍从已知的公众人物入手的隐私攻击,又能较好地保持图的结构性质.

为了保护用户隐私,找到有效的匿名化方法,需要对匿名化方法的质量(破解的难易程度)进行评估,而这样的评估依赖于对去匿名化方法的探索.去匿名化,顾名思义就是匿名化的反过程,通过一些拓扑结构信息、标签信息对匿名化之后的图进行处理,找到各个点的原始身份信息.例如,在图 1 中,左边的网络是通过爬虫获得的,带有用户的公开信息(姓名);而右边的图是发布的,不带有用户的身份信息(姓名),但是有一些其他的信息(地址).去匿名化的过程可以理解为:匹配这两张图中的点,从而知道每一个人实际的地址.

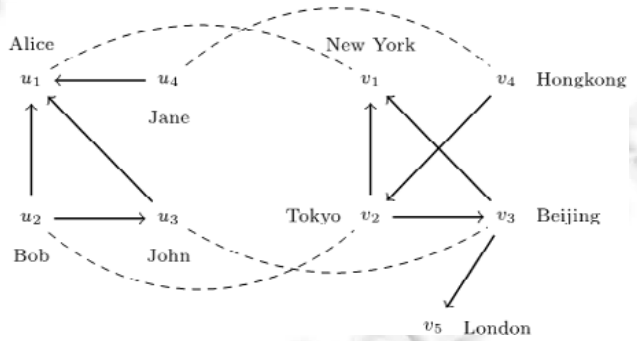


Fig.1 Crawled graph with public information (names) on the left and anonymized graph with secret information (locations) on the right

图 1 左边是附有公开信息(名字)的爬取的图,右边是匿名化之后的图,包含地址信息

近年来,针对去匿名化问题,研究者们相继提出了多种方法,根据需要提供的输入和利用的信息,主要可以将它们归为两类方法.

- 第 1 类方法是扩散性质的,需要一些已知身份信息的匹配点对,作为种子对,从它们出发不断扩展到其他的结点.由于已经有了扩散的中心,在初始中心正确的情况下,比较容易得到一些高质量的匹配对.但是这类方法的缺陷在于:往往对种子对的可靠性非常依赖,而该可靠性在实际操作中往往并不能保证,因此造成了相当的匹配错误率;
- 第 2 类方法不需要种子对,将各种结点对放在一个相对平等的初始地位上,然后通过对图结构的分析匹配并获得身份结果.这类方法主要利用的是结点的结构特征信息(例如度数、子图、结点相似度、描述性信息等),这些特征信息如果利用得好,可以得到高质量的匹配对.而由于信息多而杂,表示方法多

样,不方便进行统一度量,度量之后也难以很好地加以利用,这类方法主要的困难在于对特征的提取以及利用,这也是笔者在设计算法的时候主要考虑的内容.

本文提出了一种不需要种子的去匿名化方法 RoleMatch,它属于上述第 2 类方法,不需要初始的正确种子对,只基于图的结构信息.具体来说,只是利用图的拓扑结构信息,将每个可能点对的匹配程度用一个数值进行抽象,从而完成了从复杂多维空间到简单一维空间的有效映射,便于进行比较判断,解决了图结构信息多而杂的问题.在此基础上,该方法对此映射进行有针对性的匹配应用,获得高质量的匹配结果.

此外,现有的研究中讨论的去匿名化算法都是针对全局、基于两个大小和结构都相近的图进行的.在另一种情况下,攻击者关注的是一部分用户的个人信息,这种情形可以称为局部去匿名化.例如从真实网络中爬取一个子图,与经过匿名化的完整网络进行匹配.相对于全局的去匿名化,局部去匿名化中可能与一个结点匹配的候选结点数增加到了多个,因此具有更高的难度.并且,针对局部去匿名化的研究相对更少.本文介绍的算法对局部去匿名化也具有较好的效果.

在利用常见的匿名化方法(例如,朴素匿名化方法、交换匿名化方法、稀疏化匿名化方法等)处理的图上,本文提出的方法取得了很好的匹配结果.该方法主要可以分为两个阶段:(1) 结点相似度计算阶段;(2) 结点匹配阶段.在第 1 个阶段,笔者根据两张图中结点的结构信息,采用 RoleSim++计算结点对的相似度.在第 2 个阶段,匹配两张图中的点,从而获取点的身份信息.本文的主要贡献是:

- 提出一种基于图结构信息的结点相似度计算方法 RoleSim++,它自带归一化并对大度数和小度数的点都适用(第 2.1 节);
- 提出一种基于结点相似度的结点匹配算法,同时考虑了结点的相似程度和匹配过程中的反馈(第 2.2 节);
- 在传统去匿名化实验方法的基础上,增加了更贴合实际应用情景的局部去匿名化,并取得较好的效果(第 3.2.3 节);
- 利用这些提出的方法,在真实的社交网络数据集上评估了几种不同的匿名化算法的隐私泄漏风险(第 3 节).

本文第 1 节论述现存的相关研究.第 2 节提出去匿名化算法 RoleMatch,具体包括结点 RoleSim++相似度计算和结点匹配.第 3 节在 Live Journal 数据集上,通过模拟实验验证算法的有效性和高效性.最后,阐述研究结论.

1 相关工作

1.1 匿名化算法

根据对社交网络匿名化的一项研究[3],匿名化算法可以分为 3 类:(1) *K*-匿名算法;(2) 边随机匿名化算法;和(3) 基于聚类的泛化匿名化算法.

K-匿名方法[4,5]通过边的加减来修改图结构,使得图中的每一个点至少与 *K*–1 个其他的点在一定的结构特征上不可区分,例如拥有相同的度数,或拥有相同的邻居结构.该方法在匿名化结果上有较好的表现,但是一方面,最优化的 *K*-匿名方法实现相对复杂;另一方面,对图结构的改变程度比较大,可能对图结构的研究有一定的影响.边随机方法通过随机添加边、删除边、交换边来修改图结构[6],它在一定概率上保证了隐私安全.基于聚类的泛化方法[7]首先将结点聚类,然后把每一个子图匿名为一个不带有个体结点具体信息的超点.这样的方法在防止身份识别方面有较好的效果,但是损失了很多个体信息,也损失了规模信息,从而对社交网络分析可能有一定的妨害.

1.2 去匿名化算法

去匿名化算法主要分为两类:一类是需要初始种子对进行扩散的方法;一类是不需要种子对,仅利用图特征信息进行提取判断的算法.各个去匿名化算法之间差别多样,这也是图结构信息表示的多样性带来的特点.

