软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2016,27(8):1904−1921 [doi: 10.13328/j.cnki.jos.005092]

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

一种保持结点可达性的高效社会网络图匿名算法*

刘向宇, 李佳佳, 安云哲, 周大海, 夏秀峰

(沈阳航空航天大学 计算机学院,辽宁 沈阳 110136)

通讯作者: 刘向宇, E-mail: liuxy@sau.edu.cn

摘 要: 为了保护社会网络隐私信息,提出了多种社会网络图匿名化技术.图匿名化目的在于通过图修改操作来防止隐私泄露,同时保证匿名图在社会网络分析和图查询方面的数据可用性.可达性查询是一种基本图查询操作,可达性查询精度是衡量图数据可用性的一项重要指标.然而,当前研究忽略了图匿名对结点可达性的影响,导致较大的可达性信息损失.为了保持匿名图中结点的可达性,提出了可达性保持图匿名化(reachability preserving anonymization,简称 RPA)算法,其基本思想是将结点进行分组并采取贪心策略进行匿名,从而减少匿名过程中的可达性信息损失.为了保证 RPA 算法的实用性,针对其执行效率进行优化,首先提出采用可达区间来高效地评估边添加操作所导致的匿名损失;其次,通过采用候选邻居索引,进一步加速 RPA 算法对每个结点的匿名过程.基于真实社会网络数据的实验结果表明了 RPA 算法的高执行效率,同时验证了生成匿名图在可达性查询方面的高精度.

关键词: 社会网络;隐私;匿名;可达性

中图法分类号: TP301

中文引用格式: 刘向宇,李佳佳,安云哲,周大海,夏秀峰.一种保持结点可达性的高效社会网络图匿名算法.软件学报,2016,27(8):1904−1921. http://www.jos.org.cn/1000-9825/5092.htm

英文引用格式: Liu XY, Li JJ, An YZ, Zhou DH, Xia XF. Efficient algorithm on anonymizing social networks with reachability preservation. Ruan Jian Xue Bao/Journal of Software, 2016,27(8):1904−1921 (in Chinese). http://www.jos.org.cn/1000-9825/5092.htm

Efficient Algorithm on Anonymizing Social Networks with Reachability Preservation

LIU Xiang-Yu, LI Jia-Jia, AN Yun-Zhe, ZHOU Da-Hai, XIA Xiu-Feng

(School of Computer Science, Shenyang Aerospace University, Shenyang 110136, China)

Abstract: As a proven effective solution to privacy preservation, graph anonymization has been studied extensively. The goal of graph anonymization is to avoid disclosure of privacy in social networks through graph modifications while at the same time preserving data utility of the anonymized graph for social network analysis and graph queries. Reachability is an important graph data utility as reachable queries are not only common on graph databases but also serving as fundamental operations for many other graph queries. However, the reachability of each vertex in the anonymized graph is severely distorted after the anonymization due to neglecting that the reachability is highly sensitive to edge modifications. This work solves the problem by designing a reachability preserving anonymization (RPA) algorithm. The main idea of RPA is to organize vertices into groups and greedily anonymizes each vertex with low impact on reachability. A number of techniques are designed to make RPA efficient. Firstly, reachable interval is proposed to efficiently measure the anonymization cost incurred by an edge addition. Secondly, an index structure, CN-index is adopted to accelerate anonymizing each vertex. Extensive experiments on real datasets demonstrate that RPA performs with high efficiency and the generated anonymized social networks preserve high data utility on reachable queries.

Key words: social network, privacy, anonymization, reachability

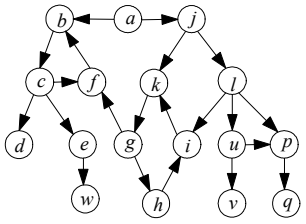
* 基金项目: 国家自然科学基金(61502316, 61502317); 沈阳航空航天大学校博士启动金(15YB36)

Foundation item: National Natural Science Foundation of China (61502316, 61502317); Doctor Startup Fund of Shenyang Aerospace University of China (15YB36)

收稿时间: 2015-12-16; 采用时间: 2016-04-14

随着社会网络的快速发展和普及,社会网络中隐私信息的安全性成为当前数据隐私保护研究中的热点问题.为了保护社会网络中的隐私信息,提出了多种社会网络图匿名化技术.图匿名化目的在于通过图修改操作来防止隐私泄露,同时保证匿名图在社会网络分析和图查询方面的数据可用性.可达性查询是一种基本的图查询操作,同时,可达性查询精度是衡量图数据可用性的一项重要指标.然而,当前图匿名技术没有考虑图匿名操作对结点间可达性的影响,导致很大的可达性信息损失.

在社交网站服务(social networking service,简称 SNS)中,通常支持两个用户之间的多种关系查询.一种常见的关系查询是可达性查询,即查看两个结点间是否存在一条可达路径.结点可达性是衡量图数据可用性的重要指标,在社会网络中,可达性查询操作更加频繁.例如,很多社会网络(Facebook、QQ 朋友网等)均支持人脉联系查询:输入用户 u 和 v,返回 u,v 之间的可达路径以及路径上包含的用户.显然,此类人脉联系查询的本质是结点间的可达性查询操作.很多社会网络应用基于结点间的可达性来进行好友推荐,提高社会网络的粘度和用户活跃度.因此,保持社会网络中结点间的可达性具有实际意义.然而在实际应用中,社会网络图中的结点可达性由于图匿名而受到严重影响,其主要原因在于当前图匿名技术忽视了图匿名过程中的边操作对结点间可达性的影响.例如,图 1 显示了一个虚构的微博网络图 G 及图中结点的度,其中,d^{in} 和 d^{out} 分别表示结点的入度和出度,有向边(a,b)表示用户 a 收听了用户 b.假设攻击者已知 Alice 没有粉丝并且 Alice 收听了其他两个用户,即 d^{in}(Alice)=0 和 d^{out}(Alice)=2,显然,攻击者可以 100%的置信度识别出结点 a 是 Alice,因为图中只有结点 a 具有和 Alice 相同的度(包括入度和出度).



(a) G

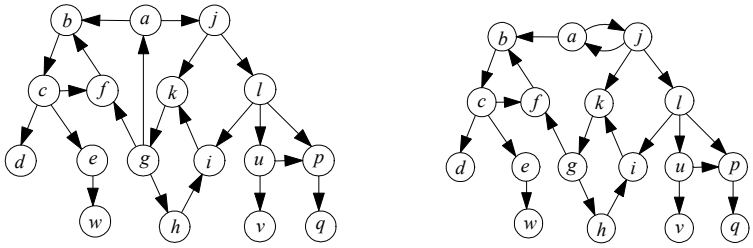
d^{in}	d^{out}	Vertices
2	1	*b*,*f*,*i*,*k*,*p*
1	3	*c*,*l*
1	2	*g*,*j*,*u*
1	1	*e*,*h*
1	0	*d*,*q*,*v*,*w*
0	2	*a*

(b) 结点的度

Fig.1 A social network graph and the degrees

图 1 虚构微博网络 G 及结点的度

文献[1]提出了 k-度匿名隐私保护模型来防止结点度作为背景知识的身份识别攻击.所谓 k-度匿名,是指对于图中任意结点 v,至少有 k–1 个其他结点与 v 的度相同,使得攻击者识别结点 v 真实身份的概率≤1/k.例如,针对图 1(a)中的 G 进行图匿名化,得到图 2 中的两个 2-度匿名图 G_1 和 G_2.



(a) G_1 (b) G_2

Fig.2 Two anonymized versions for G

图 2 G 的两个匿名图

以 G_1 为例,对于 G_1 中的任意结点,至少有另一个结点与其具有相同的出、入度,因此,攻击者基于度来识别结点身份的概率不大于 50%.如果结点 u 到 v 可达,则称⟨u,v⟩是一个可达对.虽然 G_1,G_2 均为 G 的 2-度匿名图,但

是在保持结点可达性方面具有不同效果.具体来说,在 G 中添加边(g,a)得到 G_1,导致增加 31 个新可达对,包括$\langle g,a\rangle$和$\langle h,p\rangle$等.而在 G_2 中,仅增加了一个新可达对$\langle j,a\rangle$.令 $R(G)$表示 G 中所有可达对的集合,并定义结点到其自身可达.给定图 G 及其 k-匿名图 G_k,定义匿名损失 $Cost(G,G_k)$来评估 G_k 在结点可达性上的数据可用性,具体计算如公式(1)所示.

$$Cost(G,G_k)=|R(G)-R(G_k)|+|R(G_k)-R(G)| \tag{1}$$

公式(1)计算了 G_k 中新增和减少的可达对数目.较大的 $Cost(G,G_k)$数值表示了 G_k 在结点可达性上的低数据可用性.根据公式(1),可以算得 $Cost(G,G_1)=31$ 和 $Cost(G,G_2)=1$,表明 G_2 比 G_1 更好地保持了结点间的可达性.

在本文研究工作中,针对社交网络图 G,期望在生成其匿名图 G_k 的同时最小化匿名损失 $Cost(G,G_k)$.本文的主要工作及贡献如下:

(1) 提出了一种保持结点间可达性的社会网络图匿名算法 RPA(reachability-preserving-anonymization),在进行图匿名化的同时保证了较小的可达性信息损失;

(2) 提出了可达区间以及相应的生成算法,基于可达区间实现高效评估边操作所导致的匿名损失,提高 RPA 算法的执行效率;

(3) 通过在结点匿名过程中采用候选邻居索引(CN-index),提高了 RPA 算法中候选邻居查找效率,加速了 RPA 算法对每个结点的匿名过程,保证了 RPA 算法在处理大规模社会网络图数据时的实用性.

