计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2022,31(11):21−30 [doi: 10.15888/j.cnki.csa.008822]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于节点 1-邻居图相似性的社会网络匿名技术①

李啸林[1,2], 章红艳[1,2], 许佳钰[1,2], 许　力[1,2], 黄　赞[3]

[1](福建师范大学 计算机与网络空间安全学院, 福州 350007)
[2](福建师范大学 福建省网络安全与密码技术重点实验室, 福州 350007)
[3](福建省无线通讯重点实验室, 福州 350002)
通信作者: 许　力, E-mail: xuli@fjnu.edu.cn

摘　要: 利用传统的 *k* 匿名技术在社会网络中进行隐私保护时会存在聚类准则单一、图中数据信息利用不足等问题. 针对该问题, 提出了一种利用 Kullback-Leibler (*KL*) 散度衡量节点 1-邻居图相似性的匿名技术 (anonymization techniques for measuring the similarity of node 1-neighbor graph based on Kullback-Leibler divergence, SNKL). 根据节点 1-邻居图分布的相似性对原始图节点集进行划分, 按照划分好的类进行图修改, 使修改后的图满足 *k* 匿名, 完成图的匿名发布. 实验结果表明, SNKL 方法与 HIGA 方法相比在聚类系数上的改变量平均降低了 17.3%, 同时生成的匿名图与原始图重要性节点重合度保持在 95% 以上. 所提方法在有效保证隐私的基础上, 可以显著的降低对原始图结构信息的改变.

关键词: 隐私保护; 社会网络; 概率不可区分性; *k* 匿名; 1-邻居图; 网络安全

引用格式: 李啸林,章红艳,许佳钰,许力,黄赞.基于节点 1-邻居图相似性的社会网络匿名技术.计算机系统应用,2022,31(11):21–30. http://www.c-s-a.org.cn/1003-3254/8822.html

Social Network Data Anonymization Based on Node 1-neighbor Graphs Similarity

LI Xiao-Lin[1,2], ZHANG Hong-Yan[1,2], XU Jia-Yu[1,2], XU Li[1,2], HUANG Zan[3]

[1](School of Computer and Cyber Security, Fujian Normal University, Fuzhou 350007, China)
[2](Fujian Provincial Key Lab of Network Security & Cryptology, Fujian Normal University, Fuzhou 350007, China)
[3](Key Laboratory of Wireless Communication in Fujian Province, Fuzhou 350002, China)

Abstract: Using traditional *k*-anonymization techniques to achieve privacy protection in social networks is faced with problems such as single clustering criterion and under-utilization of data and information in the graph. To solve this problem, this study proposes an anonymization technique measuring the similarity of the node 1-neighbor graph based on the Kullback-Leibler divergence (SNKL). The original graph node set is divided according to the similarity of node 1-neighbor graph distribution, and the graph is modified according to the divided classes so that the modified graph satisfies *k*-anonymity. On this basis, the anonymous release of the graph is implemented. The experimental results show that compared with the HIGA method, the SNKL method reduces the amount of change in the clustering coefficients by 17.3% on average. Moreover, the overlap ratio between the importance nodes of the generated anonymous graph and those of the original graph is maintained at more than 95%. In addition to protecting privacy effectively, the proposed method can significantly reduce the changes brought to the structural information in the original graph.

Key words: privacy protection; social network; probabilistic indistinguishability; *k*-anonymity; 1-neighbor graph; cyber security

① 基金项目: 国家自然科学基金 (U1905211, 61771140, 62171132); 福建省科技项目 (2021L3032); 企事业合作项目 (DH-1565)
收稿时间: 2022-03-09; 修改时间: 2022-04-07; 采用时间: 2022-04-25; csa 在线出版时间: 2022-07-25

社会网络 (social networks, SNs) 是一种人与人之间的关系与互动的结合[1]. 社会网络把网络中的每个节点看作参与这个网络中人的抽象, 每个人之间的关系则抽象成节点之间的连边, 网络中每个人的行为不同且具有不同的属性特征. 随着网络与计算机技术的快速发展, 人们的日常生活与互联网正在不断的相互融合, 人们通过微信、QQ、新浪微博等各种软件以及平台来发布自己的兴趣爱好、位置等信息. 随着用户发布的信息越来越多, 其网络形象就被进一步的丰富, 服务提供商可以对这些数据进行分析与筛选, 获得有用的数据来为用户带来更好的服务体验. 而攻击者可以通过数据分析获得用户的身份信息以及关系信息等隐私数据, 造成用户隐私的泄露, 对用户的个人信息及财产安全带来严重的威胁的同时也严重地影响了数据发布者的信誉. 在社交网络[2]、医疗大数据[3] 与金融机构[4] 中都存在着隐私泄露的风险, 因此隐私保护是一个不容忽视的问题. 民法典中明确规定了数据的流动和处理应建立在个人信息保护的基础之上. 因此, 如何在保证发布数据具有可用性的同时保护用户数据的隐私, 成为服务提供商以及消费者共同关注的一个问题.

图数据集中可能包含敏感、可辨别个人身份信息. 因此, 必须对公开发布的数据采取一定的隐私保护方法, 防止攻击者通过背景知识或链接攻击等手段获取到用户的隐私信息[5]. 在数据发布时, 通过去掉用户的身份标识来对原始图进行处理并不能很好的保护用户的隐私, 对手可以根据背景知识, 从发布图中识别出用户的身份[2,6]. 为了缓解社会网络中的隐私泄露问题, 产生了基于匿名化的技术[7]、加密技术[8,9] 和差分隐私技术[10,11] 等诸多隐私保护技术. 例如近期 Ding 等人[12] 提出了一种新的 k-分解方法对大型图数据集进行隐私保护, 在隐私、数据可用性和效率之间取得了较好的平衡. 许佳钰等人[13] 提出了一种基于节点平均度的 k-度匿名隐私保护方案, 利用基于平均度的贪心算法对社会网络节点进行划分, 然后利用优先保留重要边的图结构修改方法对图进行修改, 从而实现图的 k-度匿名化. 黄海平等人[14] 提出了一种基于边介数模型的差分隐私保护方案, 根据介数排序的 dk 序列将重要性相近的边归为一类进行分组加噪, 保留了更多结构特征的同时也具有较好的数据可用性. Huang 等人[15] 结合聚类和随机化算法提出了一种隐私保护算法 PBCN, 在数据可用性和隐私保护水平之间实现折中. Zhu 等人[16] 将网络物理系统中的数据发布问题转移到机器学习问题中, 减少了查询集之间的相关性, 同时对新的输入查询进行预测.