Backstorm 等人[1]提出了一种连续的攻击方式,以此来获知特定的两个点之间有没有边存在.缺陷在于,尽

管该算法对仅仅交换结点编号的朴素匿名化算法效果较好,但对那些会修改图结构的匿名化算法,该攻击方式无能为力.

Narayanan 等人[8]呈现了一种分析隐私的框架,并提出了一种去匿名化算法,只基于网络拓扑结构,并对噪音和大多数防御有较好的健壮性.它需要一些已知的种子匹配对,然后逐渐扩散到全图.然而,根据作者论文中的描述,种子对的质量对去匿名化是否成功至关重要.Narayanan[9]还提出了一种基于模拟退火的边匹配算法,替代去匿名化算法中种子对的效果.而本文考虑的算法不需要基于高质量的匹配种子对.

Yartseva 等人[10]根据图的渗流理论(graph percolation)提出了基于种子的图匹配方法.Kazemi 等人[11]在Yartseva 等人的基础上又放宽了对种子的要求,降低了所需种子的数量.Korula 等人[12]将图的渗流理论应用到了幂律分布的图上,对真实的社交网络匹配有更好的效果.

Fu 等人[13]提出了不需要种子对的社交网络去匿名化算法,该算法首先用公式计算每对结点的相似度,然后根据结点对的相似度由高到低贪心匹配.本文采用与其类似的基本思路,通过计算相似度以匹配结点,提出了更合理的计算公式,达到了更好的匹配精度.

1.3 结点相似度计算方法

去匿名化算法中,结点相似度度量的质量至关重要,因为它是后续匹配情况的最重要依据,表征了结点具有相同身份的可能性.结点相似度的度量,常常是结点本身信息和子图信息的综合考虑,不同的考虑权重,不同的综合方法,导致了不同的度量方法.

Henderson 等人[14]提出了一个相似度度量,递归地将局部特征和邻居特征结合起来产生区域特征,然后用这些区域特征去匿名化.这样的区域特征可以有效缩小可能的对应点范围,但是并无法证明这样可以找到最相似的点(点的真实身份).

著名的 Simrank 方法[15]提供了一张图内结点相似度的度量方法,这在去匿名化问题中无法直接应用.Blondel 等人[16]在此基础上提出了一个多图的结点相似度度量,将两个结点邻居的相似度求和得到相似度.HaoFu 等人[13]提出了两张图的结点相似度度量,迭代计算,对两个点匹配它们的邻居最大化匹配的相似度和.这样的方法对 Simrank 的简单移植是一种有效的改进,对度数大的结点能产生不错的度量结果.但是由于缺乏归一化,尺度不统一,该方法对小度数的结点计算得到的相似度,往往太小以至于失去意义.Jing 等人[17]提出了一种一张图内带归一化的结点相似度度量方案 RoleSim,它对点的结构信息有比较好的刻画,但是定义和计算方法限制在了单图上.本文提出的度量方法可以用于去匿名化匹配,并不带有上述问题.

还有一些研究者希望用机器学习的方法计算结点相似度.Pozzi 等人[18]提出了深度学习算法,通过截取随机游走路径,将该路径视作句子来学习结点的潜在向量表示.Tang 等人[19]提出了大规模信息网络嵌入方法,用低维向量表示大规模图中的结点.

2 基于社交网络图结构的去匿名化算法 RoleMatch

基于社交网络图结构的去匿名化算法 RoleMatch 与现有算法相比匹配结果效果更好,计算方法速度更快.

2.1 结点相似度

去匿名化算法的第 1 个阶段是提供一个结点相似度度量,衡量每一个可能的结点对的相似程度.在这一部分,我们设计了一种新的结点相似度度量,它与现有方法相比有更好的度量效果(精确度);并且有极为高效的计算方法.

2.1.1 两图结点 RoleSim++相似度

文献[5]提出了一种对两图结点相似度的度量方法,基于 Simrank[7]一样的直觉,认为邻居相似的点对本身也相似.分别用 $N_1(u)$,$N_2(v)$表示 u,v 两个来自不同图的点的邻居,该方法对 $N_1(u)$和 $N_2(v)$中相似的点匹配,最大化匹配点对的相似度和.结点相似度 $S(u,v)$的值通过如下方程组定义:

$$S(u,v)=\max_{M\text{是集合}N_1(u)\text{元素和}N_2(v)\text{元素的匹配}}\sum_{(x,y)\in M}S(x,y).$$

但是这个相似度计算缺少了归一化,效果严重依赖结点度数大小,导致不同的结点对,它们的 $S(u,v)$尺度相差很大,整个相似度矩阵也就无法达到衡量相似度的作用.

尽管如此,结点相似度仍然是去匿名化中非常重要的衡量指标,通过对相似度定义的改进,本文设计了结点相似度算法 RoleSim++.

定义. 给定两张图 $G_1=(V_1,E_1),G_2=(V_2,E_2)$,给定两个结点 $u\in V_1$ 和 $v\in V_2$,用 $N_1^+(u),N_1^-(u)$ 和 $N_2^+(v),N_2^-(v)$ 分别表示它们的邻居("+"表示出边连接的邻居,"-"表示入边连接的邻居),用 $|N_1^+(u)|,|N_1^-(u)|,|N_2^+(v)|,|N_2^-(v)|$ 分别表示对应的度数.定义两点的出度较大值、入度较大值为

$$\Delta^+(u,v)=\max\{|N_1^+(u)|,|N_2^+(v)|\},$$

$$\Delta^-(u,v)=\max\{|N_1^-(u)|,|N_2^-(v)|\},$$

以及匹配权和(即邻居点对相似度的最大匹配):

$$\Gamma^+(u,v)=\max_{M^+(u,v)}\sum_{(x,y)\in M^+(u,v)}Sim(x,y),$$

$$\Gamma^-(u,v)=\max_{M^-(u,v)}\sum_{(x,y)\in M^-(u,v)}Sim(x,y),$$

其中,$M^+(u,v)$是对 $N_1^+(u)$ 和 $N_2^+(v)$ 的一个匹配,类似定义 $M^-(u,v)$.对$\Gamma^+(u,v)$和$\Gamma^-(u,v)$的计算可以视作一个最优化的子过程.在此基础上,定义结点 RoleSim++相似度为 $Sim(u,v)=(1-\beta)\dfrac{\Gamma^+(u,v)+\Gamma^-(u,v)}{\Delta^+(u,v)+\Delta^-(u,v)}+\beta$.其中,参数β是一个衰减因子,有界 $0<\beta<1$.