本文第 1 节介绍在社会网络隐私保护方面的相关工作.第 2 节介绍相关预备知识,同时给出本文的研究问题定义.第 3 节给出可达性保持图匿名算法 RPA,同时从执行效率方面提出 RPA 算法面临的两大挑战,即为边添加匿名损失评估、选择最小匿名代价出边邻居的问题.针对边添加匿名损失评估问题,第 4 节提出可达区间定义,基于可达区间实现匿名损失的高效评估.针对选择最小匿名代价出边邻居问题,第 5 节提出采用候选邻居索引,基于 CN-index 为结点匿名提出两种结点添加策略:单邻居添加策略和多邻居添加策略,从而优化了结点匿名效率.第 6 节基于真实社会网络数据,分别从算法执行时间、匿名损失、添加边和结点数目、图结构信息损失等方面进行实验测试.第 7 节总结全文并给出未来的工作.

1 相关工作

在社会网络中,常见的隐私攻击主要分为两种类型:结点识别攻击和链接识别攻击.在结点识别攻击(结点身份泄露)中,攻击者利用攻击目标在网络中所属的子图结构来识别其在社会网络图中的位置,从而导致攻击目标的身份隐私泄露.为了防止结点识别攻击,文献[2]提出通过匿名化每个结点的邻居子图来防止结点身份泄露.在文献[3]中,通过将结点聚类生成超点,从而使位于同一超点内的结点相互间不可区分,实现对超点内每个结点的身份隐私保护.文献[4]提出 k-自同构图匿名模型,通过图匿名化操作,使得结点位于同构位置,从而实现结点隐私保护.在链接识别攻击(敏感关系泄露)中,攻击者的目的在于识别出社会网络中实体之间的敏感关系.为了防止敏感关系泄露,文献[5]提出了安全分组技术来保护二部图中的链接隐私,其主要思想是:对二部图中的结点进行分组,并且位于同一分组的结点在边连接方面相互间不可区分,分组过程中,保证每条边的端点被推演概率小于安全阈值,从而起到边隐私保护的作用.文献[6]在文献[5]的基础上,提出采用结点聚类方法来保护链接隐私.文献[7]提出子图随机化方法来保护有向图中的链接隐私,通过在子图内进行随机图操作,在为边隐私信息提供保护的同时减少图操作对于图数据可用性的影响.文献[8]研究了云环境中的邻居隐私保护技术,并保证发布数据图在最短路径查询方面的数据可用性.文献[9]将攻击目标与公众名人之间的链接作为背景知识的攻击,称为 Connection Fingerpring 攻击,并设计了两种分别基于添加伪点和边修改的图匿名算法,从而保持社会网络图的高数据可用性.文献[10,11]提出更加通用的隐私保护机制来同时保护结点和链接隐私信息.文献[10]提出 k-同构图匿名模型来同时保护结点隐私和边隐私,其基本思想是:将图分割和重新构建为 k 个相互同构的子图,使得结点和链接隐私泄露概率均小于 $1/k$.文献[11]给出了一种个性化隐私保护方法,在结点隐私和边隐私方面满足不同隐私保护层次和安全要求.上述研究工作主要防止匹配攻击所导致的隐私泄露,即防止攻击者基于攻击目标的背景知识在社交网络图中采用图匹配技术,从而唯一性确定攻击目标的位置或者边连接.目前,基于概率和

推演模型的隐私攻击[12−15]已经引起研究人员关注.文献[13]研究了如何基于社会网络签到数据采用概率模型推演用户的位置隐私,并设计了一种隐私风险预警框架,实时监控可能导致用户位置隐私泄露的位置签到并进行报警.文献[14]针对敏感链接推演攻击,分别提出了防止单步链接推演攻击和级联链接推演攻击算法.文献[15]研究了针对社会网络图中的敏感标签的推演攻击,并提出了防推演算法.文献[16]研究在图匿名过程中如何保持社会网络图在社区划分方面的数据可用性.

可达性查询是图数据上的一种基本查询,同时也是一项热点研究问题,目前,针对可达性查询已经展开了广泛研究.按照查询类别的不同,图可达性查询分为基本可达查询[17−23]、d-可达查询[24,25]、距离查询[26−28]等.其中:基本可达查询用于查询两结点间是否存在一条可达路径;所谓 d-可达查询,是指给定阈值 d,查看两结点是否在 d 距离内可达;距离查询在返回两结点间是否可达的同时,给出了可达结点间的最短距离.针对可达性查询,提出了一系列图可达性查询技术.处理可达性查询的两种直接方法分别是对图进行深度或宽度优先遍历来查看结点间是否存在可达路径和提前计算图 G 的传递闭包,并基于传递闭包返回查询结果.然而,这两种方法存在执行效率低、传递闭包的空间代价大等问题.为了解决上述问题,相关工作[17−28]通过构建可达性索引并基于索引来实现高效的可达性查询处理.可达性索引的本质是:采用不同方法来压缩图的传递闭包,并在索引创建时间、索引空间代价和查询性能之间进行权衡.可达性索引方法主要分为集合覆盖法[21,26,27]和区间标记法[17,19,20,22,23].

可以看出,当前研究工作忽视了在图匿名化过程中保持结点可达性,导致生成的匿名图在可达性查询方面的低数据可用性.本文针对此问题展开研究,提出一种保持结点可达性的高效社会网络图匿名算法,在匿名化社会网络图的同时保证匿名图在可达性查询方面的高数据可用性.

2 预备知识和问题定义

在本文中,社会网络表示为有向图 $G=(V,E)$且$|V|=n$ 和$|E|=m$,其中,V 表示结点集合,E 表示边集合;同时,$V(G)$和 $E(G)$同样分别表示图 G 的结点集和边集.结点对(u,v)表示从结点 u 指向 v 的边;边(u,v)称为 u 的出边、v 的入边;v 是 u 的出边邻居,u 是 v 的入边邻居.结点 u 的入边数目是 u 的入度,记作 $d^{\text{in}}(u)$;u 的出边数目是 u 的出度,记作 $d^{\text{out}}(u)$;结点 u 的度以$(d^{\text{in}}(u),d^{\text{out}}(u))$的形式来表示.当向 E 中添加边(u,v)时,也称连接结点 u 和 v.

2.1 社会网络隐私保护模型

在本文中,假设攻击者将攻击目标的出度和入度作为背景知识进行结点身份识别攻击.基于文献[1]中的度匿名隐私保护模型,给出本文中的社会网络隐私保护模型.

定义 1(k-度匿名). 已知图 $G(V,E)$和正整数 k,对于$\forall v\in V$,如果 G 中存在 $m(m\geqslant k-1)$个其他结点 $v_1,v_2,\ldots,v_m$ 符合 $d^{\text{in}}(v)=d^{\text{in}}(v_i)$和 $d^{\text{out}}(v)=d^{\text{out}}(v_i)(1\leqslant i\leqslant m)$,则称 G 为 k-度匿名图.

例如,图 2 中的 G_1 和 G_2 均为 2-度匿名图.本文主要研究了对有向图匿名时,如何保持结点间可达性.本文的研究方法同样适用于无向图,因为无向图是有向图的特殊形式.此外,本文所提出的算法除了可以在 k-度匿名隐私保护模型中保持结点间的可达性以外,还可以扩展至其他图匿名模型,包括邻接图匿名[2]、自同构图匿名[4]等.

2.2 问题定义

首先给出可达性的定义,本文期望在匿名图中尽量保持结点间的可达性不变.

定义 2(可达性). 已知图 $G(V,E)$,如果结点 u 到 v 之间存在一条路径,则称结点 u 到 v 可达,记作 $u\rightarrow v$,结点对$\langle u,v\rangle$称为可达对;将 G 的可达对集合记作 $R(G)$.

可达性查询是图研究领域的热点问题[17,18].在社会网络中,查询两个结点之间是否存在路径可达是一项基本操作,如果两点可达,则意味着两个用户之间具有某种关系.

已知图 $G(V,E)$和正整数 k,期望通过对 G 进行一系列的图修改操作生成一个 k-度匿名图 $G_k(V_k,E_k)$,并且 G_k 能够尽量保持 G 中结点间的可达性.在文献[4,11]中,为了实现 k-匿名,匿名图与原图的结点集和边集不完全相同,即 $V_k\approx V$ 和 $E_k\approx E$.本文进行图匿名化时仅考虑结点和边添加操作,即 $V\subseteq V_k$ 和即 $E\subseteq E_k$.问题 1 给出了可达性保持图匿名化问题.

问题 1(可达性保持图匿名化). 已知图 $G(V,E)$和正整数 k,构建一个 k-度匿名图 $G_k(V_k,E_k)$,使得 $Cost(G,G_k)$最小化.

由于在图匿名化过程中只考虑结点和边添加操作,因此 $R(G)\subseteq R(G_k)$.此时,公式(1)等于 $Cost(G,G_k)=|R(G_k)-R(G)|$,即 G_k 中增加的可达对数目.

3 可达性保持图匿名算法

本节介绍本文提出的可达性保持图匿名算法 RPA,算法 1 给出了具体过程.RPA 算法在对社会网络进行 k-度匿名的同时保持结点间的可达性.

算法 1 的基本思想是:将具有相近度的结点分配到一个分组中,并通过图修改操作使得同一分组中的结点具有相同的度.给定入度和出度,RPA 算法采用贪心策略匿名化每个结点,并最小化由于匿名操作而导致的可达性信息损失.由于社会网络中结点度符合幂律分布,因此,RPA 算法首先从具有高入度和出度的结点开始匿名,剩余的具有较小入度和出度的结点比较容易匿名.

算法 1 首先将所有结点标记为“unanonymized”,并将这些结点存储于 Set_{ua} 中(第 2 行、第 3 行).在每次迭代过程中,算法均选择 Set_{ua} 中具有 $d^{in}+d^{out}$ 最大值的结点作为种子结点 $Seed$(第 5 行).如果 Set_{ua} 中的结点数目大等于 $2k$,算法在 Set_{ua} 中选择 k 个距离 $Seed$ 最近的结点来生成集合 $vSet$(第 6 行、第 7 行).如果 Set_{ua} 中的结点数目小于 $2k$,则将 Set_{ua} 中的剩余结点来生成集合 $vSet$(第 9 行).其中,对于任意结点 u,基于 u 的度$(d^{in}(u),d^{out}(u))$将其映射到二维空间中,采用曼哈顿距离来衡量两个结点之间的距离.