k-匿名与差分隐私是两种常用的轻量级匿名术. 在 k-匿名中, 节点被重新识别的概率最多为 $1/k$, 数据发布者可以通过调整 k 值的大小来调整匿名数据集的信息损失以及所能达到匿名程度. 数据所有者可以利用 k-匿名技术修改图结构, 然后发布匿名后的图用于图数据挖掘以及分析[17–20]. 使用差分隐私对图数据进行扰动, 可以最大限度地提高对统计值的查询精度, 同时, 通过添加特定的噪音, 可以最大限度地降低识别出个体的概率[21–23]. 与 k 匿名技术不同, 差分隐私技术具有数学证明来保证其安全性. 然而, 当在图数据上使用差分隐私时, 因为差分隐私只支持有限的统计值, 这就限制了其在高级数据挖掘中的操作. 相比之下, k 匿名的安全性很容易证明, 通过要求共享数据中存在一定数量 ($\geqslant k$) 在某项统计值上不可区分的记录, 使攻击者最多只能以 $1/k$ 的概率通过个体的某项统计特征关联出个体身份, 从而保护了个体的隐私[24]. 在 k 较大的情况下个体被重新识别的概率很小, 而且还可以支持结构化查询, 适用于实践中复杂的图数据挖掘和分析任务[12]. 因此, 在本文中使用 k 匿名技术来使 k 个节点的 1-邻居图同构, 当 k 值较大时具有较为直观的安全性, 同时对于 1-邻居图这种较为复杂的子图结构, 差分隐私在设计时需要考虑较为复杂的敏感度定义, 而 k 匿名所对应的统计特征可直接定义为节点的 1-邻居图特征, 为方案的可实施性带来了优势, 有利于方案在实际场景中广泛的应用.

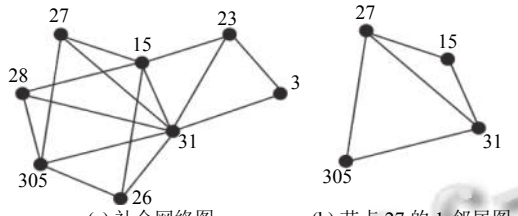
最近的研究表明, 如果攻击者拥有目标节点的度、邻居或子图等背景知识信息, 则可以通过实施 1-邻居攻击以较大的概率在匿名图中重新识别出目标节点. Zhou 等人[25] 利用 k-邻居匿名技术, 使得每个节点的 1-邻居图与其他 k–1 个节点的 1-邻居图同构, 能够较好地抵御 1-邻居攻击. Liu 等人[26] 为了抵抗 1-邻居攻击, 同时保持发布图的高可用性, 提出了一种启发式不可区分群体匿名化方案, 在真实和合成数据集上取得了较好的效果.

但是, 在多数 k-邻居匿名方案中, 未充分利用待修改图中节点 1-邻居图之中的相似性. 导致匿名后的图可用性较低, 或者在针对度分布比较特殊的图进行修改时, 鲁棒性较差. 针对以上问题, 本文提出了一种基

于 *KL* 散度的社会网络数据匿名案. 与传统方案相比, 本方案通过在粗划分过程中考虑同类节点之间的度差异, 初步将同一类中大多数节点匿名前后的度改变量控制在一个较小的范围内; 使用类的平均度与平均聚类系数来定义类距离, 可以反映出类中节点的平均度以及 1-邻居图的聚集程度, 在合并过程中考虑待合并类与其上下类的类距离, 使得同一类中的节点具有相似的度以及 1-邻居结构; 在类拆分时使用节点的 1-邻居特征分布来代表节点的 1-邻居结构特征, 使用概率分布来对节点的 1-邻居特征进行建模, 通过 *KL* 散度反映类中任意两节点的 1-邻居特征分布相互拟合所产生的信息损耗, 将 1-邻居分布最相似的节点拆分为同一类, 来充分利用节点 1-邻居结构的相似性, 使得本方案修改后的图可用性较高.

1 相关定义

定义 1 (1-邻居图). 给定图的节点集$S \in V(G)$, G 在 S 上的导出图为$G(S)=(S,E_S)$, 其中, $E_S=\{(u,v)|(u,v)\in E(G)\wedge u,v\in S\}$. 节点 u 的 1-邻居图定义为图 G 在节点 u 邻居集合上的导出子图$Neighbor_G(u)=G(N_u+u)$, 其中, $N_u=\{v|(u,v)\in E(G)\}$. 如图 1 所示, 图 1(b) 为图 1(a) 中节点 27 的 1-邻居图.



(a) 社会网络图 (b) 节点 27 的 1-邻居图

图 1 1-邻居示意图

定义 2 (1-邻居攻击). 给定图$G=(V,E)$, V 和 E 分别代表图中的节点和边. 攻击者拥有目标节点 u 的 1-邻居图背景知识$Neighbor_G(u)$, 在匿名图 G^*中, 攻击者可以利用$Neighbor_G(u)$重新识别出节点 u 在匿名图 G^*中的 1-邻居图$Neighbor_{G^*}(u)$.