为了将更多的结构信息考虑进去,$Sim(u,v)$分别计算两个方向的边(入边和出边),并且用它们的和作为相似度值.这样的相似度度量,符合相似的点有相似的邻居这一基本直觉,并且自带归一化,对度数无论大小的点都有较好的度量.

在该度量中,可以看到一些与去匿名化相关的事实:(1) 如果两张图形态是完全一样的,那么一个点肯定与另一张图中的对应点最相似(Sim 值为 1);(2) 该算法在计算的时候,不会忽略度数小的点,并且有一个比较合适的尺度.

在该定义的基础上,可以证明迭代计算的单调性,并在证明了单调性后,结合 $Sim^k(u,v)\geqslant\beta$,立即得到收敛性.

接下来证明该相似度度量在迭代计算中的相关收敛性质.

引理 1(单调性). 用 $Sim^k(u,v)$和 $Sim^{k+1}(u,v)$表示点对(u,v)在第 k 轮之后的相似度值,那么对任意的 k 和(u,v),有 $Sim^k(u,v)\geqslant Sim^{k+1}(u,v)$.

证明:按照迭代轮数归纳.注意到 $Sim^0(u,v)=1$,那么:

$$Sim^1(u,v)=(1-\beta)\frac{\min\{|N_1^+(u)|,|N_2^+(v)|\}+\min\{|N_1^-(u)|,|N_2^-(v)|\}}{\max\{|N_1^+(u)|,|N_2^+(v)|\}+\max\{|N_1^-(u)|,|N_2^-(v)|\}}+\beta.$$

因此,$Sim^1(u,v)\leqslant 1$.

在第 $k+1$ 轮,(u,v)的邻居对的相似度值不会比第 k 轮有所增加,那它们之间的匹配权和也不会增加.因此对任意(u,v)、任意 k,有 $Sim^k(u,v)\geqslant Sim^{k+1}(u,v)$. □

在单调性的基础上,结合 $Sim^k(u,v)\geqslant\beta$,容易证明收敛性.

命题 1(收敛性). 上文定义的相似度,对任意(u,v)收敛,即 $\lim_{k\to\infty}Sim^k(u,v)=Sim(u,v)$.

下面的命题表明了β对迭代收敛速度的影响.$Sim^k(u,v)$与 $Sim(u,v)$之间的差距以指数的速度减小.

命题 2. 对任意点对(u,v),用$\varepsilon^k(u,v)$表示 $Sim^k(u,v)-Sim(u,v)$,那么有$\varepsilon^k(u,v)\leqslant(1-\beta)^{k+1}$.

证明:从引理 1 可知 $Sim^k(u,v)-Sim(u,v)\geqslant 0$,接下来,按照迭代轮数归纳证明命题.

由定义得知 $Sim(u,v)\geqslant\beta$,因此 $Sim^0(u,v)-Sim(u,v)\leqslant(1-\beta)^{0+1}=1-\beta$.

在第 k 轮迭代之后,有:

$$\begin{aligned}\varepsilon^k(u,v) &= Sim^k(u,v) - Sim(u,v)\\ &= (1-\beta)(\Gamma_k^+(u,v) + \Gamma_k^-(u,v) - \Gamma_{k+1}^+(u,v) + \Gamma_{k+1}^-(u,v))/(\Delta^+(u,v) + \Delta^-(u,v))\\ &\leqslant (1-\beta)(\min\{|N_1^+(u)|,|N_2^+(v)|\} + \min\{|N_1^-(u)|,|N_2^-(v)|\})(1-\beta)^k/(\Delta^+(u,v) + \Delta^-(u,v))\\ &\leqslant (1-\beta)(1-\beta)^k\\ &= (1-\beta)(1-\beta)^{k+1}.\end{aligned}$$

因此有$\varepsilon^k(u,v)\leqslant(1-\beta)^{k+1}$. □

注意到:如果令β=0.15,迭代轮数为 5,根据命题 2,相似度的理论值和迭代值相差不超过0.85^5=0.44,这仍然不是一个非常小的值.但是在实际计算中,增加迭代轮数对结果的精度几乎没有影响,笔者会在后面的实验部分继续讨论这个问题.

2.1.2 结点相似度计算

基于上面讨论的性质,采用迭代的方式计算 *Sim* 矩阵,伪代码见算法 1.在每一轮(第 3 行~第 9 行),枚举所有的结点对(第 4 行),根据之前的定义计算它们的相似度(第 5 行、第 6 行),在每一轮之后更新矩阵值(第 8 行).迭代若干轮以后得到结果.

算法 1. 迭代更新相似度矩阵.

Input: $G_1=(V_1,E_1), G_2=(V_1,E_1)$;

Output: *Sim*.

1 $Sim \leftarrow MatrixAllOne$;

2 **for** $i=1\rightarrow nRounds$ **do**

3 **for** $\langle u,v\rangle \in V_1\times V_2$ **do**

4 $r \leftarrow \dfrac{\gamma(N_1^+(u),N_2^+(v)) + \gamma(N_1^-(u),N_2^-(v))}{\Delta^+(u,v)+\Delta^-(u,v)}$;

5 $Sim'(u,v) \leftarrow (1-\beta)\cdot r + \beta$;

6 **end**

7 $Sim \leftarrow Sim'$;

8 **end**

9 **return** *Sim*

伪代码中的函数$\gamma(N_u,N_v)$是选择实现算法描述中匹配子过程的方式(也就是Γ^+和Γ^-的实现).在这个子过程中,首先用 N_u 和 N_v 中的点建立二分图,边权为上一轮计算中的相似度.然后,Γ函数用一个最大匹配作为这个子过程的结果.与之对应的,实际计算中采用的方法是:依边权降序排列这些边,然后依次贪心选择,优先选择权值大的边.本文的选择是对原最优化问题的一种高效近似,降低了计算复杂度.

以图 1 为例,考虑其中的点 u_1,取β=0.15,在第 1 轮迭代中,就有:

$$Sim(u_1,v_1) = (1-\beta)\frac{0+2}{0+3} + \beta = 0.72, Sim(u_1,v_1) = (1-\beta)\frac{0+1}{2+3} + \beta = 0.32.$$

类似得到 $Sim(u_1,v_3)$=0.32,$Sim(u_1,v_4)$=0.1,$Sim(u_1,v_5)$=0.43.显而易见:尽管第 1 轮迭代只使用了相邻结点的信息,点 u_1 与 v_1 的相似度显著高于与其他点的相似度.经过 5 轮迭代之后,点对之间的相似度情况见表 1.