算法 1. 可达性保持图匿名算法(RPA).

输入:社会网络图 $G=(V,E)$;匿名参数 k.

输出:匿名图 G_k.

1. $G_k\leftarrow G$;
2. 将 G_k 中结点标记为“unanonymized”;
3. Set_{ua}={G_k 中标记“unanonymized”的结点};
4. **Repeat**
5. $Seed\leftarrow Set_{ua}$ 中具有最大 $d^{in}(s)+d^{out}(s)$的结点 s;
6. **If** $|Set_{ua}|\geqslant 2k$ **Then**
7. $vSet\leftarrow Set_{ua}$ 中距离 $Seed$ 最近的 k 个结点;
8. **Else**
9. $vSet\leftarrow Set_{ua}$ 中剩余结点;
10. $d^{in}\leftarrow vSet$ 中结点的最大入度;
11. $d^{out}\leftarrow vSet$ 中结点的最大出度;
12. **For** $\forall u\in vSet$ **Do**
13. *anonymizeOutDegree*($G_k,u,d^{out},vSet$);
14. *anonymizeInDegree*($G_k,u,d^{in},vSet$);
15. 将结点 u 标记为“anonymized”,并从 Set_{ua} 中删除;
16. **Until** $Set_{ua}==\varnothing$;
17. **Return** G_k;

在图匿名过程中,如果只允许结点和边添加操作,则图中结点的入度和出度只会增加,不会减小.因此,算法 1 在匿名化 $vSet$ 中的结点时,采用 $vSet$ 中结点的最大入度和出度进行匿名(第 10 行~第 15 行).给定正整数 d^{out} 和结点 u,算法 2 采用贪心策略选择 $d^{out}-d^{out}(u)$个结点并连接 u 和这些结点,使得 u 的出度为 d^{out}.其中,每个被选结点 v 应该符合如下条件:

(1) $(u,v)\notin E(G_k)$;

(2) 结点 v 是未匿名结点.

在这两个条件中,条件(1)是结点 v 成为 u 的新邻居的必要条件,条件(2)保证了匿名化结点 u 不会影响其他结点的匿名状态.将此两条件称为候选邻居条件(记作 CNC),符合 CNC 的结点称为 u 的候选出边邻居.对于 u 的候选入边邻居 v,条件(1)则更改为$(v,u)\notin E(G_k)$.为了保持入度小于等于 d^{in},算法不会选择 *vSet* 中结点作为 u 的候选出边邻居.在算法 2 的第 2 行~第 5 行循环中,算法采用贪心策略将导致最小 $Cost(u,v)$的结点 v 作为 u 的新出边邻居,其中,$Cost(u,v)$是指在 G_k 中添加边(u,v)所导致的可达性信息损失,计算如公式(2):

$$Cost(u,v)=|R(G_k\cup\{(u,v)\})|-|R(G_k)| \tag{2}$$

在本文中,将此贪心策略称为单邻居添加策略,并在第 5 节中详细讨论.算法会将 u 的新出边邻居从 *candNeighbors* 中删除(第 5 行).当 *candNeighbors* 为空并且 d^{out} 仍然大于 $d^{\text{out}}(u)$时,算法 2 添加 $d^{\text{out}}-d^{\text{out}}(u)$个结点到 G_k 中,并连接 u 至这些新添加结点(第 6 行~第 8 行).显然,新添加结点的度为(1,0).根据度幂律分布可知,G_k 中具有度(1,0)的结点数目远大于匿名参数 k,因此可以安全地将这些结点标记为“anonymized”(第 7 行).对于结点的入度匿名过程与算法 2 相似,本文不再赘述.RPA 算法在图匿名过程中虽然考虑了特定优化指标,即,保持结点间的可达性,但是由于生成的匿名图符合 k-度匿名模型,基于结点度的隐私泄露概率小于等于 $1/k$,此种优化指标不会为攻击者提供背景知识从而导致额外的隐私泄露.

算法 2. anonymizeOutDegree.

输入:图 G_k;结点 $u\in V(G_k)$;整数 d^{out};结点集 *vSet*.

输出:匿名图 G_k.

1. *candNeighbors*$\leftarrow\{G_k$ 中的候选出边邻居$\}-$*vSet*;
2. **While** $d^{\text{out}}(u)<d^{\text{out}}$ && *candNeighbors*$\neq\varnothing$ **Do**
3. $u'\leftarrow$*candNeighbors* 中具有最小 $Cost(u,v)$的结点 v;
4. 在 $E(G_k)$中添加边(u,u');
5. 将 u'从 *candNeighbors* 中删除;
6. **If** $t=d^{\text{out}}-d^{\text{out}}(u)>0$ **Then**
7. 在 $V(G_k)$中添加 t 个结点并标记为“anonymized”;
8. 连接 u 和新添加结点;

RPA 算法在图匿名的同时保持了结点间的可达性,但是社会网络通常具有数以千万的结点和边,导致 RPA 算法的执行效率面临以下两个挑战:

(1) 已知图 G 具有 n 个结点和 m 条边,如何高效地评估任意图操作所导致的可达性信息损失?

在 RPA 算法中,评估添加边所导致的可达性信息损失是一项频繁操作,与 RPA 算法的执行效率密切相关.然而,此项操作非常耗时.例如,当评估在 G 中添加边(u,v)时的可达性信息损失时,一种简单方法就是计算图中每个结点的可达结点集合并比较其在边添加操作前后的变化.生成任意结点的可达结点集合需要一次宽度优先遍历,需要 $O(n+m)$时间.因此,评估一个边添加操作的可达性信息损失需要 $O(2n(n+m))$时间.对于结点 u 来说,由于在 *candNeighbors* 中至多包含$(n-1)$个结点,因此算法 2 需要 $O(2n^2(n+m))$时间来寻找使得 $Cost(u,v)$最小的结点 v(第 3 行).对于大型社会网络数据,如此高的时间复杂度使得此项操作不具有可行性.作为 RPA 算法的一项基本操作,对如何高效地评估边添加操作导致的可达性信息损失提出了一项挑战.

(2) 当需要将结点 u 的出度增加 3 时,RPA 算法如何在图中高效地搜索 u 的 3 个新出边邻居,使得可达性信息损失最小?

显然,候选出边邻居的组合数目为 $\binom{n-d^{\text{out}}(u)}{3}$. 当社会网络图中结点数目很大时,RPA 算法不可能通过枚举的方式选择最优组合.因此,匿名结点时如何高效地选择最小化可达性信息损失的新出边邻居是一个挑战性问题.

为了降低 RPA 算法的时间复杂度,使其具有可行性,本文第 4 节、第 5 节主要研究两个问题:

(1) 如何高效评估 $Cost(u,v)$?

(2) 对于结点 u,如何高效地寻找结点 v,使得 $Cost(u,v)$最小?

4 高效评估图匿名损失

本节研究如何高效地评估 $Cost(u,v)$.首先提出可达区间的定义,然后给出基于可达区间的匿名损失高效评估方法.

4.1 采用可达区间评估匿名损失

给定图 $G(V,E)$和结点 $p\in V$,$R(p)$表示 p 在图 G 中可达的结点集合,$R^{-1}(p)$表示 G 中可达 p 的结点集合.为了方便讨论,假设每个结点到其自身可达,即 $p\in R(p)$和 $p\in R^{-1}(p)$.由于在图匿名过程中只采用结点和边添加操作,采用$\Delta R(p)$来表示由于添加边(u,v)而导致 $R(p)$所增加的结点集合.显然,当添加边(u,v)时,并不是所有结点的可达结点集都会受到影响.

定理 1. 在图 G 中添加边(u,v),可达结点集受影响的结点属于 $R^{-1}(u)$.

证明:反证法.当在图 G 中添加边(u,v)时,假设结点 p 的可达结点集受影响并且 $p\notin R^{-1}(u)$.令结点 p 的可达结点集 $R(p)$中增加可达结点的集合为$\Delta R(p)$.对于$\forall q\in\Delta R(p)$,则存在一条从 p 至 q 的路径,并且该路径经过边(u,v).显然,结点 p 至 u 是可达的,则 $p\in R^{-1}(u)$,与假设 $p\notin R^{-1}(u)$矛盾. □

根据定理 1,可以通过检验每个结点 $p\in R^{-1}(u)$的$\Delta R(p)$来计算匿名代价 $Cost(u,v)$.此时,公式(2)等价于采用公式(3)来计算:

$$Cost(u,v)=\sum_{p\in R^{-1}(u)}|\Delta R(p)| \tag{3}$$

为了高效地计算$\Delta R(p)$,提出可达区间数据结构.首先为图中的每个结点分配一个 id,然后,基于结点的 id 来生成每个结点的可达区间.

定义 3(可达区间). 对于任意结点 u,u 的可达区间由若干整数区间组成,记录了 u 在图中可达结点的 id 范围,记作 $R_I(u)$.

存在多种结点 id 分配方式,其中一种简单方法是随机分配方法,如图 3 所示,图中结点被随机分配 id,位于区间$[0,n)$内.此时,结点 f 和 g 的可达区间为

$$R_I(f)\rightarrow[4,6)\cup[8,10)\cup[13,14)\cup[15,16);$$

$$R_I(g)\rightarrow[1,2)\cup[4,10)\cup[11,12)\cup[13,14)\cup[15,16).$$

令 p_{id} 表示结点 p 的 id,对于结点 u 和 p,显然,$p_{id}\in R_I(u)$当且仅当 $u\rightarrow p$.因此对于结点 p,$|\Delta R(p)|=|\Delta R_I(p)|$成立.此时,可以通过计算$\Delta R_I(p)$来代替公式(3)中的$\Delta R(p)$.