定义 3 (聚类系数). 给定图$G=(V,E)$, 节点 u 的度值为 $k(u)$, $E(u)$ 为节点 u 的 $k(u)$ 个邻居之间实际存在的边数, 则聚类系数 C_u 如式 (1) 所示:

$$C_u=\frac{2E(u)}{k(u)(k(u)-1)} \tag{1}$$

2 基于 *KL* 散度的社会网络数据匿名

本文从提高数据可用性, 降低信息损失量等方面出发, 提出了一种基于 *KL* 散度的社会网络数据匿名方案. 该方案主要分为 4 步: (1) 粗划分. 将度差值小于Δ的节点划分到同一类中. 其中, Δ为一个预先设定好的值且Δ为一个正整数. (2) 类合并. 对于节点数量小于 k 的待合并类, 计算其平均度与平均聚类系数的加权平均值, 与上下类的平均度和平均聚类系数进行比较, 得出待合并类与上下类的距离, 将待合并类中的所有节点加入距离最小的类中, 使得所有类的节点数量大于 k. (3) 类拆分. 对于同一类中的节点, 使用节点 1-邻居图的 3 种分布来表示其特征, 使用 *KL* 散度计算节点之间的相似度矩阵, 使用相似度矩阵对数量超过 $2k$–1 的类进行拆分, 使得所有类节点的数量大于 k 小于 $2k$–1. (4) 图修改. 对拆分好的类进行图修改, 实现 k-邻居匿名. 与传统方案相比, 本方案在划分类时考虑了更多的结构信息, 使得同类的节点具有较高的结构相似度, 减少了图修改过程中的信息损失, 生成匿名图的可用性较高.

2.1 粗划分

本方案通过 k 匿名来进行隐私保护, 首先对原始节点集进行粗划分, 将节点按度值从大到小排序, 从度最大的节点开始, 将度差值小于Δ的节点划分到同一类中. 通过调整Δ的大小, 可以控制同类节点之间度的差异程度. Δ的取值主要取决于两个方面: 一是网络度分布的方差, 当度分布的方差比较大时, 说明节点之间度的差异性较大, 一般使用较大的Δ; 二是取决于 k 值的大小, 当 k 值较大时, Δ的值也应相应的增大.

2.2 类合并

定义 4 (类距离 $Dist(C_1, C_2, W_1, W_2)$). 类与类之间的度与聚类系数加权距离. 为了衡量两个类之间的相似程度, 使得待合并的小数量类能够加入最相近的类中, 本方案使用类与类之间平均聚类系数与平均度加权求和之后的差来衡量类之间的距离, 其值可用式 (2) 计算出:

$$Dist=W_1\times\frac{|AVGD(C_1)-AVGD(C_2)|}{AVGD(C_1)}+W_2\times\frac{|AVGC(C_1)-AVGC(C_2)|}{AVGC(C_1)} \tag{2}$$

其中, $AVGD(C)$为类 C 的平均度, $AVGC(C)$为类 C 的平均聚类系数, W_1 为平均度在计算距离时所占的权重, W_2 为平均聚类系数在计算距离时所占的权重, 且

$W_1 + W_2 = 1$.

定义 5 (类指针$Before(M,C)$,$After(M,C)$). 当前类的上一类与下一类. 在类合并过程中, 当前的类可能会被合并, 为了使之后的类在进行类合并时能够快速找到其前一类与后一类, 本文使用 *Before* 与 *After* 指针分别指向当前类的上一类与下一类, 假设当前类为 B, 其前一类与后一类分别为 A 和 C. 在当前类进行合并之后, *Before* 指针不做修改, 仍然指向 A, 如果当前类没有进行合并, 则将 *Before* 指向 B, 最后将当前类设置为 C. 在本方案中, 指针 *After* 不需要做特殊处理, 直接指向当前类的下一类即可.

完成粗划分后, 同一类中节点之间的度差值小于一个预设值Δ. 为了使处理后的图满足 k 匿名, 需要使每一个类中的节点数量满足$k \leqslant N(C) \leqslant 2k-1$, $N(C)$ 为类 C 中节点的数量. 对于粗划分后的类集合 M, $\exists C \in M$, 有 $len(C) < k$, 即类 C 中的元素个数小于 k. 因此, 需要对类集合 M 中的小数量类 C 进行合并处理, 使得类集合 M 中所有类的元素数量大于 k. 以下是类合并的流程: (1) 对于集合 M 中的每一个类, 使用定义 5 中的类指针获得其所对应的上下类. (2) 若当前类为节点数量小于 k 的待合并类, 使用式 (2) 计算出待合并类与上下类的距离 *Dist*1、*Dist*2. (3) 判断 *Dist*1、*Dist*2 的大小, 将待合并类加入到距离最小的类中. (4) 若当前类节点数量大于 k, 则无需进行合并操作. (5) 返回合并后的类集合 P. 类合并的算法如算法 1.

算法 1. 类合并

输入: 待修改图 G, k 匿名值 k, 度差值在合并时所占权重 W_1, 聚类系数在合并时所占权重 W_2, 粗划分得到的类集合 M
输出: 若干个节点数量大于 k 的集合 P

```
1)  for each class in M
2)      BC←Before(M, C);
3)      AC←After(M, C);
//获得当前类 C 的上一类 BC 和下一类 AC
4)  if len(C) < k
5)      TOP←Dist(C, BC, W1, W2);
6)      SUB←Dist(C, AC, W1, W2);
//获得当前类 C 与上下类 BC 和 AC 的度与聚类系数的加权距离
7)  if TOP < SUB
8)      Add(BC, C);
9)  else
10)     Add(AC, C);
11) end for
12) return P
```

2.3 类拆分

定义 6 ($KL(L_i, L_j)$). 同类节点 i 与 j 之间的相对熵. 为了衡量同类节点之间的相似程度, 在类拆分时将最相似的放在一起, 使用式 (3) 进行相对熵的计算:

$$KL(L_i, L_j) = \frac{1}{2}\left(\sum_{t=1}^{n} P_{it} \times \log\frac{P_{it}}{P_{jt}} + \sum_{t=1}^{n} P_{jt} \times \log\frac{P_{jt}}{P_{it}}\right) \quad (3)$$

其中, L_i 和 L_j 是同一类中两个节点的 1-邻居度特征分布, n 为 L_i 和 L_j 中节点个数的最大值, P_{it}为节点 i 的 1-邻居图中第 t 个节点的特征的值除以节点 i 的 1-邻居图中所有节点特征值之和所得的数值.