Table 1 An example of similarities after 5 iterations

表 1 相似度迭代结果示例(5 轮)

	v_1	v_2	v_3	v_4	v_5
u_1	0.38	0.20	0.22	0.15	0.26
u_2	0.15	0.38	0.36	0.31	0.15
u_3	0.27	0.36	0.38	0.24	0.31
u_4	0.15	0.26	0.26	0.34	0.15

在实际计算中,迭代在几轮之内就能收敛.实际上,在实验中,5 轮以内的结果就已经足够令人满意了.

2.1.3 加速相似度计算

前面介绍了本文提出的结点相似度度量方法以及一个能在几轮内有效收敛的迭代计算方法.然而,考虑到每轮的计算复杂度都有至少$\Omega(|V_1||V_2|d^2)$,其中,d 是平均度数.这样的复杂度仍然是一个不小的负担,尤其是当面对大规模图网络的时候.更重要的是,大多数计算并不必要因为很多结点对本身就不相似.因此,笔者充分利用计算的中间结果,在每一轮只计算有可能匹配的结点对,针对该度量设计了一种快速计算的方法α-Rolesim++".

引入一个参数α,介于 0 和 1 之间,来帮助控制计算量,伪代码见算法 2.当对一个结点 $u\in V_1$ 计算相似度的时候(第 4 行~第 14 行),找到上一轮相似度最高的对应点 $top=\max_{v\in V_2} Sim(u,v)$ (第 5 行),然后设置阈值$\theta=\alpha\cdot top$.V_2 的点中,上一轮相似度不低于θ的被视作有可能与 u 匹配的点,这一轮继续计算(第 8 行、第 9 行);上一轮相似度低于θ的被视作不可能与 u 匹配的点,这一轮被忽略不计算(第 11 行).

算法 2. 迭代快速更新相似度矩阵.

Input : $G_1=(V_1,E_1),G_2=(V_1,E_1)$;

Output : Sim.

1 $Sim\leftarrow InitSim(G_1,G_2)$;

2 **for** $i=1\rightarrow nRounds$ **do**

3 **for** $u\in V_1$ **do**

4 $top\leftarrow \max\{Sim(u,v)\mid v\in V_2\}$;

5 **for** $v\in V_2$ **do**

6 **if** $Sim(u,v)\geqslant \alpha\cdot top$ **then**

7 $r\leftarrow \dfrac{\gamma(N_1^+(u),N_2^+(v))+\gamma(N_1^-(u),N_2^-(v))}{\Delta^+(u,v)+\Delta^-(u,v)}$;

8 $Sim'(u,v)\leftarrow (1-\beta)\cdot r+\beta$;

9 **else**

10 $Sim'(u,v)\leftarrow Sim(u,v)$;

11 **end**

12 **end**

13 **end**

14 $Sim\leftarrow Sim'$;

15 **end**

16 **return** Sim

参数α具体值的决定方式和敏感度分析,将在实验部分详细展开.

用这样的计算方法,原先的全置一的初始化方式不再合适,需要模拟 RoleSim++第 1 轮的结果,而这一结果可以在 $O(|V_1||V_2|)$时间复杂度内计算,因为已知在这样的情况下:

$$\gamma(N_1^+(u),N_2^+(v))=\min\{|N_1^+(u)|,|N_2^+(v)|\},$$
$$\gamma(N_1^-(u),N_2^-(v))=\min\{|N_1^-(u)|,|N_2^-(v)|\}.$$

到目前为止,用该快速计算的方法以及一个合适的参数α,可以对基本的迭代计算有一个非常显著的速度改进.

2.2 结点匹配

本节讨论不同的跨图结点匹配算法,并评估每一个算法的复杂度.准确性的实验评估将在后面给出.

2.2.1 不使用结构信息的匹配

直觉上,基于结点相似度的去匿名化,考虑二分图的带权最大匹配.使用 KM 算法,最大匹配可以在时间复杂度 $O(n^3)$内被计算.

然而,最大匹配过程求解复杂度高,难以在大规模图中被采用.Fu 等人[13]采用了一种贪心的方法,提供了一

种对全局最优匹配的近似,在一定的精确度牺牲的情况下,能在 $O(n^2\log n)$时间复杂度内计算.

2.2.2 使用邻居信息的匹配

在第 2.2.1 节中提到的两种算法都只考虑了相似度的大小,而图的结构信息被忽略了.通常,高相似度的结点对在去匿名化过程中更重要,通过图结构的传递,能够给它周围的其他结点的匹配提供相当的反馈信息.基于这一点,先匹配最高相似度的结点对,然后在后续的匹配中给它们的邻居一个更高的优先级作为反馈,辅助修正整个匹配过程.笔者设计了两种算法——DfsMatch 和 NeighborMatch,二者的时间复杂度都是 $O(n^2)$.

DfsMatch 按照深度优先的顺序匹配结点:首先,具有最高相似度的一对结点被匹配;然后,每当一对结点被匹配,它们的所有邻居都被加入到一个等待匹配的队列里,根据相似度的降序排列.对朴素匿名化方法,DfsMatch 表现接近完美,但是对网络噪音的容忍度很差,因为一旦有一对错误匹配,所有的邻居对都会被影响,然后匹配错误.

对 NeighborMatch 来说,邻居的优先级按照一个相对少一点的量递增,从而避免深度优先序那样的限制及其严重后果.具体见算法 3,其过程可以描述如下.

(1) 首先,预处理相似矩阵 *Sim*,另一个矩阵 *rank_score* 被初始化为原来的相似矩阵的拷贝,并影响下一步中的结点匹配.对每一个匿名图中的结点 u,在另一张图找与它有最高相似度的点,记为 $top(u)$;

(2) 每一次根据 $top()$挑选出具有最高相似度的点对(u,v)并匹配.既然 v 被匹配了,用当前未被匹配的点中相似度最高的更新 $top(w)=v$ 的点 w 的 *top* 值;另一方面,在矩阵 *rank_score* 将二者邻居对的值都增加原来相似度值的大小,并更新相应的 *top* 值.

重复上述过程(2),直到所有结点被匹配.

算法 3. 结点匹配.