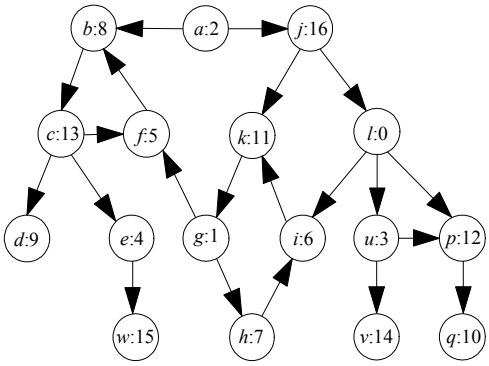


Fig.3 Random id assignment

图 3 随机结点 id 分配

定理 2. 在图 G 中添加边(u,v),对于结点 $p\in R^{-1}(u)$,可得$\Delta R_l(p)=R_l(v)-R_l(p)$.

证明:反证法.在添加边(u,v)后,假设 p 的可达区间由 $R_l(p)$变为 $R_l'(p)$.显然可知:

$$\Delta R_l(p)=R_l'(p)-R_l(p)\supseteq R_l(v)-R_l(p).$$

假设存在结点 q 满足 $q_{\mathrm{id}}\notin R_l(v)-R_l(p)$和 $q_{\mathrm{id}}\in\Delta R_l(p)$,可知添加边$(u,v)$使得 $p\to q$,因此,边(u,v)位于从 p 到 q 的路径上,从而可以推断 $v\to q$.由 $q_{\mathrm{id}}\in R_l(v)$和 $q_{\mathrm{id}}\notin R_l(v)-R_l(p)$可知 $q_{\mathrm{id}}\in R_l(p)$,与假设 $q_{\mathrm{id}}\in\Delta R_l(p)$矛盾,因此,结点 q 不存在并且$\Delta R_l(p)=R_l(v)-R_l(p)$. □

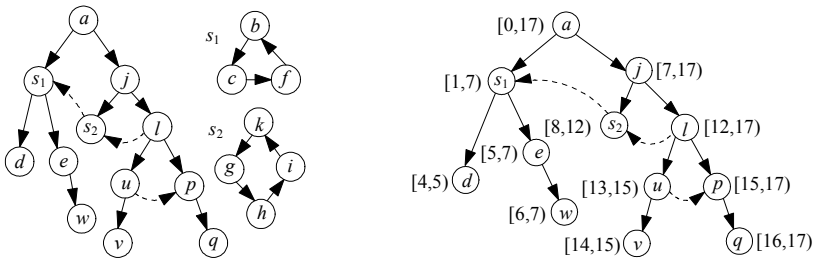
采用结点 id 随机分配方式时,对于任意结点 p,$R_l(p)$存在至多 $\frac{n}{2}$ 个整数区间,因此,计算$\Delta R_l(p)$需要 $O(n)$时间.结点 id 的分配方式决定了可达区间的大小,而可达区间的大小决定了计算$\Delta R_l(p)$的时间复杂度.

4.2 生成结点的可达区间

为了高效地计算$\Delta R_l(p)$,本文扩展了文献[19]中的 Dual Labeling 方法来生成结点的可达区间,从而在 $O(t)$ $(t<<n)$时间内完成$\Delta R_l(p)$的计算.

Dual Labeling 方法主要包含两步:(1) 基于输入图构建一个生成树;(2) 为每个结点分配一个区间标签,并记录非树边的连接情况,从而保证图可达性信息的完整性.

真实社会网络通常包含环,已知图 $G(V,E)$且$|V|=n$ 和$|E|=m$,找出 G 的强连通分量,并将每个分量以一个超点来表示.采用文献[29]中的 Tarjan 算法,可以在 $O(n+m)$时间内完成此超点生成过程.对于图 1(a)中的 G,图 4(a)显示了对应的生成树 T和所包含的超点.在图 4(a)中,生成树由实线箭头所组成,虚线箭头表示非树边.显然,位于同一超点内的结点具有相同的可达结点集.本文假设社会网络图是强连通图,如果不是,则通过设定一个虚拟根结点并将各连通分量连接起来即可.



(a) 生成树 T 及超点 (b) 区间标签

Fig.4 Extended Dual Labeling coding scheme

图 4 扩展 Dual Labeling 方法

由于生成树 T 上的某些结点为超点并包含多个图结点,为了准确计算匿名损失,本文对文献[19]中的 Dual Labeling 技术进行扩展来分配结点 id.对生成树 T 进行一次遍历,每个结点 u 被赋值区间标签$[u_{\mathrm{start}},u_{\mathrm{end}})$.结点区间标签的具体赋值方法为:(1) u_{start} 被赋值为 $u'_{\mathrm{start}}+cnt(u')$, 其中,$u'$是 u 的前序父结点,$cnt(u')$等于 u'中所包含的结点数目(当 u 为 T 根结点时,$u_{\mathrm{start}}=0$);(2) 当 u 是 T 的非叶子结点时,u_{end} 被赋值 u''_{end},,其中,u''是 u 的后序父结点;(3) 当 u 是 T 的叶子结点时,u_{end} 被赋值 $u_{\mathrm{start}}+cnt(u)$.此时,将 u_{start} 作为结点 u 的 id.图 4(b)显示了生成树 T 上的结点区间标签.采用传递链接表 L 来记录非树边的传递闭包,L 具体为:

$$8\to[1,7),$$
$$12\to[8,12),$$
$$13\to[15,17),$$
$$12\to[1,7).$$

基于结点 u,v 的区间标签和传递链接表 L,可达性查询 $u\to v$ 可以在 $O(1)$时间内完成,具体细节参见文献[19],

本文不再赘述.

对于生成树 T 上的结点 u,算法 3 为其生成一个可达区间 $R_I(u)$,包含了 u 通过树边和非树边可达的结点 id 区间.显然,在树结点 u 中,每个图结点的可达区间是 $R_I(u)$.假设 T 中包含 t 个非树边,则 L 至多包含 $\frac{t(t+1)}{2}$ 条记录,因此,算法 3 的时间复杂度是 $O(t^2)$.显然,可达区间 $R_I(u)$至多包含 t+1 个区间.例如,图 4 中结点 f 和 g 的可达区间为:

$$R_I(f)\rightarrow[1,7),$$
$$R_I(g)\rightarrow[1,7)\cup[8,12).$$

算法 3. 可达区间生成算法.

输入:生成树结点 u;传递链接表 L.

输出:可达区间 $R_I(u)$.

1. $R_I(u)\leftarrow[u_{\text{start}},u_{\text{end}})$;
2. **For** each $v_{\text{start}}\rightarrow[v'_{\text{start}},v'_{\text{end}})$ **Do**
3. **If** $v_{\text{start}}\in[u_{\text{start}},u_{\text{end}})$ **Then**
4. $R_I(u)\leftarrow R_I(u)\cup[v'_{\text{start}},v'_{\text{end}})$; ;
5. **Return** $R_I(u)$;

文献[19]中提出,通过寻找最小等价图可以使得非树边的数目最小.当向图中添加边(u,v)时,对于结点 $p\in R^{-1}(u)$,由于 $R_I(v)$和 $R_I(p)$至多包含 t+1 个区间,因此需要 $O(t)$时间来计算$\Delta R_I(p)=R_I(v)-R_I(p)$.令 $r=|R^{-1}(u)|$,则计算公式(3)需要 $O(rt)$时间,r 和 t 在真实社会网络中满足 $r<<n$ 和 $t<<n$.

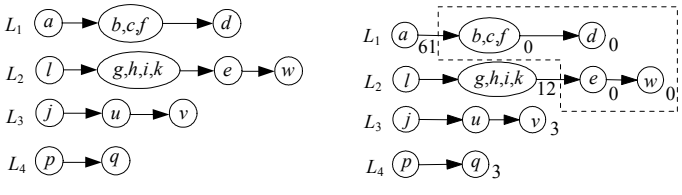
5 优化结点匿名效率

在匿名化结点 u 时,算法 2(第 3 行)需要 $O(nrt)$时间来寻找具有最小 $Cost(u,v)$数值的结点 v.为了加速此过程,本文采用文献[20]中的 Chain Cover 算法生成的索引结构,并基于此索引为结点匿名提出两种结点添加策略:单邻居添加策略和多邻居添加策略.

如文献[20]中所描述,拟采用的索引结构是一个链表集合,包含了链表 $L_1,L_2,\ldots,L_s$,链表 $L_i=s_{i1}\rightarrow s_{i2}\rightarrow\ldots\rightarrow s_{in_i}$,其中,

- 链表结点 s_{ij} 包含了图 G 中多个结点;
- 所有链表结点的并集等于 G 的结点集合,并且任意两个链表结点的交集为空;
- 任意链表 L_i 符合:
  (1) 对于结点$\forall u\in s_{ij}$ 和$\forall v\in s_{i(j+1)}$,满足 $u\rightarrow v$ 并且 $v\nrightarrow u$;
  (2) 对于结点$\forall u,v\in s_{ij}$,满足 $u\rightarrow v$ 并且 $v\rightarrow u$.

为描述方便,本文将该索引称为候选邻居索引(记作 CN-index).文献[20]给出了 CN-index 的构建方法以及当发生图操作时如何动态维护 CN-index,本文不再赘述.例如,对于图 1(a)中的 G,图 5(a)显示了与其对应的一个 CN-index.



(a) CN-index (b) 为结点 b 选择 out-OCN

Fig.5 Selecting the out-OCN using the CN-index

图 5 基于 CN-index 选择 out-OCN

给定图结点 u 和 CN-index 中的索引结点 n_s,假设结点 v 属于 n_s,则将添加边$(u,v)/(v,u)$称作连接 u 至 n_s(或 n_s 至 u).下面介绍如何基于 CN-index 来加速结点匿名.