完成类合并后, 类集合 P 中每一个类中节点的数量都大于 k. 其中节点个数在 k 到 $2k$–1 之间的类, 由于本文使用类距离作为合并标准, 使用平均度保证了在同一类中的节点邻居数量相似, 同时, 采用平均聚类系数保证了同一类中节点的 1-邻居图具有相似的聚集程度, 因此, 在同一类中的节点具有相似的 1-邻居结构, 满足 k 匿名需求. 为了使处理后的图满足 k 匿名, 使得每个类中的节点数量满足$k \leqslant N(C) \leqslant 2k-1$, $N(C)$ 为类 C 中节点的数量. 对于类合并后的集合 P, $\exists C \in P$, $N(C) > 2k-1$. 为此, 对类集合 P 中的大数量类 C 进行拆分, 使得 P 中所有类的元素数量大于 k 小于等于 $2k-1$.

下面给出类拆分的具体流程: (1) 寻找节点数量大于 $2k$–1 的类. (2) 使用节点 1-邻居图的 3 种度分布 (节点在整个网络中的度、节点在 1-邻居图中的度、整个网络中的度与 1-邻居图中的度的差值) 的组合来表示节点的 1-邻居度分布特征. (3) 对于同一类中的节点, 使用 *KL* 散度计算节点 1-邻居分布之间差异程度, 得到差异化矩阵 S, 使用 1–S 得到同类节点之间的相似度矩阵 M. 矩阵中 i 行 j 列的值 $M[i][j]$ 代表了同类节点 i 与 j 之间的相似程度. (4) 在相似度矩阵中寻找最大值. 最大值在矩阵中的行列值即为当前类中最相似的节点对. 将最相近的节点对放入新的类 $New[num]$ 中. 将矩阵中的最大值对应的行与列赋值为最小值, 重复步骤 (4), 直到新类 $New[num]$ 中节点数量大于 k, 或相似度矩阵中最大值与最小值相等. (5) 若新类 $New[num]$ 中节点的数量大于 k, 令 num 加一, 创建新的类 $New[num]$, 重复步骤 (4). (6) 若相似矩阵中的最大值与最小值相等, 则说明拆分已经完成. 类拆分和相似度矩阵生成的具体算法如算法 2.

算法 2. 类拆分

输入: 合并完后类的集合$P=\{P_1,P_2,\cdots,P_n\}$
输出: 划分好的类的集合$C=\{C_1,C_2,\cdots,C_m\}$

```
for each Class in P
num = 0;
    if len(Class) > 2k – 1
        L←Generate(Class);
//生成同一类节点的 1-邻居图度分布列表 L
        Matrix←Sim(L, len(Class));
//使用算法 3 生成相似度矩阵
        Min←min(Matrix);
        Max←max(Matrix);
        while(Min!=Max)
            (i, j)←Index(Matrix, Max);
//找出最大值所对应的行列值
            if len(C[num]) < k
                Add(C[num], Class[i], Class[j]);
                Setcr(Matrix, Min);
//将矩阵第 i 行和第 j 列设为最小值
            else
                num = num + 1;
//在类集合中选择下一个空的新类
                Add(C[num], Class[i], Class[j]);
                Setcr(Matrix, Min);
end for
return C
```

算法 3. 相似度矩阵生成

输入: 同一类节点的 1-邻居图特征分布列表 L, 类中的节点个数 $length$
输出: 相似度矩阵 $Matrix$

```
for i in range(length)
    for j in range(i+1, length)
        Matrix[i][j]←KL(L_i, L_j);
End for
Min←min(Matrix);
Matrix←Setlowertriangle(Min);
//将矩阵下三角全部设置为最小值
Matrix←(1–Matrix);
//使用一个上三角除对角线部分元素全为 1 的矩阵减去矩阵 Matrix, 得到相似度矩阵 Matrix
return Matrix
```

如算法 2 所示, 类拆分的过程包括 4 个部分: 1) 相似度矩阵生成; 2) 相似度矩阵最大值与最小值的查找; 3) 寻找最大值所对应的节点对, 将其加入长度小于 k 的新类中; 4) 将矩阵中最大值所对应的行与列设置为最小值. 下面详细介绍这 4 个部分的处理过程.

步骤 1. 相似度矩阵生成

对于合并完的类集合 P, 有$\forall x \in C_i$, 其中, $C_i \in \{C|len(C) > 2k-1, C \in P\}$, 令$G_N$为节点 x 的 1-邻居图, $V(G_N)$为 1-邻居图的节点集. 节点 1-邻居图的 3 种分布为:

$$\begin{cases} L_1 = \{d|d = d_G(y), y \in V(G_N)\} \\ L_2 = \{d|d = d_{G_N}(y), y \in V(G_N)\} \\ L_3 = \{d|d = d_G(y) - d_{G_N}(y), y \in V(G_N)\} \end{cases}$$

其中, $d_G(y)$为节点 y 在图 G 中的度, $d_{G_N}(y)$为节点 y 在图G_N中的度. 对于$\forall x_1, x_2 \in C_i$, 得到其 1-邻居的 3 种度分布后, 使用式 (3) 计算 x_1, x_2 的 1-邻居图度分布之间的 3 种 KL 散度: $KL_1(L_{x_1}, L_{x_2})$, $KL_2(L_{x_1}, L_{x_2})$, $KL_3(L_{x_1}, L_{x_2})$. 然后使用式 (4) 计算相似度矩阵:

$$sim = \begin{pmatrix} 0 & 1 & \cdots & 1 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \cdots & \cdots & 0 \end{pmatrix} - I[i] \begin{pmatrix} 0 & KL_i(L_0, L_1) & \cdots & KL_i(L_0, L_n) \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & KL_i(L_{n-1}, L_n) \\ 0 & \cdots & \cdots & 0 \end{pmatrix} \quad (4)$$

其中, $i \in [1,3]$且$\sum_{i=1}^{3} I[i] = 1$.

