计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2021,30(12):308−316 [doi: 10.15888/j.cnki.csa.008230]

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

社会网络中基于节点平均度的 k-度匿名隐私保护方案①

许佳钰[1,2], 章红艳[1,2], 许　力[1,2], 周赵斌[1,2]

[1](福建师范大学 数学与信息学院, 福州 350007)
[2](福建省网络安全与密码技术重点实验室, 福州 350007)
通讯作者: 许　力, E-mail: xuli@fjnu.edu.cn

摘　要: 社会网络数据的发布可能导致用户隐私被泄露, 例如用户的身份信息可能被恶意攻击者通过分析网络中节点的度数识别出来, 针对这个问题提出一种基于节点平均度的 k-度匿名隐私保护方案. 方案首先利用基于平均度的贪心算法对社会网络节点进行划分, 使得同一分组中节点的度都修改成平均度, 从而生成 k-度匿名序列; 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. 本方案在生成 k-度匿名序列时引入平均度, 提高了聚类的精度, 降低了图结构修改的代价. 同时, 由于在图结构修改时考虑了衡量边重要性的指标—邻域中心性, 重要的边被优先保留, 保持了稳定的网络结构. 实验结果表明, 本方案不仅能有效地提高网络抵抗度攻击的能力, 还能极大降低信息损失量, 在保护用户隐私的同时提高了发布数据的可用性.

关键词: 社会网络; 隐私保护; k-度匿名; 平均度; 重要边

引用格式: 许佳钰,章红艳,许力,周赵斌.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. http://www.c-s-a.org.cn/1003-3254/8230.html

k-Degree Anonymous Privacy Protection Scheme Based on Average Degree of Node in Social Networks

XU Jia-Yu[1,2], ZHANG Hong-Yan[1,2], XU Li[1,2], ZHOU Zhao-Bin[1,2]

[1](College of Mathematics and Informatics, Fujian Normal University, Fuzhou 350007, China)
[2](Fujian Provincial Key Laboratory of Network Security and Cryptology, Fuzhou 350007, China)

Abstract: The release of social network data may lead to the disclosure of user privacy; for example, the user identity may be recognized by malicious attackers by analyzing the degree of nodes in the network. Concerning this problem, a k-degree anonymous privacy protection scheme based on the average degree of nodes is proposed. The scheme first depends on the greedy algorithm based on the average degree to divide social network nodes, so that the degrees of nodes in the same group are modified to the average degree, thus generating k-degree anonymous sequences; then the graph structure modification method with priority to retain important edges is used to modify the graph, thus achieving k-degree anonymity of the graph. In this scheme, the average degree is introduced when k-degree anonymous sequences are generated, which improves clustering accuracy and reduces the cost of graph structure modification. At the same time, because the indicator-neighborhood centrality, which measures the importance of edges, is considered in the graph structure modification, important edges are retained in preference, and a stable network structure is maintained. The experimental results show that this scheme improves the network resistance to degree attacks, greatly reduces information

① 基金项目: 国家自然科学基金 (U1905211, 61771140, 61702100, 61702103); 福建省教育厅中青年科研项目 (JAT200968); 企事业合作项目 (DH-1565, DH-1412)
Foundation item: National Natural Science Foundation of China (U1905211, 61771140, 61702100, 61702103); Mid-Aged and Young Talent S & T Program of Education Bureau, Fujian Province (JAT200968); Enterprise Cooperation Project (DH-1565, DH-1412)
收稿时间: 2021-03-04; 修改时间: 2021-03-31; 采用时间: 2021-04-16

loss, and improves the utility of published data while protecting user privacy.

Key words: social network; privacy protection; k-degree anonymity; average degree; important edges

近年来, 随着使用微博、Facebook、Twitter 等社交网站的用户数快速增加, 产生 了大规模的社会网络数据. 这些数据具有巨大的商业价值和应用场景, 同样也包含了很多敏感信息[1]. 研究者开发出了大量的数据挖掘技术和社会网络分析方法, 用来挖掘和分析这些数据背后的价值. 但如果发布的数据被不正确使用, 用户可能会遭到恶意攻击和面临隐私泄露问题[2], 在数据挖掘的过程中需要保护用户隐私[3]. 民法典中也确立了平衡个人信息保护与信息合理使用之间的基本准则. 因此, 对发布的社会网络数据进行隐私保护尤为重要[4], 在发布数据的同时应该保护好个人隐私信息. 如何在有效地保护用户隐私的同时又能保证发布的数据具有可用性[5], 这是人们一直在研究的问题.