5.1 单邻居添加策略

为了给结点 u 添加 k 个出边邻居,本节提出了单邻居添加策略.算法 2 进行结点匿名时采用了单邻居添加策略.单邻居添加策略的基本思想是:为结点 u 不断地选择和添加导致最小匿名损失的候选出边邻居,直至新添加的邻居数目为 k.

定义 5(最优候选出边/入边邻居(out/in-OCN)). 已知图 G 和结点 $u\in V$,如果结点 v 是 u 的候选出边/入边邻居并且具有最小 $Cost(u,v)/Cost(v,u)$,则称 v 是 u 的最优候选出边/入边邻居,记作 out/in-OCN.

单邻居添加策略的关键是查找结点 u 的 out/in-OCN.根据定理 3,可以基于 CN-index 来加速 out/in-OCN 查找过程.

定理 3. 如果 $u\rightarrow v$ 并且 $v\nrightarrow u$,对于结点 p,则 $Cost(p,v)\leqslant Cost(p,u)$.

证明:由于 $u\rightarrow v$ 并且 $v\nrightarrow u$,显然可得 $R_l(v)\subseteq R_l(u)$.如果 $p\rightarrow u$,则 $Cost(p,v)=Cost(p,u)=0$;如果 $p\nrightarrow u$,则

$$Cost(p,v)=\sum\nolimits_{q\in R^{-1}(p)}|R_l(v)-R_l(q)|<\sum\nolimits_{q\in R^{-1}(p)}|R_l(u)-R_l(q)|=Cost(p,u).$$ □

算法 4 显示了基于 CN-index 来查找结点 u 的 out-OCN.算法首先在 $R(u)$中查找 out-OCN(第 2 行),这是因为 $R(u)$中的结点不会导致匿名损失.如果 $R(u)$中没有候选出边邻居,则从 CN-index 中每个链表的尾部开始查找 out-OCN(第 3 行~第 14 行).对链表 L_i 进行查找时,如果遇到候选出边邻居结点,则在该链表的查找过程停止.算法将具有最小匿名代价的候选出边邻居存储于 *ocnset*.当在 *ocnset* 中选择 out-OCN 时,选择具有最小入度的结点作为 out-OCN(第 15 行),从而保持度幂律分布.

算法 4. 选择 out-OCN.

输入:结点 u.

输出:结点 u 的 out-OCN.

```
1.  ocn←null;
2.  ocnset←R(u)中的候选出边邻居;
3.  If ocnset==∅ Then
4.    cost←+∞;
5.    For CN-index 中每个链表 L_i Do
6.      For j=n_i to 1 Do
7.        candset←s_ij 中的候选出边邻居;
8.        If candset≠∅ Then
9.          If cost>Cost(u,candset) Then
10.           cost←Cost(u,candset);
11.           ocnset←candset;
12.         If cost==Cost(u,candset) Then
13.           ocnset←ocnset∪candset;
14.         Break;
15. ocn←ocnset 中具有最小入度的结点 v;
16. Return ocn;
```

令 $r=|R^{-1}(u)|$,由于 CN-index 中包含 s 个链表,则算法 4 的时间复杂度为 $O(srt)$,其中,t 表示生成树中非树边的数目.

例如在图 1(a)中,当为结点 b 选择 out-OCN 时,图 5(b)显示了对应的 CN-index.当 G 中的结点未匿名时,将在 $R(b)$中选择候选出边邻居,如虚线框中所示的 d,e,f 和 w.如果这 4 个结点已经匿名,则在 CN-index 的每个链表结

尾开始查找 out-OCN.如图 5(b)所示,经过检验的索引结点被标记上了连接 b 和该索引结点的匿名代价.在此示例中,结点 q 和 v 具有最小的匿名代价,可以随机选择任意结点作为 out-OCN.

为结点 u 选择 in-OCN 的过程与算法 4 相似,本文进行简单介绍.

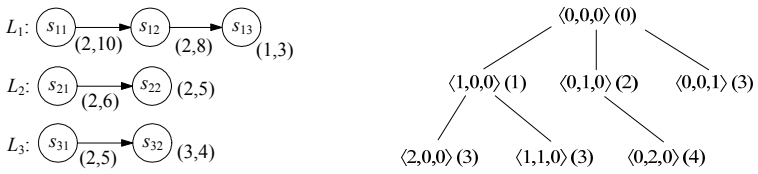
定理 4. 如果 $u\to v$ 并且 $v\nrightarrow u$,对于结点 p,则 $Cost(u,p)\leqslant Cost(v,p)$.

根据定理 4,应该优先选择 $R^{-1}(u)$中的候选入边邻居作为 u 的 in-OCN.如果不存在此种候选入边邻居,则从 CN-index 中每个链表的头结点开始查找 in-OCN.当从具有最小匿名代价的候选入边邻居中选择 in-OCN 时,选择具有最小出度的结点作为 in-OCN.

5.2 多邻居添加策略

为结点 u 添加 $k(k>1)$个出边邻居时,单邻居添加策略具有很高的执行效率,但是导致较大的匿名损失.为了解决此问题,本文提出了多邻居添加策略.

例如,当为结点 u 添加 3 个出边邻居时,图 6(a)显示了对应的 CN-index.在 CN-index 中,每个链表结点 s 标记上了(n_o,n_c),其中,n_o 和 n_c 分别表示 s 中候选出边邻居数目和连接 u 至 s 的匿名损失.为了方便讨论,假设不同链表上结点的匿名损失数值相互独立,即,连接 u 至 L_i 上的结点不影响 $L_j(i\neq j)$上结点的匿名损失数值.当采用单邻居添加策略时,首先连接 u 至 s_{13} 中的一个候选出边邻居,导致的匿名损失为 3;然后,连接 u 至 s_{32} 中的两个候选出边邻居,导致的匿名损失为 4.因此,采用单邻居添加策略导致的匿名损失为 3+4=7.与单邻居添加策略不同,另外一种邻居添加方法是连接 u 至 s_{32} 中的 3 个候选出边邻居,导致的匿名损失为 4.可见,此方法的匿名损失小于单邻居添加策略.



(a) CN-index (b) 枚举树 T_e

Fig.6 An example of adding multiple out-neighbors

图 6 添加多个出边邻居示例

定义 6(最优候选出边/入边邻居集(out/in-OCNS)). 已知图 $G(V,E)$和结点 $u\in V$,假设在图匿名过程中需要为 u 添加 k 个出/入边邻居,令 $V_c\subseteq V$ 是 u 的候选出/入边邻居集,如果 $V_o\subseteq V_c$ 满足$|V_o|=k$ 并且对于$\forall V'\subseteq V_c(|V'|=k)$都有 $Cost(u,V_o)<Cost(u,V')/Cost(V_o,u)<Cost(V',u)$,则 V_o 是 u 的最优候选出边/入边邻居集,记作 out/in-OCNS.

在定义 6 中,$Cost(u,V_c)$表示连接 u 至 V_c 中每个结点所导致的匿名损失.多邻居添加策略的主要思想是,枚举所有可能候选出边/入边邻居集并选择导致最小匿名损失的集合作为 out/in-OCNS.本文设计一种剪枝-枚举算法来生成 out/in-OCNS.

5.2.1 评估多边添加匿名损失

首先介绍如何评估添加多边所导致的匿名损失.对于结点 u 和 v,显然,边(u,v)的添加不会影响 $R^{-1}(u)$.当连接 u 至多个候选出边邻居时,对于结点 $p\in R^{-1}(u)$,根据定理 5 来计算$\Delta R_l(p)$.

定理 5. 当连接 u 至 CN-index 的链表结点 $s_1,\ldots,s_m$ 时,对于结点 $p\in R^{-1}(u)$, $\Delta R_l(p)=\bigcup_{i=1}^{m}R_l(s_i)-R_l(p)$.

定理 6 中指出,计算$\Delta R_l(p)$时没有必要考虑 u 连接的所有链表结点.

定理 6. 当连接 u 至 CN-index 的链表结点 s_{ij} 和 $s_{il}(j<l)$时,对于结点 $p\in R^{-1}(u)$,$\Delta R_l(p)=R_l(s_{ij})-R_l(p)$.

证明:给定结点 $p\in R^{-1}(u)$,首先连接 u 至链表结点 s_{ij},此时$\Delta R_l(p)$为 $R_l(s_{ij})-R_l(p)$,并且可知 $p\to s_{il}$,因此,连接 u 至 s_{il} 不会影响 $R_l(p)$. □

根据定理 6,在计算$\Delta R_l(p)$时,只需考虑每个链表 L_i 中被连接且具有最小 j 值的链表结点 s_{ij}.例如在图 6(a)中,

假设连接结点 u 至链表结点 $s_{12},s_{13},s_{21},s_{22},s_{32}$ 时,对于结点 $p\in R^{-1}(u)$,则 p 的可达区间变化为

$$\Delta R_I(p)=R_I(s_{12})\cup R_I(s_{21})\cup R_I(s_{32})-R_I(p).$$

5.2.2 采用剪枝枚举来生成 out-OCNS

对于链表 L_i,层次 l_i 表示链表结点 $s_{i(n_i-l_i+1)}$ 至 s_{in_i} 的候选邻居集合,而层次序列 $S=\langle l_1,l_2,\ldots,l_s\rangle$ 表示序列中每个链表层次所规定的候选邻居集合的并集.对于层次序列 $S=\langle l_1,l_2,\ldots,l_s\rangle$ 和 $S'=\langle l'_1,l'_2,\ldots,l'_s\rangle$,当 $\exists i\in[1,s]$ 使得 $l_i<l'_i$ 并且 $l_j\leqslant l'_j(j\neq i)$ 时,则称 S 和 S' 满足偏序关系 $S\prec S'$.偏序关系 $\prec$ 具有传递性,即,满足 $S\prec S'$ 和 $S'\prec S''$ 时,则 $S\prec S''$.例如在图 6(a)中,层次序列 $\langle 2,1,0\rangle$ 表示链表结点 s_{12},s_{13},s_{22} 中的候选出边邻居集合,并且可得 $\langle 1,1,0\rangle\prec\langle 2,1,0\rangle$.给定结点 u 和层次序列 S,$Cost(u,S)$ 表示连接 u 至 S 中每个结点所导致的匿名损失.