Input: $G_1=(V_1,E_1),G_2=(V_2,E_2),score[n][n]$;

Output: $node\_match[n]$.

```
1 rank_sim[n][n] ← score[n][n];
2 for i = 1 → n do
3   top[i] ← top rank id
4 end
5 for i = 1 → n do
6   u ← FindMax(top);
7   v ← top[u];
8   node_match[u] ← v;
9   for ∀neighbor pair(x,y) of (u,v) do
10    rank_score[x][y] ← rank_score[x][y] + score[u][v];
11  end
12  update top[n]
13 end
14 return node_match
```

仍以图 1 的情况为例,在表 1 的相似度结果基础上,第 1 次可选择点 u_1 和 v_1 匹配;同时,点对$(u_2,v_2),(u_2,v_3),(u_3,v_2),(u_3,v_3),(u_4,v_2),(u_4,v_3)$的 *rank_score* 将分别被加上 0.38.经过 4 次匹配之后,能够将图中的点对都正确配对.

3 实验分析

本节通过实验评估了相似度度量和结点匹配算法的性能表现,将两者和 Fu 等人[13]方法的对应部分比较,在比较中主要关注以下 3 个方面:(1) 结点相似度度量作为中间结果的准确度;(2) 去匿名化算法整体的准确度;(3) 相似度计算的时间开销以及效率和准确度之间的权衡.

3.1 实验基本设置

本节讨论在实验中使用的公开社交网络数据集,以及如何从中提取子图作为去匿名化的输入.

3.1.1 数据集

本文选择 LiveJournal 数据集(LiveJournal 数据集发布在 http://snap.stanford.edu/data/)和 Enron 数据集(Enron 数据集发布在 https://www.cs.cmu.edu/~./enron/).

LiveJournal 的图包含 4 847 571 个结点和 68 993 771 条有向边,其中,每一个结点表示一个 Live Journal 用户,每条边表示网站上两个用户之间的朋友关系(有向).由于该图对现有去匿名化算法而言规模过大,本文随机在其中选取大小为 10 000 个点的子图用以分析,并记为 $G_0=(V_0,E_0)$.

从 G_0 中抽取生成一对图 $G_1=(V_1,E_1)$和 $G_2=(V_2,E_2)$,并保证交叠率λ,以此模拟爬虫爬取的社交网络和组织公开发布的匿名化以后的图,其中交叠率λ定义为 $\frac{V_1\cap V_2}{V_1\cup V_2}$.这可以通过宽度优先搜索算法保证.首先,使用宽度优先搜索爬取一张交叠图 $G_0=(V_0,E_0)$,并且使得$|V_0|=\lambda|V_0|$;然后,随机将剩余的点一半归入 G_1,一半归入 G_2.在这样生成的两张图上,对 G_2 应用选择的匿名化算法后,与 G_1 一起作为去匿名化的输入.

Enron 数据集是一个 email 关系图,包含 36 692 个结点(表示邮件地址)和 367 662 条有向边(表示邮件通信关系).对该数据集,全图匿名化后与原图做去匿名化实验.

3.1.2 匿名化算法

在这里列举在实验中用到的所有匿名化算法.

- 朴素匿名化(Naive):该算法简单地打乱结点的标志(编号),保持结构不变;
- 稀疏化(Sparsify):稀疏化方法随机移除 $p|E|$条边,其中,参数 p 控制匿名化程度;
- 扰动(Perturb):扰动方法先与稀疏化一样随机移除边,然后增加不存在的边,直到总的边数和匿名化之前一样.该方法可以看做是对社交网络演化的一种模拟,或者是无目的性的匿名化;
- 交换(switch):交换方法随机选择两条边$(u_1,u_2),(v_1,v_2)$,并且不存在边$(u_1,v_2),(u_2,v_1)$.然后交换两条边,也就是说:边$(u_1,u_2),(v_1,v_2)$被删除,而边$(u_1,v_2),(u_2,v_1)$被添加.这样的过程重复 $p|E|/2$ 次.

为了保证数据的有效性,在实验中设置 p=0.1.

3.1.3 去匿名化算法

在实验中比较了以下 3 种去匿名化算法.

- 基准算法(Baseline):Fu 等人[13]提出的去匿名化算法.该算法先通过度数相关的公式计算点对相似度,然后根据相似度贪心匹配结点对;
- RoleMatch(Rolesim++):用算法 1 计算相似度,再用算法 3 进行结点匹配.该算法在迭代中,每轮计算所有点对的相似度,然后使用相似度加邻居信息匹配结点;
- RoleMatch(α-Rolesim++):在 RoleSim++的基础上用算法 2 加快相似度计算,再用算法 3 进行结点匹配.该算法在相似度迭代中忽略不相似点对,然后使用相似度加邻居信息匹配结点.

3.1.4 评估准则

为了对 3 种算法有一个清晰的评估,在评估去匿名化算法的整体效果时,提出 3 个重要的指标.

- 精确度分数:为了评估一个算法,最直观的方式就是通过正确匹配的比例.这里定义正确匹配的比例为 $W=W'/|V_1\cap V_2|$,其中,W'是正确匹配的对数.由于在同一组去匿名化实验中$|V_1\cap V_2|$是一个常数,因此,本文在实验结果中以正确匹配的对数 W'来代表每个算法的精确度分数;
- 中间结果分数:由于评估的算法都会生成相似度矩阵来代表结点对的相似度,就用最相似对恰好是正确匹配的数量来评估相似度模型.也就是说,令 S'为 G_1 中这样的点的数量,它对应的正确匹配点恰好是相似度最高的点,那么定义相似度比例为 $S=S'/|V_1\cap V_2|$.与精确度分数相同,本文在实验结果中以 S'来代表中间结果分数;
- 执行时间:如果一个算法时间复杂度非常高,那么该算法就不可能被应用于大规模的图,应用性不好.本

文用 T 来表示实验中算法的执行时间.

3.1.5 实验环境

在实验中使用了一台配置为 AMD Opteron Processor 4180(6 核,12 线程,2.6GHz),48GB 内存的机器.由于迭代计算的阶段可以直接并行,实验中使用了 8 个线程.所有的算法都用 C++实现.

3.2 评 估

本文在不同的交叠率和匿名化算法下进行了一系列实验,并评估了 3 种去匿名化算法的性能(基于之前提出的评估准则).在实验中,交叠率分别被设置为 50%和 100%.在每一种情况下,为 3 种算法随机生成 10 对图,并用平均结果来分析.