步骤 2. 矩阵最大值与最小值的查找

在本步操作中, 由相似矩阵的计算过程可知, 相似矩阵的最大值对应的行列值, 是同类中最相似的节点对的坐标.

步骤 3. 寻找最大值所对应的节点对, 将其加入长度小于 k 的新类中

在本步操作中, 获得最大值所对应的节点对, 需要将其加入同一个类中, 这时在新类的构造时, 为了使匿名后的图满足 k 匿名, 需要类中的节点数量在 k 到 $2k$–1 的范围之内, 所以当上一个类中的节点数量大于 k 时, 需要创建新的类来加入当前的节点对.

步骤 4. 将矩阵中最大值所对应的行与列设置为最小值

上面的操作已经将矩阵最大值所对应的节点对加入新的类中, 所以在下面进行循环的过程中, 最大值所对应的行与列已经不存在于待拆分的类中. 如图 2 所示, 将相似度矩阵最大值所对应的行与列设置为矩阵的最小值, 在后面的循环中就不再考虑这些元素, 当矩阵所有的元素都为最小值时, 说明当前类已经拆分完成, 退出循环.

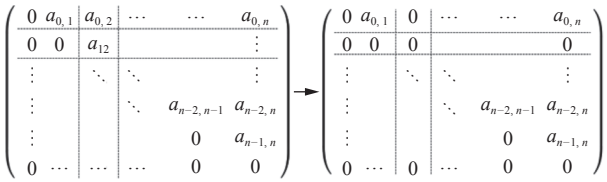


图 2 相似矩阵变化

2.4 图修改

完成类拆分后, 每一个类中都有 (k, $2k$–1) 个节点, 要实现 k 匿名, 同时实现同一类中节点的 1-邻居图之间具有概率不可区分性, 需要对在同一类中的每个节点的 1-邻居图进行图修改, 使得同类中每个节点的 1-邻居图同构, 从而抵御 1-邻居攻击. 以下是图修改的流程: (1) 对于类集合中的每一个待修改类, 寻找类中度最大的节点 *Max*. (2) 对于当前待修改类中的每一个节点, 调用最优图编辑距离算法, 获得当前节点所对应的 1-邻居图与 *Max* 节点所对应的 1-邻居图同构的节点与边修改序列. (3) 根据修改序列, 将当前节点所对应的 1-邻居图中的节点与边进行增删操作, 完成子图重构. (4) 当类集合中所有类的节点所对应的 1-邻居图完成重构操作后, 返回修改后的图 G^*. 图修改的算法如算法 4.

算法 4. 图修改

输入: 待修改的图 G, 划分好的类的集合$C=\{C_1,C_2,\cdots,C_m\}$
输出: 修改后的图 G^*

```
1)  for each Class in C
2)      Max←max(Class);
//寻找同一类中度最大的节点
3)      for each node in Class
4)          Edit_list←OEP(G, node, Max);
//使用 optimize_edit_path 算法计算使节点 node 的 1-邻居图与节点 Max 的 1-邻居图同构的修改序列
5)          G_mo(G, Edit_list, node);
//根据修改序列对节点 node 进行图修改
6)      end for
7)  end for
8)  return G*
```

在本步操作中, 使用每个节点所对应的修改序列, 对原图中节点的 1-邻居图进行相应的修改, 使得节点所在的 1-邻居图与最大度节点所在的 1-邻居图同构. 如图 3 所示为节点 232 修改前后的拓扑图, 修改后节点 232 的 1-邻居图与 *Max* 节点 1-邻居图同构.

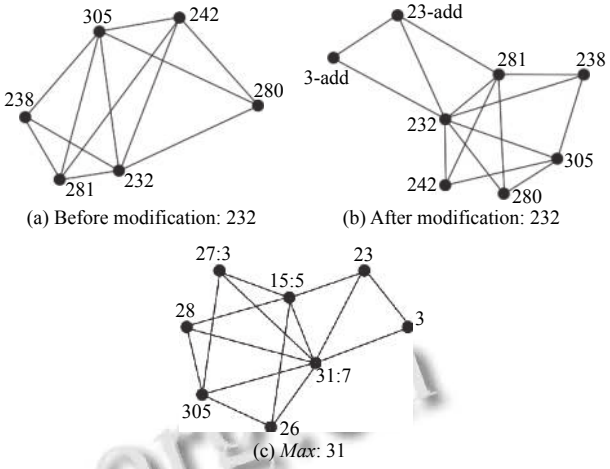


图 3 1-邻居图修改

3 实验分析

实验采用真实数据集 email-Enron、ca-CondMat、ca-HepTh 和 ego-Facebook. 这 4 个真实数据集来自于斯坦福大学大型网络数据集 (http://snap.stanford.edu/data/). email-Enron 包含 36 692 个节点和 183 831 条边; ca-CondMat 包含 23 133 个节点和 93 497 条边; ca-HepTh 包含 9877 个节点和 25 998 条边; ego-Facebook 包含 4039 个节点和 88 234 条边. 算法代码用 Python 编程实现, 实验环境为 Intel Xeon E5-2650 CPU 2.20 GHz, 755.6 GB 内存, 操作系统为 Ubuntu 20.04.1 LTS.

3.1 隐私保护程度分析

本节分析了可能的隐私攻击手段以及本方案对此类攻击的抵挡能力.

(1) 攻击者已知目标节点的邻居节点数.

在子图同构的过程中, 本方案会使同一类中每个节点的 1-邻居图与度最大节点的 1-邻居图同构, 所以在每一类处理完成后, 同一类中每个节点的度值与当前类中度最大节点的度值相同, 因此也满足 k 度匿名, 使得攻击者从匿名图 G^*中重新识别出目标节点的概率不超过 1/k.

(2) 攻击者拥有原图中目标节点的 1-邻居图结构信息.