数据的隐私保护问题已经得到了广泛的研究, Sweeney[6] 在 2002 年最早提出 k-匿名模型, 而最近趋向于个性化的 k-匿名[7] 的研究. 在 k-匿名模型提出之后, l-多样化[8], t-接近[9] 等隐私保护模型也被先后提出. 然而社会网络中的节点之间存在相关性, 如果仅对节点进行匿名处理, 攻击者仍可能会根据边权值或图结构对网络进行攻击[10−12]. 目前针对社会网络数据的隐私保护方法大致可分为基于聚类和基于图修改两种.

其中, 基于聚类的社会网络隐私保护方法是通过特定的聚类规则将一些节点和边进行聚类, 然后通过泛化达到匿名化效果. Hay 等[13] 提出对网络中相似节点进行聚合, 聚合后每个块中所包含的节点数 n 满足 $k \le n \le 2k-1$的条件, 这样使得攻击成功的概率不高于 $1/k$. Skarkala 等[14] 使用节点聚类和边聚类相结合的方法对加权无向网络进行泛化, 以实现 k-匿名. 姜火文等[15] 利用属性图表示社交网络数据, 综合根据节点间的结构和属性相似度, 将图中所有节点聚类成一些包含节点个数不小于 k 的超点. 然而, 基于聚类的社会网络隐私保护方法将节点聚类成超点或将边聚类成超边会导致严重的边信息损失, 破坏网络结构, 大大降低数据可用性.

通过图修改的方法实现社会网络数据的隐私保护方法已成为近些年来研究者关注的热点. Liu 等[16] 首次提出图的 k-度匿名化概念, 并采用增加边的图结构修改方式来实现图的 k-度匿名化, 以抵抗节点度攻击. Chester 等[17] 首次提出一种加边与加点相结合的方法来实现 k-度匿名图. 针对节点具有标签的社会网络, 文献 [18,19] 都提出了 k-度-l 多样化匿名模型, 该模型在满足 k 度匿名的基础上, 要求度数相同的 k 个节点至少要有 l 种不同标签, 并通过增加或删除边以及添加噪声节点的方法实现匿名. Casas-Roma 等[20] 采用穷举法和贪心算法生成度匿名序列, 通过邻居中心性边选择方法和随机边选择方法实现 k-度匿名. 周克涛等[21] 针对传统的 k-度匿名方法添加的噪声数据过多, 提出了改进的基于邻居度序列相似度的 k-度匿名保护方法, 可以抵御以节点的度结合邻居度序列作为背景知识的攻击. Macwan 等[22] 提出了改进的 k-度匿名方法, 该模型保留了网络结构属性以及用户隐私. Kamalkumar 等[23] 针对大规模社会网络提出快速的隐私保护方法, 对小社区实施个性化 k-度匿名化. Kiabod 等[24] 引入一种节省时间的 k-度匿名化方法, 该方法利用基于树结构计算图的匿名度序列, 利用基于匿名化级别对图形自底向上节点进行分区, 实现隐私保护级别的动态变化. 张晓琳等[25] 针对大规模社会网络有向图, 提出了一种基于层次社区结构的大规模社会网络 k-出入度匿名方法, 该方法提高了处理大规模社会网络有向图数据的效率, 并在匿名后保证了数据发布时社区结构分析的可用性. 以上这些基于图修改的社会网络隐私保护方法大多都采用添加、删除边或添加节点以及子图同构等扰动方式实现 k-度匿名, 但这些方法还存在信息损失较为严重的问题.

针对以上两种方法存在的问题, 本文提出一种基于节点平均度的 k-度匿名隐私保护方案, 用来解决社会网络数据的发布可能导致用户隐私泄露的问题, 在保护用户隐私的同时提高了发布数据的可用性. 本方案首先利用基于平均度的贪心算法生成 k-度匿名序列, 然后采用优先保留重要边的图结构修改方法来对图进行修改, 实现图的 k-度匿名化. 本方案不仅能有效地提高网络抵抗度攻击的能力, 还能克服传统方案在对网络匿名后所产生的信息损失严重的问题, 在保护用户隐私的同时提高了发布数据的可用性.

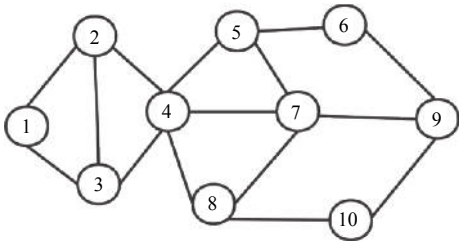
1 相关定义

在本文中, 用一个无向无权的图来表示社会网络, $G=(V,E)$, V 代表用户实体, E 代表实体间的关系, $V=\{v_1, v_2,\cdots,v_n\}$是节点的集合, $E=\{(v_i,v_j)|v_i,v_j\in V\}$是边的集合且$1\le i,j\le n$. d_G代表图 G 的度序列, $d=[d_1,d_2,\cdots,d_n]$. 其中, $d(i)$或$d(v_i)$代表图中第 i 个节点v_i的度.