3.2.1 参数选择

- 衰减因子:根据 RoleSim[8]的结果,选取β=0.15.
- 迭代轮数:3 种算法在实现时均采用了迭代计算,迭代计算结点相似度时,迭代轮数会影响精度,迭代轮数越多时精度越高,但是花费的时间也更多.因此需要通过实验确定迭代轮数与精度间的平衡.实验结果如图 2 所示,在 LiveJournal 数据集上使用稀疏化匿名化算法以及 RoleSim++去匿名化.实验的结果表明,正确的匹配数在 5 轮以后就基本保持不变.因此,本文将后续实验中的迭代轮数统一设置为 5 轮;
- 加速算法中的参数α:从算法 3 中可以直观地看出:当α越小时,每轮迭代的点数越多,精度越高,然而时间开销也越大(当α=0 时与 RloeSim++等价).我们需要选定一个α值,在精度得到保持的情况下,时间开销尽量小.调整α从 0.95~0.50,步长 0.05,在 10 对从 LiveJournal 数据集中随机提取的图上跑 RoleSim++算法和α-Rolesim++算法,结果如图 3 所示.当α增大时,时间开销几乎是线性减少的;而在$\alpha\leqslant$0.85 时,精度能得到较好的保持.因此在后续实验中,选定α=0.85.

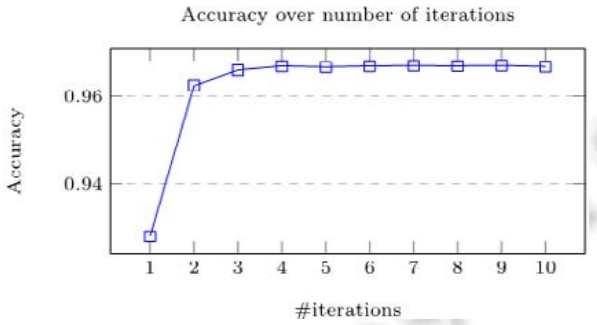


Fig.2 Accuracy over number of iterations (Sparsify, RoleSim++)

图 2 精确度关于迭代轮数的变化(Sparsify, RoleSim++)

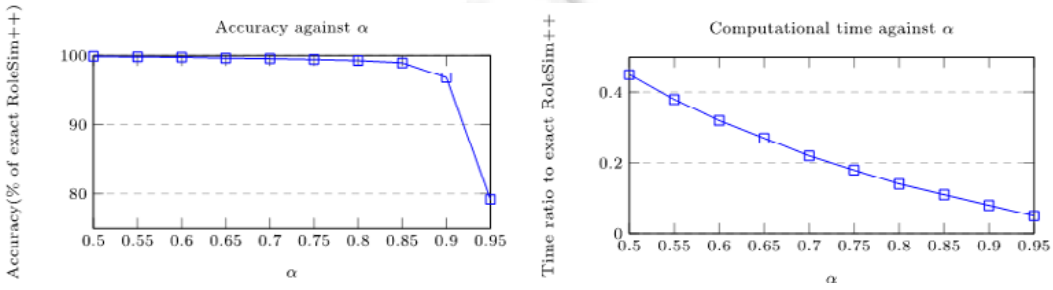


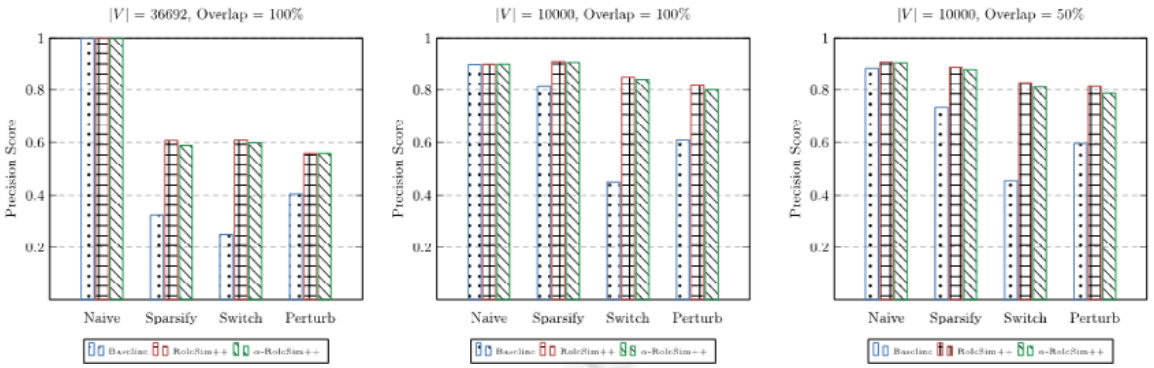
Fig.3 Accuracy and time over α (RoleSim++, α-RoleSim++)

图 3 精确度和时间关于α的变化(RoleSim++,α-RoleSim++)

3.2.2 对匹配精度的比较

本节实验对比了 3 种去匿名化算法的精度.在实验中,对各种匿名化算法生成的目标图,用 3 种算法进行去

匿名化,统计正确匹配顶点的数量.在 LiveJournal 数据集上,分别按交叠率 50%和 100%实验;在 Enron 数据集上,设定交叠率为 100%,实验的匹配结果如图 4 所示.



(a) Enron 数据集 (b) Livejournal 数据集 (c) Livejournal 数据集

Fig.4 Accuracy over different anonymization algorithms

图 4 不同匿名化算法下的精确度比较

在这两个数据集上的 3 小组实验,结果是一致的.对经过朴素匿名化的目标图,基准算法(Baseline)在交叠率 50%和 100%的图上,效果几乎和本文提出的算法相当.这是因为匿名化算法不改变交叠部分的图结构,给去匿名化降低了难度.当不同的去匿名化算法被应用的时候,本文提出的 RoleMatch 算法精度远远高于基准算法.后者的表现在不同的匿名化方法上差异很大,在交换匿名化上表现最差,当交叠率为 50%时,只去匿名化了 40%的交叠点.RoleMatch(Rolesim++)算法和 RoleMatch(α-Rolesim++)算法在每一种匿名化算法下都保持了较好的健壮性,在 LiveJournal 数据集的实验中,大约能正确匹配 80%以上的交叠点;在 Enron 数据集的结果中,也能正确匹配一半以上.

3.2.3 局部去匿名化

本节实验对比了 3 种算法在非匿名图和匿名图大小不同时的去匿名化的效果.实验中,我们从 LiveJournal 图中爬取一定大小的非匿名图,并应用匿名化算法于实验图中得到匿名图.这样,非匿名图即为匿名图的局部子图.两张图的交叠率从 15%~35%进行匹配实验,多次匹配并求取平均值.实验结果如图 5 所示.