攻击者使用原始图中节点 t 的 1-邻居图结构 $G(t)$(攻击者的背景知识), 尝试从发布的匿名图 G^*中重新识别出目标图 $G^*(t)$, 可以知道 $G^*(t)$ 一定属于一个概率不可区分的组合 g, 其中$\|g\| \geqslant k$. 同时, 对于原图中的每一个节点 v, 攻击者仅通过比较 $G^*(v)$ 与 $G^*(t)$ 无

法推测出在原图中 $G(v)$ 是否与 $G(t)$ 相同.

由此可知, 本方案对于度攻击以及 1-邻居攻击具有较好的抵抗性. 能够在攻击者拥有目标节点度以及 1-邻居结构的背景知识之下, 保护目标节点的隐私.

3.2 数据可用性分析

本文通过衡量平均度 (AVE) 改变量、平均聚类系数 (ACC) 改变量、平均最短路径 (APL) 改变量来衡量匿名前后图结构数据的变化程度, 其变化越小, 说明数据可用性越好. 如表 1 所示, 本文分析了随着 k 值 (每一个概率不可区分的组合中节点数量的大小) 从 5 变化到 25 (隐私要求越来越高), 本方案在不同数据集上关于结构可用性保持情况方面的表现. 第 1 行为图的名称; 第 2 行分别为: (1) 平均度 (AVE); (2) 平均聚类系数 (ACC); (3) 平均最短路径 (APL). 下一行为原始图中的各项指标大小, 下面各行为 k 取不同值的情况下, 匿名图各项指标的大小. 此表分析了匿名程度不同情况下, 匿名前后图可用性的变化. 可以看出, 随着 k 的增加, 匿名图的可用性会呈降低趋势, 本方案通过在类划分过程中对同类节点之间的相似度进行控制, 使得修改前后图的结构可用性得到了较好的保证.

表 1 匿名图的可用性测试

k	Facebook networks			Enron networks			ca-CondMat networks			ca-HepTh networks		
	AVE	ACC	APL	AVE	ACC	APL	AVE	ACC	APL	AVE	ACC	APL
Original	44.0	0.605	4.70	10.0	0.497	4.90	8.0	0.633	6.40	5.7	0.481	5.90
5	42.7	0.581	4.04	10.7	0.474	4.69	8.6	0.596	6.21	6.2	0.459	5.65
10	42.4	0.571	3.83	11.0	0.469	4.66	8.8	0.587	6.08	6.4	0.454	5.62
15	44.6	0.565	3.78	11.4	0.474	4.61	8.9	0.585	6.02	6.4	0.452	5.59
20	46.8	0.575	3.76	11.9	0.477	4.52	8.9	0.581	6.06	6.6	0.451	5.36
25	49.7	0.593	3.80	12.0	0.480	4.48	9.1	0.581	5.99	6.6	0.450	5.48

图 4–图 7 表示在修改后的图中度值排在前 1%、5%、10% 的节点集与原图中度值在前 1%、5%、10% 的节点集的重合度, 可以看出随着 k 值的增加, 重合度不会发生很大的变化, 大多保持在 95% 以上. 因此可知, 使用本方案进行匿名能够以很大的概率保留在原图中比较重要的节点.

图 8 和图 9 分析了本方案与 Liu 等人[26] 的 HIGA 匿名方案关于匿名后图可用性的对比. 可以发现, 两种方案随着 k 值的增加, 对原始图各项指标改变的百分比都是呈现出上升趋势, 在 Facebook 数据集上, HIGA 模型生成的匿名图可用性随着 k 的增大迅速下降, 不能够保证匿名图的可用性, 这是因为 Facebook 数据集的度分布十分不平衡, 存在着少量且度数相差较大的大度节点, 同时存在着大量的十分密集的小度节点, 这就使得 HIGA 模型在对这种度分布差异较大的图进行修改时, 需要大量的修改才能保证匿名程度. 本方案在对节点进行分类时, 考虑了 1-邻居节点的分布特征, 保证了在同一类中的节点拥有相同的结构特征, 同时在进行修改时, 将同一节点的 1-邻居图修改成同构, 在一定程度上保留了节点本身的结构特征, 因此对原图的结构破坏较小, 保留了较高的可用性, 对于类似 Facebook 这种度分布十分不均匀的图, 也具有很好的鲁棒性. 对于平均聚类系数指标, 与 HIGA 方法相比, SNKL 方法在聚类系数上的改变量平均降低了 17.3%, 对于平均度与平均最短路径指标, SNKL 方法与 HIGA 方法相差较小, 因此使用 SNKL 方法进行数据匿名, 可以在保证隐私的前提下, 使得匿名后图的结构特征得到更好的保留.

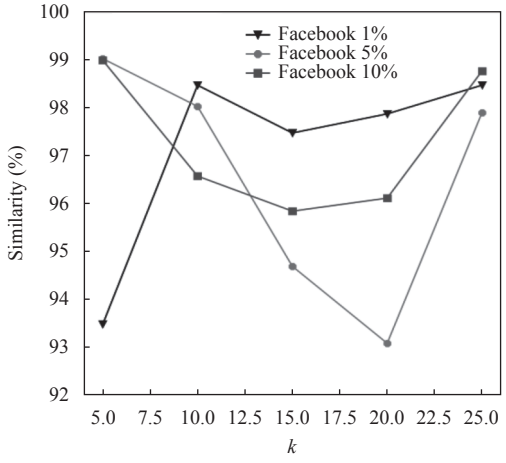


图 4 Facebook 保持百分比

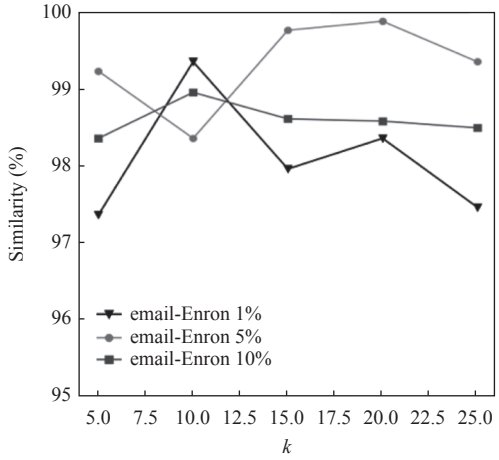


图 5 email-Enron 保持百分比

处理, 在实现匿名的同时减少修改量, 进一步提高匿名图的可用性.