本文设计一种剪枝枚举方法来生成 out-OCNS,剪枝枚举具体包括 3 个步骤:

(1) 给定结点 u,枚举所有包含至少 k 个候选出边邻居的层次序列 S 并计算 $Cost(u,S)$,选择最小匿名损失的层次序列作为最优层次序列;

(2) 给定最优层次序列 S_o,通过不断提高 S_o 中的链表层次来获得 S'_o,提高链表层次过程中,使得 $S_o\prec S'_o$ 并且 $Cost(u,S_o)=Cost(u,S'_o)$;

(3) 在 S'_o 中选择 k 个结点来生成 u 的 out-OCNS.

剪枝枚举需要枚举出所有包含至少 k 个候选出边邻居的层次序列.在枚举过程中,可以采用如下性质来对层次序列进行剪枝:对于任意两个层次序列 S 和 S',如果 $S\prec S'$,根据定理 6 可知 $Cost(u,S)<Cost(u,S')$,因此,当 S 包含至少 k 个候选出边邻居时,可以安全地将 S' 剪枝.在图 6(b)中,枚举树 T_e 显示了剪枝枚举方法所检验的层次序列.在 T_e 中,每个树结点表示一个层次序列,在树结点上标记了其所包含的候选邻居数目,剪枝枚举方法在 T_e 的叶子结点中选择最优层次序列.例如在图 6 的示例中,由于层次序列 $S=\langle 1,1,0\rangle$ 已经包含了 3 个候选出边邻居,因此剪枝掉了层次序列 $S_1=\langle 1,1,1\rangle$ 和 $S_2=\langle 1,2,0\rangle$.如图 6(b)所示,如果枚举所有的层次序列,则需要评估 3×2×2=12 个层次序列,而剪枝枚举方法检验和评估了 4 个层次序列.

选择最优层次序列问题等价于“小盒放球”问题:将 k 个相同的球放入 s 个不同盒子中,有多少种方法?该问题的答案是 $\binom{k+s-1}{s-1}$.由于 $\binom{k+s-1}{s-1}=\frac{\prod_{i=0}^{k-1}(i+s)}{k!}\leqslant\frac{(k+s)^k}{k!}$,根据 Stirling 近似公式可知:

$$k!=\sqrt{2\pi k}\left(\frac{k}{e}\right)^k e^{\lambda_k}\left(\frac{1}{12k+1}<\lambda_k<\frac{1}{12k}\right).$$

因此,对于 $\forall k$ 均有 $k!>\sqrt{2\pi k}\left(\frac{k}{e}\right)^k$,从而可得 $\frac{(k+s)^k}{k!}<\frac{(k+s)^k}{\sqrt{2\pi k}\left(\frac{k}{e}\right)^k}=\frac{1}{\sqrt{2\pi k}}\left(e+\frac{se}{k}\right)^k$.选择最优层次序列的时间复杂度为 $O\left(\left(e+\frac{se}{k}\right)^k rt\right)$,其中,$r=|R^{-1}(u)|$,$t$ 为生成树中非树边数目.

给定最优层次序列 S_o,剪枝枚举方法查找满足如下条件的层次序列 S'_o:

(1) $S_o\prec S'_o$ 并且 $Cost(u,S_o)=Cost(u,S'_o)$.

(2) 对于 $\forall S''_o$ 满足 $S'_o\prec S''_o$,均有 $Cost(u,S'_o)<Cost(u,S''_o)$.该步骤的目的在于使得 S'_o 包含尽可能多的候选出边邻居,从而连接 u 至 S'_o 中的候选出边邻居时可以选择更多具有小入度的结点.

为了寻找 S'_o,首先基于 S_o 来计算 $R_I(u)$.在 CN-index 的链表 L_i 中,采用二分查找来获得具有最小 j 值并且 u 可达的链表结点 s_{ij},然后,将层次 l_i 增加至 n_i-j+1.在链表 L_i 上进行二分查找需要 $O(\log_2(n_i))$ 时间,因此,获得 S'_o 的时间复杂度为 $O(s\log_2(n))$.

定理 7. 当为结点 u 添加 k 个出边邻居时,$\forall C\subseteq S'_o$ 且 $|C|=k$ 是 u 的 out-OCNS.

证明:令 C 为包含 S'_o 中 k 个结点的任意子集.根据剪枝枚举方法可知,S_o 是具有最小 $Cost(u,S_o)$ 的层次序列.由于 $Cost(u,S_o)=Cost(u,S'_o)$ 并且 $C\subseteq S'_o$,可知 $Cost(u,C)=Cost(u,S'_o)$.因此,连接 u 至 C 中每个结点会导致最小匿

名损失,显然,C 是 u 的 out-OCNS. □

基于 S_o 生成 u 的 out-OCNS 时,为了保持度幂律分布,将 S'_o 中的候选出边邻居按照入度进行升序排列,并选择前 k 个结点作为 u 的 out-OCNS.

5.2.3 生成 in-OCNS

在介绍如何生成 in-OCNS 之前,首先讨论为结点 u 添加多个入边邻居所导致的匿名损失.

对于结点 u 和 v,添加边(u,v)不会影响 $R(u)$.假设 S 是候选入边邻居集合,当连接 S 中每个结点至 u 时,导致的匿名损失可计算为

$$Cost(S,u)=\sum_{p\in R^{-1}(S)}|R_l(u)-R_l(p)| \tag{4}$$

其中, $R^{-1}(S)=\bigcup_{v\in S}R^{-1}(v)$.对于 L_i 中的链表结点 s_{ij} 和 $s_{il}(j<l)$,显然 $R^{-1}(s_{ij})\subseteq R^{-1}(s_{il})$.如定理 8 所示,基于 CN-index 获得 $R^{-1}(S)$.

定理 8. 给定结点集合 S 并连接 S 中每个结点至 u,对于 CN-index 的链表 L_i 上包含 S 中结点的链表结点 s_{ij},令 s_{ic_i} 表示具有最大 j 值的链表结点,则 $R^{-1}(S)=\bigcup_{i=1}^{s}R^{-1}(s_{ic_i})$.

在图 6 中,当连接链表结点 $s_{12},s_{13},s_{21},s_{22},s_{31},s_{32}$ 至 u 时,可达性受影响的结点集合为

$$R^{-1}(S)=R^{-1}(s_{13})\cup R^{-1}(s_{22})\cup R^{-1}(s_{32}).$$

当为结点 u 生成 in-OCNS 时,层次 l_i 表示了链表 L_i 上的结点 s_{i1} 至 s_{il_i} 中的候选入边邻居集合,而层次序列 $S=\langle l_1,l_2,\ldots,l_s\rangle$表示了每个层次所代表的候选入边邻居集合的并集;对于任意的层次序列 S 和 S',如果 $S\prec S'$,则 $Cost(S,u)<Cost(S',u)$.此时,可以采用剪枝枚举方法来选择最优层次序列并生成 in-OCNS.

5.3 考虑伪点的匿名损失评估

为了方便讨论,之前在评估匿名损失时没有考虑新添加结点(即伪点)的影响.对于结点 v 和伪点 v',如果 $v\to v'$(或者 $v'\to v$),则$\Delta R(G)$应该包含可达对$\langle v,v'\rangle$(或者$\langle v',v\rangle$).

对于结点 p,令 $f_i(p)$和 $f_o(p)$分别表示 p 的入边邻居伪点和出边邻居伪点.通过扩展公式(3),公式(5)给出了当考虑伪点时如何计算匿名代价 $Cost(u,v)$:

$$Cost(u,v)=\sum_{p\in R^{-1}(u)}(f_i(p)+1)\left(|\Delta R_l(p)|+\sum_{q_{id}\in\Delta R_l(p)}f_o(q)\right) \tag{5}$$

其中,$\Delta R_l(p)=R_l(v)-R_l(p)$.如表 1 中的“分类”列所示,将匿名图中增加的可达对分为 4 类.例如,如果可达对$\langle v,v'\rangle$是$\langle$fake,true$\rangle$种类,则表明 v 是一个伪点,v'是一个真结点.通过扩展公式(5),可以得到 4 个因式,表 1 列出了每个因式所代表的可达对种类.

Table 1 Incremental reachable pair categories represented by terms in Eq.(5)

表 1 公式(5)中因式所代表的新增可达对种类

种类	可达对数目
⟨fake,true⟩	$f_i(p)\lvert\Delta R_l(p)\rvert$
⟨fake,fake⟩	$f_i(p)\sum_{q_{id}\in\Delta R_l(p)}f_o(q)$
⟨true,true⟩	$\lvert\Delta R_l(p)\rvert$
⟨true,false⟩	$\sum_{q_{id}\in\Delta R_l(p)}f_o(q)$

6 实验分析

本文对提出的 RPA 算法进行性能分析和评价.在测试过程中,采用了两个真实社会网络图数据集 Eu-Email, Epinions 进行实验分析和测试.

6.1 实验设置

Eu-Email 网络是基于欧洲一个研究中心的电子邮件数据所生成,时间范围是 2003 年 10 月~2005 年 5 月.每个结点对应一个电子邮件地址,一条有向边(i,j)表示至少有一封邮件由 i 发给 j.Eu-Email 图数据集包含 265 214 个结点和 420 045 条边.Epinions 网络是一个用户在线评论网络,有向边(u,v)表示了 u 信任 v,该数据集包含了 75 879 个结点和 508 837 条边.表 2 显示了图数据集相关统计数据.