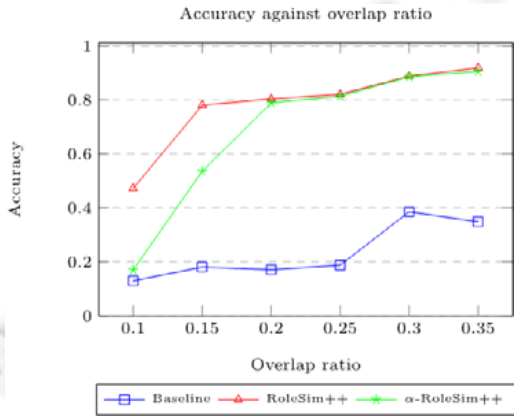


Fig.5 Accuracy against overlap ratio for the local deanonymization problem

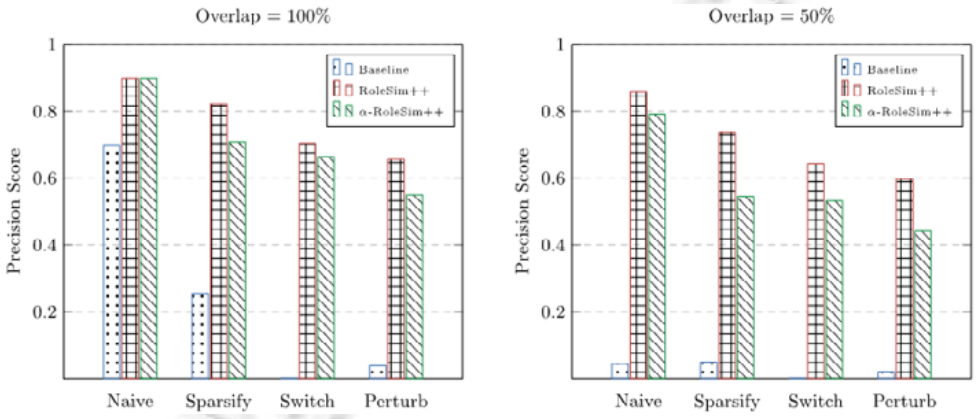
图 5 局部去匿名化中不同交叠率的精确度比较

在图 5 中可以看到:随着交叠率的增加,3 种算法的匹配精度均出现增长.当交叠率低于 0.15 时,3 种算法的去匿名化精度均低于 50%.这是因为当交叠率低于 15%时,匿名图中其他点过多对去匿名化造成影响较大.当交

叠率高于 0.20 后,Rolesim 算法和 Rolesim++算法都取得了 80%以上的去匿名化精度,而 Baseline 算法的去匿名化精度始终低于 40%.

3.2.4 对中间结果的考察

为评估相似度算法的合理性,本节实验对比了 3 种算法的中间结果精度.在实验中,选择第 1 阶段迭代计算之后得到的相似度矩阵,对每个匿名化后的顶点,统计与其相似度最高的顶点恰好是正确匹配的数量.进而又统计了对每个匿名化后的顶点,正确匹配的相似度位于前 1%~10%的比例.比较得到的结果如图 6 和图 7 所示.



(a) Top-1 精度 (b) Top-1 精度

Fig.6 Intermediate results over different anonymization algorithms

图 6 不同匿名化算法下的中间结果比较

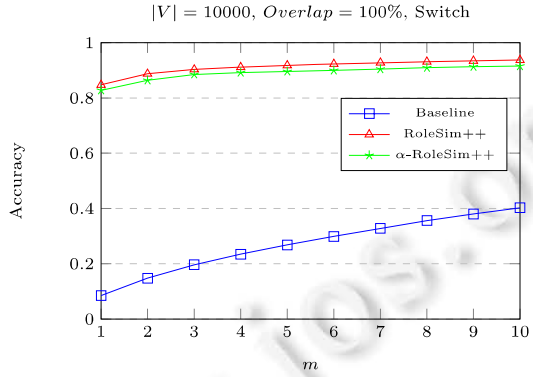


Fig.7 Accuracy of intermediate results (top m%)

图 7 中间结果的精确度(前 m%)

Rolesim++算法和α-Rolesim++算法在不同的匿名化算法下比基准算法有更好的表现,尤其是当交叠率不高的时候.另外,当交叠率逐渐降低时,Rolesim++算法和α-Rolesim++算法之间的差别逐渐明显.与图 4 相对照,更好的相似度度量可以带来更高的去匿名化精度.基准算法在交换匿名化上有最差的精度表现,它的相似度分数也在该匿名化上表现最差(低于 1%).

在这个最相似点对的基础上,可以通过优先匹配相似度高的点对获得更高的精确度,因为它们往往有更大的概率是正确的匹配.这也是为什么很多最相似度不排在第 1 的点对,在结点匹配阶段之后能够被正确匹配,而最后的去匿名化结果比中间结果要好.

3.2.5 执行时间

本节实验比较 Rolesim++算法和α-Rolesim++算法的时间性能.Baseline 算法的时间复杂度和实验中的实际

耗时与 Rolesim++算法并没有显著区别,因此不再与之比较.对于每种去匿名化算法,分别使图中点的数量|*V*|与边的平均密度 *d* 增加.

图 8 展示了算法执行时间的变化.Rolesim++算法的执行时间随着点数与边的密度的增加而显著增加,而 α-Rolesim++算法总是比它更快,而且耗时的增加相对缓慢.由此体现了 α-Rolesim++算法的高效性.