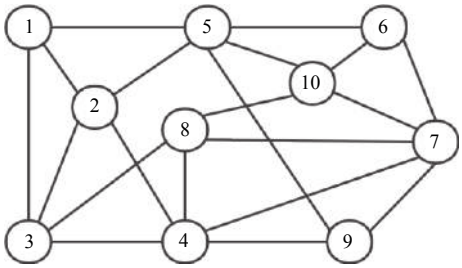
定义 1. 向量 k-匿名. 如果整数向量 V 是 k-度匿名的, 那么向量 V 中每个值都出现至少 k 次. 例如, 向量 $V=[5,5,3,3,2,2,2]$是 2-度匿名的.

定义 2. 图的 k-度匿名. 如果图 G 的度数序列 d 是 k-匿名的, 那么图 $G=(V,E)$ 是 k-度匿名的.

显然, 只要找到原图的最优 k-度匿名向量, 就可以根据该向量在原图基础上增补出新的 k-度匿名图. 如图 1(a) 是一个没有进行过度匿名的原始图, 度序列为 [2,3,3,5,3,2,4,3,3,2], 给定度序列和相应的节点 ID 序列. 由于只有节点 4 具有度 5, 而只有节点 7 具有度 4, 所以任何人都可以重新识别出节点 4 和节点 7. 将图 1(a) 匿名成图 1(b), 度序列为 [3,4,4,5,3,5,5,4,3,4], 变成一个 3-度匿名图, 这时图中任何一个节点都至少有 2 个节点与其度数相同, 可以重新识别节点 4 或节点 7 的概率都减少到 1/3.



(a) 原始图



(b) 3-度匿名图

图 1 匿名前后对比图

因此, 如果一个图满足 k-度匿名, 则表明图中任一个节点至少与其他 k–1 个节点具有相同的度, 利用节点度数作为背景知识的攻击者能够识别目标个体的概率不超过 $1/k$[16]. 我们用 k 值这个指标来衡量社会网络的抵御攻击能力, k 值越大, 目标个体身份被攻击者识别的概率就越小, 社会网络的抵御攻击能力就越强, 隐私保护程度也就越高.

根据以上定义, 为了将输入图 G 转换为结构上类似于 G 的 k-度匿名图$\hat{G}$, 我们首先需要将 G 的度序列 d 转换成 k-度匿名序列, 然后根据度匿名序列对 G 进行图结构修改构造出$\hat{G}$, 我们在图结构修改时是通过增加、删除或者交换边来实现节点度数的调整. 以单纯对边进行操作, 不增加节点的策略为例, 我们希望选择边变化最少 (度数变化最少) 的方案来实现 k-度匿名, 这样可以保证匿名前后图结构的相似性.

定义 3. 基于平均度的图匿名代价. 图匿名代价可用匿名前后边的变化数来计算, 根据握手定理可知一条边贡献两个度, 即匿名前后度的总变化数刚好是边的总变化数的两倍, 因此边的变化数与度的变化数直接相关.

图匿名代价$G_A(G,\hat{G})$由式 (1) 计算得到:

$$G_A(\hat{G},G)=\left|\hat{E}\right|-|E|=\frac{1}{2}L_1(\hat{d}-d) \tag{1}$$

其中, 匿名前后度序列之间的距离$L_1(\hat{d}-d)$由式 (2) 计算得到:

$$L_1(\hat{d}-d)=\sum_i|\hat{d}(i)-d(i)| \tag{2}$$

对于 $i<j$, 节点 i 到 j 之间的所有节点的平均度为 $d(a)$, 由式 (3) 计算得到:

$$d(a)=\frac{\sum_{l=i}^{j}d(l)}{j-i+1} \tag{3}$$

如果节点 i 到 j 之间的所有节点形成同一个匿名组, 同一组中所有节点的度都匿名成该组所有节点的平均度$d(a)$, 则有式 (4) 成立:

$$\hat{d}(i)=\hat{d}(i+1)=\cdots=\hat{d}(j-1)=\hat{d}(j)=d(a) \tag{4}$$

将该组的匿名代价记为$I(d[i,j])$, 则基于节点平均度的 k-度匿名代价由式 (5) 计算得到:

$$I(d[i,j])=\sum_{l=i}^{j}\left|\hat{d}(i)-d(i)\right|=\sum_{l=i}^{j}|d(a)-d(i)| \tag{5}$$

在进行图修改操作时要实现式 (5) 的最小化, 以保持数据可用性.