Table 2 Statistics of datasets

表 2 数据集相关统计数据

	Eu-Email	Epinions
结点数目	265 214	75 879
边数目	420 045	508 837
最大入度	7 631	3 035
最大出度	930	1 801
平均出/入度	1.58	6.71
(0,1)度结点数目	170 768	18 328
(1,0)度结点数目	36 922	11 774
平均聚集系数	0.0671	0.1378
三角数目	267 313	1 624 481
直径(最长最短路径)	14	14

本文实现了两个版本的 RPA 算法:Single-RPA 和 Multi-RPA,分别在图匿名过程中采用了单邻居添加策略和多邻居添加策略.除了实现 RPA 算法以外,为了进行对比,本文对文献[2]中的图匿名算法进行了修改并实现,记作 Neighbor-Greedy.实现 Neighbor-Greedy 时,主要修改了文献[2]中的匿名损失函数,将其中评价两个结点的邻居图相似度部分替换为评价两个结点的度相似度.

本文测试了算法的执行时间、匿名损失、添加的边和结点数目、图结构信息损失.在测试中,匿名参数 k 的取值为 10,20,30,40,50.实验测试的软硬件环境如下:

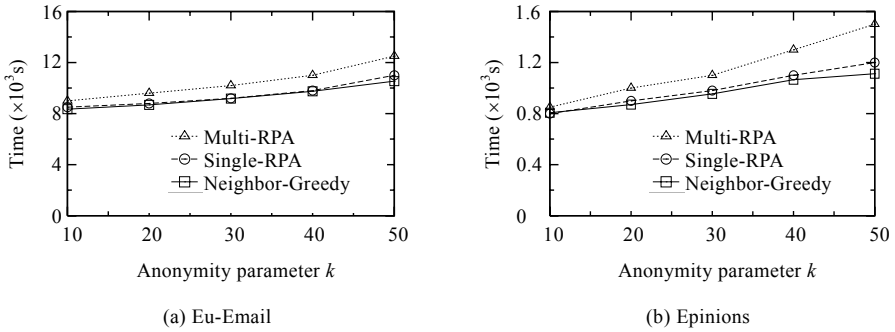
(1) 硬件环境:Intel Core 2 Duo 2.33GHz CPU,4GB DRAM 内存.

(2) 操作系统平台:Microsoft Windows XP.

(3) 编程环境:Java,Eclipse.

6.2 执行时间

本文测试了算法的执行时间随着 k 值的变化情况,实验结果如图 7 所示.从实验结果中可以看出,执行时间随着 k 值的增大而增加.由于需要生成 out/in-OCNS,Multi-RPA 算法的执行时间最高.从理论上来说,Single-RPA 算法中耗时的 out/in-OCN 查找操作会使得 Single-RPA 的执行效率低于 Neighbor-Greedy,然而图 7 中显示,这两个算法的执行时间基本相同,这是因为 CN-index 提高了 out/in-OCN 查找操作的效率.



(a) Eu-Email (b) Epinions

Fig.7 Runtime of graph anonymization

图 7 图匿名执行时间

6.3 匿名损失

给定图 G 和其 k-度匿名图 G_k,本文定义增量率 *incremental ratio*=$\frac{|R(G_k)-R(G)|}{|R(G_k)|}$ 来评估匿名图在结点可达性上的信息损失,可见增量率的数值范围为[0,1].其中,$R(G_k)$包含了 G_k 中的所有可达对,包括含有伪点的新增可达对.显然,增量率越大,则可达性信息损失越大.在图 8 中给出了实验结果.

从实验结果中可以看出,Neighbor-Greedy 的增量率最大,比 Single-RPA 和 Multi-RPA 平均高出 0.25.这是因为在 Neighbor-Greedy 算法中,忽视了边修改操作对于结点可达性的影响.与 Single-RPA 算法相比,Multi-RPA 算法具有更低的增量率,表明多邻居添加策略比单邻居添加策略导致更小的匿名损失.图 8 显示了 RPA 算法的增量率平均低于 0.02,证明了 RPA 算法可以很好地保持匿名图在可达性方面的数据可用性.

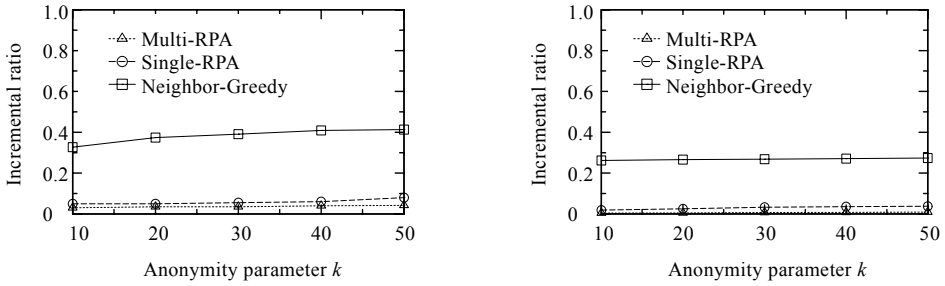


Fig.8 Incremental ratios

图 8 增量率

6.4 添加边和结点数目

为了评估 G_k 中新添加边所占比例,定义了边添加率 *adding edge ratio*=$\frac{|E(G_k)-E(G)|}{|E(G_k)|}$, 在图 9 中给出了实验结果.从图 9 中可以看出,由于 Neighbor-Greedy 算法在图匿名过程中采用添加边数目来评价匿名损失,因此其边添加率最低.在 RPA 算法中由于采用了基于结点度进行聚类,从而将具有相近度的结点匿名化为相同度结点,因此在实验结果中可以看出,Single-RPA 和 Multi-RPA 算法的边添加率与 Neighbor-Greedy 非常接近.

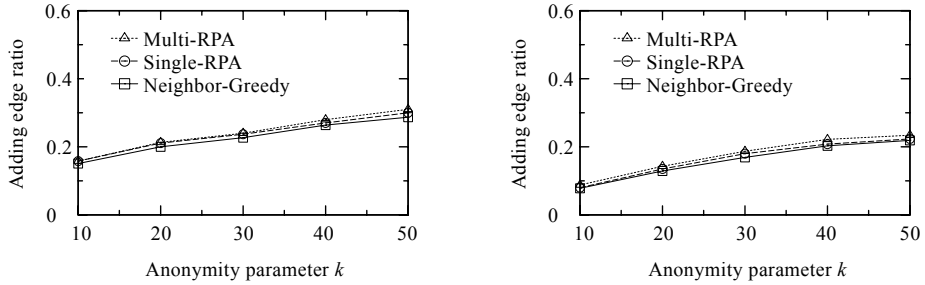


Fig.9 Adding edge ratios

图 9 边添加率

表 3 中显示了匿名图中伪点的添加数目.在任意测试数据集中,每种算法的伪点添加数目均不超过 70 个,而匿名图中度为(0,1)或(1,0)的结点数目均大于 10 000,因此,攻击者不会通过推断哪些结点是伪点并通过删除伪点及其连接边来进一步获得隐私信息.

Table 3 Number of added fake vertices

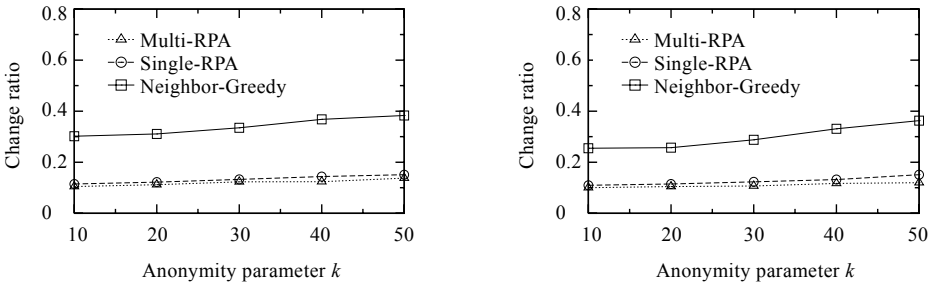
表 3 添加伪点数目

	Eu-Email					Epinions				
k	10	20	30	40	50	10	20	30	40	50
Neighbor-Greedy	0	0	18	48	58	8	20	30	26	42
Single-RPA	0	0	15	46	61	12	23	28	30	36
Multi-RPA	0	2	19	47	57	10	26	30	31	40

6.5 图结构信息损失

为了测试生成匿名图 G_k 在图结构方面的信息损失,本文测试了匿名图的结构变化率,包括聚集系数(clustering coefficient)和平均路径长度(average path length),来评估图结构数据可用性.

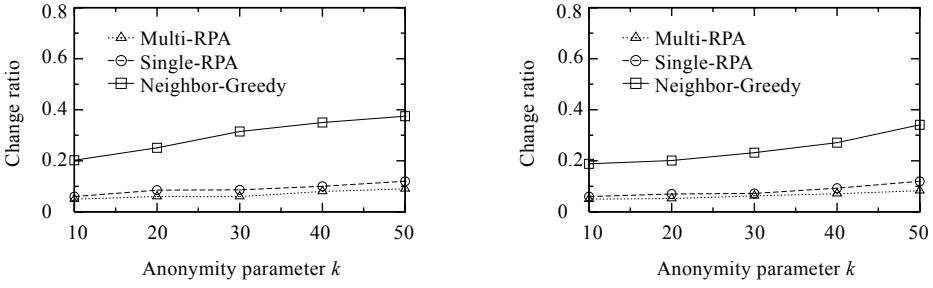
对于一个结点 v,v 的聚集系数是指其所有邻居结点对中具有边连接的比率;两个结点的路径长度是指该两点间的最短路径长度.定义图结构变化率 *Change ratio*=$|\tilde{P_o}P_a|/|P_o|$来衡量匿名图结构的变化,其中,P_o和P_a分别表示原图和匿名图中的图结构数值.在测试平均路径长度时,随机选择了 10 000 个结点对作为测试对象.图 10 和图 11 分别显示了聚集系数和平均路径长度的变化率.可以看出,RPA 算法生成匿名图的图结构变化率均小于 Neighbor-Greedy 算法.因为图结构与可达性相关,RPA 可以保持匿名图的可达性,因此也实现了图结构的保持.随着 k 值的增大,RPA 算法的图结构变化率基本不变,可见,保持结点间的可达性是保持图结构的基础.