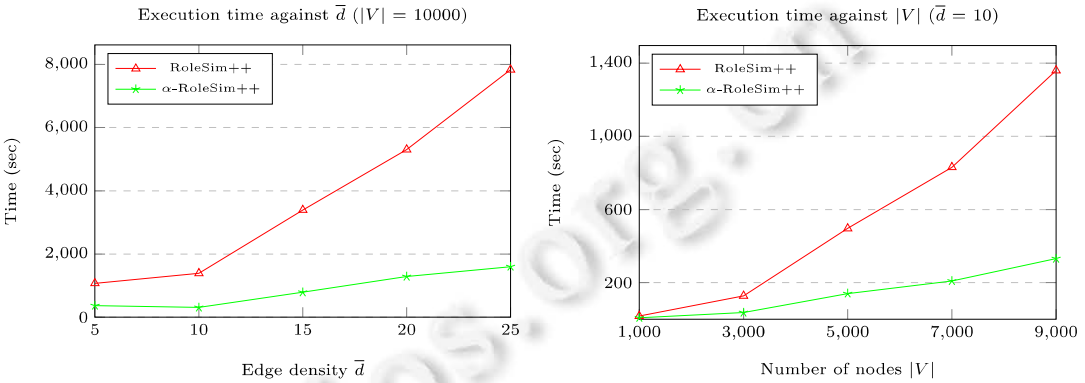


Fig.8 Execution time over |*V*| and *d*

图 8 执行时间与|*V*|和 *d* 的关系

4 结 论

本文提出了一种不需要已知种子的去匿名化算法 RoleMatch,分为结点相似度计算阶段和结点匹配阶段.其中,结点相似度阶段计算的精确而具有容错性的相似度,刻画了不同的图中结点有同一身份的可能性;动态的结点匹配阶段同时考虑了结点的相似度对匹配的影响,以及之前匹配结果的反馈.这一算法仅需要社交网络图的拓扑结构信息,产生非常好的去匿名化结果,并有一个快速计算的版本,能够高效去匿名化.在 LiveJournal 真实数据集上进行的实验,实验方式在传统对称去匿名化的基础上还进行了更贴合应用情景的局部去匿名化实验,各项实验的结果进一步表明了去匿名化的准确性和高效性.接下来,我们将进一步考虑算法的分布式实现,并考虑在富信息图上利用更多的信息进行去匿名化.

References:

[1] Backstrom L, Dwork C, Kleinberg J. Wherefore art thou r3579x?: Anonymized social networks, hidden patterns, and structural steganography. In: Proc. of the 16th Int'l Conf. on World Wide Web. ACM Press, 2007. 181−190. [doi: 10.1145/1242572.1242598]

[2] Wang Y, Zheng B. Preserving privacy in social networks against connection fingerprint attacks. In: Proc. of the 2015 IEEE 31st Int'l Conf. on Data Engineering. IEEE. 2015. 54−65. [doi: 10.1109/ICDE.2015.7113272]

[3] Wu X, Ying X, Liu K, Chen L. A survey of privacy-preservation of graphs and social networks. In: Proc. of the Managing and Mining Graph Data. Springer-Verlag, 2010. 421−453. [doi: 10.1007/978-1-4419-6045-0_14]

[4] Liu K, Terzi E. Towards identity anonymization on graphs. In: Proc. of the 2008 ACM SIGMOD Int'l Conf. on Management of Data. ACM Press, 2008. 93−106. [doi: 10.1145/1376616.1376629]

[5] Zhou B, Pei J. Preserving privacy in social networks against neighborhood attacks. In: Proc. of the 2008 IEEE 24th Int'l Conf. on Data Engineering. IEEE, 2008. 506−515. [doi: 10.1109/ICDE.2008.4497459]

[6] Bonchi F, Gionis A, Tassa T. Identity obfuscation in graphs through the information theoretic lens. In: Proc. of the Information Sciences. Elsevier, 2014. 232−256. [doi: 10.1016/j.ins.2014.02.035]

[7] Zheleva E, Getoor L. Preserving the privacy of sensitive relationships in graph data. In: Proc. of the Privacy, Security, and Trust in KDD. Berlin, Heidelberg: Springer-Verlag, 2008. 153−171. [doi: 10.1007/978-3-540-78478-4_9]

[8] Narayanan A, Shmatikov V. De-Anonymizing social networks. In: Proc. of the 2009 30th IEEE Symp. on Security and Privacy. IEEE, 2009. 173−187. [doi: 10.1109/SP.2009.22]

[9] Narayanan A, Shi E, Rubinstein BI. Link prediction by de-anonymization: How we won the kaggle social network challenge. In: Proc. of the 2011 Int'l Joint Conf. on Neural Networks (IJCNN). IEEE, 2011. 1825−1834. [doi: 10.1109/IJCNN.2011.6033446]

[10] Yartseva L, Grossglauser M. On the performance of percolation graph matching. In: Proc. of the 1st ACM Conf. on Online Social Networks. ACM Press, 2013. 119−130. [doi: 10.1145/2512938.2512952]

[11] Kazemi E, Hassani SH, Grossglauser M. Growing a graph matching from a handful of seeds. Proc. of the VLDB Endowment, 2015, 8(10):1010−1021. [doi: 10.14778/2794367.2794371]

[12] Korula N, Lattanzi S. An efficient reconciliation algorithm for social networks. Proc. of the VLDB Endowment, 2014,7(5): 377−388. [doi: 10.14778/2732269.2732274]

[13] Fu H, Zhang A, Xie X. Effective social graph deanonymization based on graph structure and descriptive information. ACM Trans. on Intelligent Systems and Technology (TIST), 2015,6(4):49. [doi: 10.1145/2700836]

[14] Henderson K, Gallagher B, Li L, Akoglu L, Eliassi-Rad T, Tong H, Faloutsos C. It's who you know: Graph mining using recursive structural features. In: Proc. of the 17th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2011. 663−671. [doi: 10.1145/2020408.2020512]

[15] Jeh G, Widom J. SimRank: A measure of structural-context similarity. In: Proc. of the 8th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2002. 538−543. [doi: 10.1145/775047.775126]

[16] Blondel VD, Gajardo A, Heymans M, Senellart P, Van Dooren P. A measure of similarity between graph vertices: Applications to synonym extraction and Web searching. Siam Review, 2004,46(4):647−666. [doi: 10.1137/S0036144502415960]

[17] Jin R, Lee VE, Hong H. Axiomatic ranking of network role similarity. In: Proc. of the 17th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2011. 922−930. [doi: 10.1145/2020408.2020561]

[18] Perozzi B, Al-Rfou R, Skiena S. Deepwalk: Online learning of social representations. In: Proc. of the 20th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2014. 701−710. [doi: 10.1145/2623330.2623732]

[19] Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q. Line: Large-Scale information network embedding. In: Proc. of the 24th Int'l Conf. on World Wide Web. ACM Press, 2015. 1067−1077. [doi: 10.1145/2736277.2741093]

刘家霖(1995－),男,福建晋江人,主要研究领域为数据库.

史舒扬(1994－),男,学士,主要研究领域为数据库.

张悦眉(1995－),女,主要研究领域为数据库.

邵蓥侠(1988－),男,博士,主要研究领域为数据库,知识图谱数据管理,并行图计算,知识工程.

崔斌(1975－),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为数据库,大数据管理分析.