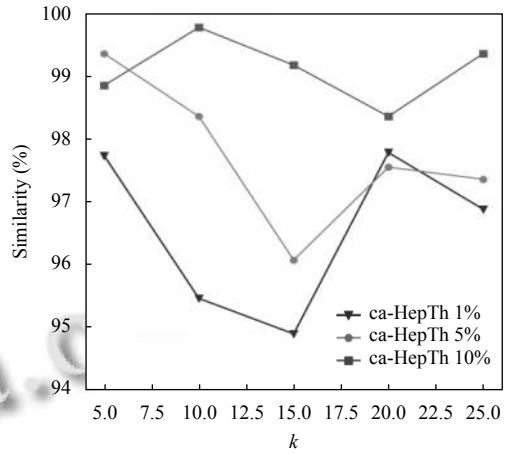


图 6 ca-HepTh 保持百分比

4 结语

本文研究了社会网络中用户的匿名问题, 并提出了一种基于节点 1-邻居图相似性的社会网络数据匿名方案. 该方案通过对 1-邻居结构相似的节点集进行修改, 实现同类中的节点之间近似同构, 使得拥有在原图中目标节点 1-邻居结构这种背景知识的攻击者在匿名后的图中成功识别出目标节点的概率不会超过 1/k, 从而实现群体化匿名. 同时该算法在进行类合并和类划分时, 考虑了同类节点间的结构特征, 使得修改后的图也会保留主要的特征结构, 减少了信息损失. 实验结果表明, 该算法在实现群体化匿名的同时, 还保留了较高的可用性. 下一步可以考虑对类似于 Facebook 这种度分布十分不均匀的图, 分析其独特的分布趋势, 将数量较少的大度节点单独

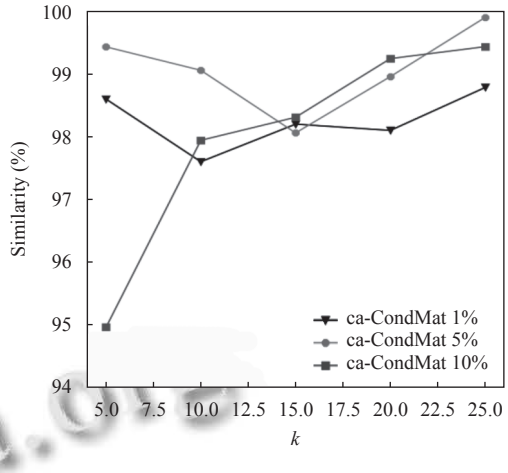


图 7 ca-CondMat 保持百分比

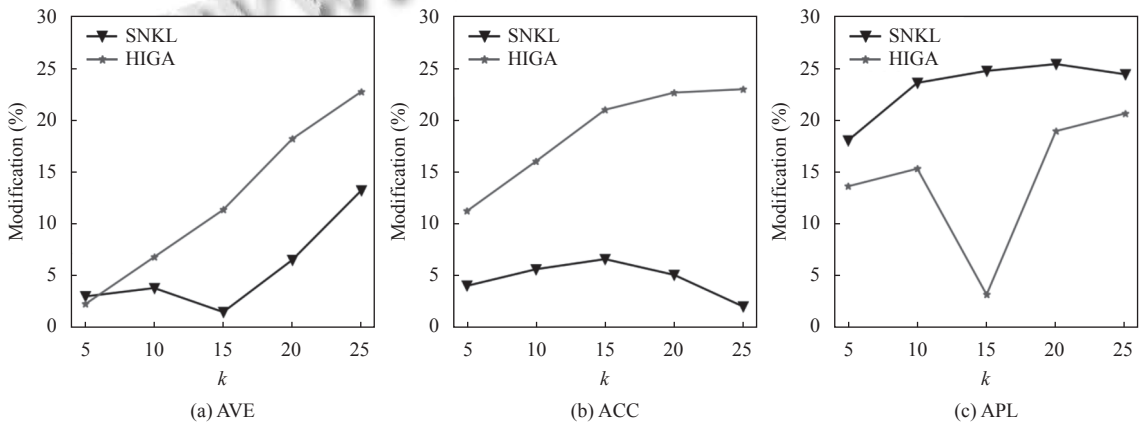


图 8 Facebook 可用性对比

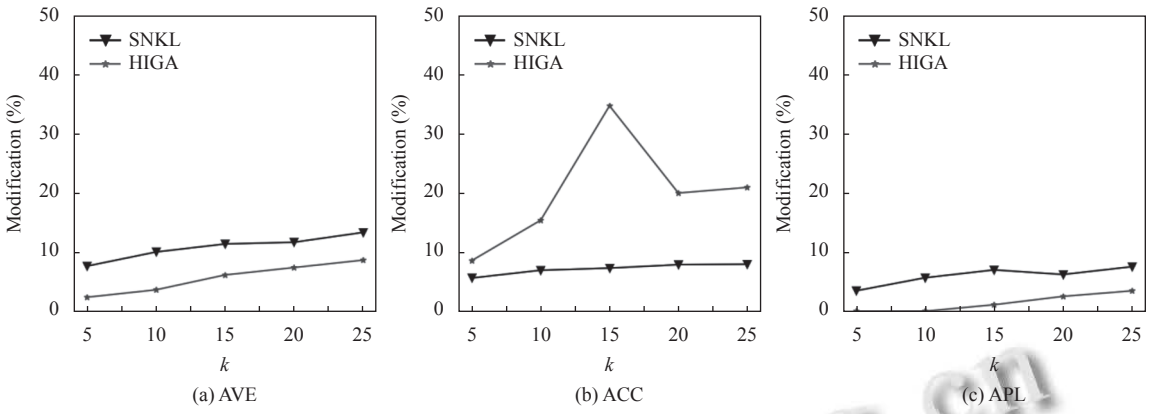


图 9 ca-CondMat 可用性对比

参考文献

1 Banerjee S, Jenamani M, Pratihar DK. A survey on influence maximization in a social network. Knowledge and Information Systems, 2020, 62(9): 3417–3455. [doi: 10.1007/s10115-020-01461-4]

2 Ji SL, Li WQ, Srivatsa M, *et al*. General graph data de-anonymization: From mobility traces to social networks. ACM Transactions on Information and System Security, 2016, 18(4): 12.