(a) Eu-Email (b) Epinions

Fig.10 Change ratio of clustering coefficient

图 10 聚集系数变化率



(a) Eu-Email (b) Epinions

Fig.11 Change ratio of average path length

图 11 平均路径长度变化率

7 总结以及未来的工作

本文提出了在社会网络图匿名过程中保持结点间可达性问题.针对此问题,本文提出了可达性保持图匿名化算法(简称 RPA 算法).RPA 算法通过将结点进行分组并采取贪心策略进行匿名,从而减少匿名过程中的可达

性信息损失.同时,本文设计了一系列优化技术来保证 RPA 算法的执行效率,使其面对大规模社会网络图数据时具有可行性.本文首先提出采用可达区间来高效地评估边添加操作所导致的匿名损失;其次,通过采用候选邻居索引,进一步加速 RPA 算法对每个结点的匿名过程,保证了 RPA 算法的实用性.基于真实社会网络数据的实验结果表明了 RPA 算法的高执行效率,同时验证了生成匿名图在可达性查询方面的高精度.

距离查询是一种特殊的可达查询,其返回两个查询结点间的最短距离.距离查询是很多图应用的基础,保持匿名图在距离查询方面的可用性具有实际意义和应用价值.因此,如何在图匿名过程中保证距离查询的精度,成为我们下一阶段的研究方向.当前,我们正在研究图匿名操作对结点间最短距离的影响以及如何在图匿名过程中保持指定结点对间的最短距离.

References:

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

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

[3] Hay M, Miklau G, Jensen D, Towsley D. Resisting structural identification in anonymized social networks. In: Proc. of the 34th Int'l Conf. on Very Large Databases. 2008. 102−114. [doi: 10.14778/1453856.1453873]

[4] Zou L, Chen L, Ozsu MT. *K*-Automorphism: A general framework for privacy preserving network publication. In: Proc. of the 35th Int'l Conf. on Very Large Databases. 2009. 946−957. [doi: 10.14778/1687627.1687734]

[5] Cormode G, Srivastava D, Yu T, Zhang Q. Anonymizing bipartite graph data using safe groupings. In: Proc. of the 34th Int'l Conf. on Very Large Databases. 2008. 833−844. [doi: 10.14778/1453856.1453947]

[6] Bhagat S, Cormode G, Krishnamurthy B, Srivastava D. Class-Based graph anonymization for social network data. In: Proc. of the 35th Int'l Conf. on Very Large Databases. 2009. 766−777. [doi: 10.14778/1687627.1687714]

[7] Fard AM, Wang K, Yu PS. Limiting link disclosure in social network analysis through subgraph-wise perturbation. In: Proc. of the 15th Int'l Conf. on Extending Database Technology. 2012. 109−119. [doi: 10.1145/2247596.2247610]

[8] Gao J, Xu JY, Jin R, Zhou J, Wang T, Yang D. Neighborhood-Privacy protected shortest distance computing in cloud. In: Proc. of the 2011 ACM SIGMOD Int'l Conf. on Management of Data. 2011. 409−420. [doi: 10.1145/1989323.1989367]

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

[10] Cheng J, Fu AWC, Liu J. *K*-Isomorphism: Privacy preserving network publication against structural attacks. In: Proc. of the 2010 ACM SIGMOD Int'l Conf. on Management of Data. 2010. 459−470. [doi: 10.1145/1807167.1807218]

[11] Yuan M, Chen L, Yu PS. Personalized privacy protection in social networks. In: Proc. of the 36th Int'l Conf. on Very Large Databases. 2010. 141−150. [doi: 10.14778/1921071.1921080]

[12] Wang L, Meng XF. Location privacy preservation in big data era: A survey. Ruan Jian Xue Bao/Journal of Software, 2014,25(4): 693−712 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4551.htm [doi: 10.13328/j.cnki.jos.004551]

[13] Huo Z, Meng X, Zhang R. Feel free to check-in: Privacy alert against hidden location inference attacks in GeoSNs. In: Proc. of the 18th Int'l Conf. on Database Systems for Advanced Applications. 2013. 377−391. [doi: 10.1007/978-3-642-37487-6_29]

[14] Liu X, Yang X. Protecting sensitive relationships against inference attacks in social networks. In: Proc. of the 17th Int'l Conf. on Database Systems for Advanced Applications. 2012. 335−350. [doi: 10.1007/978-3-642-29038-1_25]

[15] Yuan M, Chen L, Yu PS, Yu T. Protecting sensitive labels in social network data anonymization. IEEE Trans. on Knowledge and Data Engineering, 2013,25(3):633−647. [doi: 10.1109/TKDE.2011.259]

[16] Wang Y, Xie L, Zheng B, Lee KCK. High utility *K*-anonymization for social network publishing. Knowledge and Information Systems, 2014,41(3):697−725. [doi: 10.1007/s10115-013-0674-2]

[17] Agrawal R, Borgida A, Jagadish HV. Efficient management of transitive relationships in large data and knowledge bases. In: Proc. of the 1989 ACM SIGMOD Int'l Conf. on Management of Data. 1989. 253−262. [doi: 10.1145/67544.66950]

[18] Anyanwu K, Sheth A. ρ-Queries: Enabling querying for semantic associations on the semantic Web. In: Proc. of the 12th Int'l Conf. on World Wide Web. 2003. 690−699. [doi: 10.1145/775152.775249]

[19] Wang H, He H, Yang J, Yu PS, Yu JX. Dual labeling: Answering graph reachability queries in constant time. In: Proc. of the 22nd Int'l Conf. on Data Engineering. 2006. 75−75. [doi: 10.1109/ICDE.2006.53]

[20] Chen Y, Chen Y. An efficient algorithm for answering graph reachability queries. In: Proc. of the 20th Int'l Conf. on Data Engineering. 2008. 893−902. [doi: 10.1109/ICDE.2008.4497498]

[21] Cheng J, Yu JX, Lin X, Wang H, Yu PS. Fast computing reachability labelings for large graphs with high compression rate. In: Proc. of the 11th Int'l Conf. on Extending Database Technology. 2008. 193−204. [doi: 10.1145/1353343.1353370]

[22] Seufert S, Anand A, Bedathur S, Weikum G. Ferrari: Flexible and efficient reachability range assignment for graph indexing. In: Proc. of the 2013 IEEE Int'l Conf. on Data Engineering. 2013. 1009−1020. [doi: 10.1109/ICDE.2013.6544893]

[23] Xie N, Shen D, Feng S, Kou Y, Nie T, Yu G. RIAIL: An index method for reachability query in large scale graphs. Ruan Jian Xue Bao/Journal of Software, 2014,25:213−224 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/14039.htm

[24] Cheng J, Shang Z, Cheng H, Wang H, Yu JX. *K*-Reach: Who is in your small world? In: Proc. of the 2012 VLDB Endowment. 2012. 1292−1303. [doi: 10.14778/2350229.2350247]

[25] Li M, Gao H, Zou Z. *K*-Reach query processing based on graph compression. Ruan Jian Xue Bao/Journal of Software, 2014,25(4): 797−812 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4567.htm [doi: 10.13328/j.cnki.jos.004567]

[26] Cohen E, Halperin E, Kaplan H, Zwick U. Reachability and distance queries via 2-hop labels. In: Proc. of the 13th Annual ACM-SIAM Symp. on Discrete Algorithms. 2002. 937−946.

[27] Bramandia R, Choi B, Ng WK. On incremental maintenance of 2-hop labeling of graphs. In: Proc. of the 17th Int'l Conf. on World Wide Web. 2008. 845−854. [doi: 10.1145/1367497.1367611]

[28] Wei F. TEDI: Efficient shortest path query answering on graphs. In: Proc. of the 2010 Int'l Conf. on Management of Data. 2010. 99−110. [doi: 10.1145/1807167.1807181]

[29] Paige R, Tarjan RE. Three partition refinement algorithms. SIAM Journal on Computing, 1987,16(6):973−989. [doi: 10.1137/0216062]

附中文参考文献:

[12] 王璐,孟小峰.位置大数据隐私保护研究综述.软件学报,2014,25(4):693−712. http://www.jos.org.cn/1000-9825/4551.htm [doi: 10.13328/j.cnki.jos.004551]

[23] 解宁,申德荣,冯朔,寇月,聂铁铮,于戈.RIAIL:大规模图上的可达性查询索引方法.软件学报,2014,25:213−224. http://www.jos.org.cn/1000-9825/14039.htm

[25] 李鸣鹏,高宏,邹兆年.基于图压缩的 k 可达查询处理.软件学报,2014,25(4):797−812. http://www.jos.org.cn/1000-9825/4567.htm [doi: 10.13328/j.cnki.jos.004567]

刘向宇(1981－),男,辽宁开原人,博士,讲师,主要研究领域为社会网络隐私保护.

周大海(1979－),男,讲师,主要研究领域为管理信息系统与数据库,工程数据库.

李佳佳(1987－),女,博士,讲师,主要研究领域为空间数据管理,移动对象数据管理,智能交通.

夏秀峰(1964－),男,博士,教授,CCF 高级会员,主要研究领域为工业大数据与企业私有云技术,数据挖掘与决策支持技术,NoSQL 数据库 技术.

安云哲(1978－),男,讲师,主要研究领域为管理信息系统与数据库,工程数据库,信息与系统集成.