3 Liu Y, Ma Z, Liu XM, *et al*. Privacy-preserving object detection for medical images with faster R-CNN. IEEE Transactions on Information Forensics and Security, 2022, 17: 69–84. [doi: 10.1109/TIFS.2019.2946476]

4 Chen C, Wu BZ, Wang L, *et al*. Nebula: A scalable privacy-preserving machine learning system in ant financial. Proceedings of the 29th ACM International Conference on Information & Knowledge Management. ACM, 2020. 3369–3372.

5 Ye AY, Jin JL, Yang ZJ, *et al*. Evolutionary game analysis on competition strategy choice of application providers. Concurrency and Computation Practice and Experience, 2021, 33(8): e5446.

6 Ozalp I, Gursoy ME, Nergiz ME, *et al*. Privacy-preserving publishing of hierarchical data. ACM Transactions on Privacy and Security, 2016, 19(3): 7.

7 Sweeney L. *k*-Anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-based Systems, 2002, 10(5): 557–570. [doi: 10.1142/S0218488502001648]

8 Meng XR, Kamara S, Nissim K, *et al*. GRECS: Graph encryption for approximate shortest distance queries. Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. Denver: ACM, 2015. 504–517.

9 Ding XF, Liu P, Jin H. Privacy-preserving multi-keyword top-*k* similarity search over encrypted data. IEEE Transactions on Dependable and Secure Computing, 2019, 16(2): 344–357. [doi: 10.1109/TDSC.2017.2693969]

10 马苏杭, 龙士工, 刘海, 等. 面向高维数据发布的个性化差分隐私算法. 计算机系统应用, 2021, 30(4): 131–138. [doi: 10.15888/j.cnki.csa.007870]

11 Dwork C. Differential privacy. Proceedings of the 33rd International Colloquium on Automata, Languages and Programming. Venice: Springer, 2006. 1–12.

12 Ding XF, Wang C, Choo KKR, *et al*. A novel privacy preserving framework for large scale graph data publishing. IEEE Transactions on Knowledge and Data Engineering, 2021, 33(2): 331–343.

13 许佳钰, 章红艳, 许力, 等. 社会网络中基于节点平均度的 *k*-度匿名隐私保护方案. 计算机系统应用, 2021, 30(12): 308–316. [doi: 10.15888/j.cnki.csa.008230]

14 黄海平, 王凯, 汤雄, 等. 基于边介数模型的差分隐私保护方案. 通信学报, 2019, 40(5): 2019095.

15 Huang HP, Zhang DJ, Xiao F, *et al*. Privacy-preserving approach PBCN in social network with differential privacy. IEEE Transactions on Network and Service Management, 2020, 17(2): 931–945. [doi: 10.1109/TNSM.2020.2982555]

16 Zhu TQ, Xiong P, Li G, *et al*. Differentially private model publishing in cyber physical systems. Future Generation Computer Systems, 2020, 108: 1297–1306. [doi: 10.1016/j.future.2018.04.016]

17 Casas-Roma J, Herrera-Joancomartí J, Torra V. A survey of graph-modification techniques for privacy-preserving on networks. Artificial Intelligence Review, 2017, 47(3): 341–366. [doi: 10.1007/s10462-016-9484-8]

18 Huang XZ, Liu JQ, Han Z, *et al*. Privacy beyond sensitive

values. Science China Information Sciences, 2015, 58(7): 1–15.

19 Mahanan W, Chaovalitwongse WA, Natwichai J. Data privacy preservation algorithm with *k*-anonymity. World Wide Web, 2021, 24(5): 1551–1561. [doi: 10.1007/s11280-021-00922-2]

20 Singh A, Singh M, Bansal D, *et al*. Optimised K-anonymisation technique to deal with mutual friends and degree attacks. International Journal of Information and Computer Security, 2021, 14(3–4): 281–299.

21 Liu F. Generalized Gaussian mechanism for differential privacy. IEEE Transactions on Knowledge and Data Engineering, 2019, 31(4): 747–756. [doi: 10.1109/TKDE.2018.2845388]

22 Wang Q, Zhang Y, Lu X, *et al*. Real-time and spatio-temporal crowd-sourced social network data publishing with differential privacy. IEEE Transactions on Dependable and Secure Computing, 2018, 15(4): 591–606.

23 Backes M, Berrang P, Humbert M, *et al*. Membership privacy in microrna-based studies. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. Vienna: ACM, 2016. 319–330.

24 张强, 叶阿勇, 叶帼华, 等. 最优聚类的 *k*-匿名数据隐私保护机制. 计算机研究与发展, 2022, 59(7): 1625–1635.

25 Zhou B, Pei J. The *k*-anonymity and *l*-diversity approaches for privacy preservation in social networks against neighborhood attacks. Knowledge and Information Systems, 2011, 28(1): 47–77. [doi: 10.1007/s10115-010-0311-2]

26 Liu Q, Wang GJ, Li F, *et al*. Preserving privacy with probabilistic indistinguishability in weighted social networks. IEEE Transactions on Parallel and Distributed Systems, 2017, 28(5): 1417–1429. [doi: 10.1109/TPDS.2016.2615020]

(校对责编: 牛欣悦)