2 社会网络的 k-度匿名隐私保护方案

本节提出了一种基于节点平均度的 k-度匿名隐私保护方案, 方案主要包括度匿名序列生成和图结构修改两个阶段. 首先利用基于平均度的贪心算法对社会网络节点度序列进行划分, 生成 k-度匿名序列; 然后根据生成的 k-度匿名序列对图进行修改实现图的 k-度匿名化, 修改时采用优先保留重要边的图结构修改方法. 与传统方案相比, 在对图结构的修改程度一样的前提下, 本方案可以达到更大的 k 值, 说明本方案相对传统方案的抵御攻击能力有显著提高, 提供了更强的隐私保护.

本文中的符号说明见表 1.

表 1 符号说明

符号	含义
V	点集
E	边集
G	图
$\hat{G}$	匿名图
d	度序列
$\hat{d}$	匿名度序列
$d(a)$	平均度
$\Gamma(v_i)$	节点v_i的相邻节点集合
$L_1(\hat{d}-d)$	匿名前后度序列之间的距离
$D_A(d[1,n])$	度序列d的匿名代价
$I(d[i,j])$	同一匿名组中的第i到第j个节点的匿名代价

2.1 度匿名序列生成

本方案利用基于平均度的贪心算法 (算法 1) 生成原始图的 k-度匿名序列.

下面给出了生成 k-度匿名序列的算法.

算法 1. k-度匿名序列生成算法

输入: 原始图 G, 正整数 k
输出: k-度匿名序列$\hat{d}$

1. d←degree sequence of G
2. *sort*(d)
3. put the first k nodes into group g_i
4. $i=1$
5. **while** until every node gets the group **do**
6. count C_{merge}, C_{new}
7. **if** C_{merge}> C_{new}**then**
8. i++
9. put nodes n_{k+1} ~n_{2k} into new group g_i
10. $k=k+2k$
11. **else**
12. put node n_{k+1} into group g_i
13. $k=k+1$
14. **end if**
15. count $d_a(g_i)$//计算的平均度
16. the degree of node in g_i becomes $d_a(g_i)$
17. **end while**

该算法将社会网络图 G 和整数 k 作为输入, 首先找到输入图 G 的度数序列并将其按度数降序的顺序进行排序, 然后将前 k 个节点放入同一组, 接着分别根据式 (6) 和式 (7) 计算比较 C_{merge} 和 C_{new} 两个成本, 来决定应该将第 (k+1) 个节点合并到当前的分组中, 还是在位置 (k+1) 处开始一个新组. 其中 C_{merge} 表示把第 (k+1) 个节点合并到当前分组所产生的成本, 由式 (6) 计算得到; C_{new} 表示把第 (k+1) 个节点放入一个新的分组所产生的成本, 由式 (7) 计算得到.

$$C_{\text{merge}} = |d(a)-d(k+1)| + I(d[k+2,2k+1]) \quad (6)$$

$$C_{\text{new}} = I(d[k+1,2k]) \quad (7)$$

当 C_{merge}> C_{new} 时, 将第 k+1~2k 的节点放入一个新的分组, 然后计算和比较第 2k+1 个节点的成本并放入相应的分组中, 以此类推.

当 C_{merge}< C_{new} 时, 将第 k+1 个节点合并到上一个分组, 然后计算和比较第 k+2 个节点的成本并放入相应的分组中, 以此类推.

直到将所有节点分完组后, 计算每一个分组中节点的所有节点的平均度 $d(a)$, 然后令该组中所有节点的度都变为平均度, k-度匿名序列生成.

2.2 图结构修改

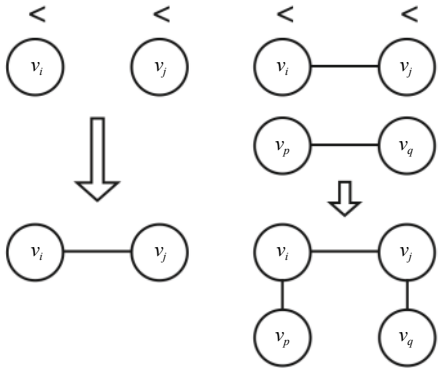
在上一节中, 原始图的度序列已经被匿名成为 k-度匿名序列. 根据生成的 k-度匿名序列对图结构进行修改, 使得修改后的匿名图的度序列满足匿名要求.

本方案中进行图结构修改时对边的操作方式主要包含以下 3 种:

(1) 边增加策略: 如图 2 所示, 我们选择两个不同节点$v_i,v_j \in V$, 若$(v_i,v_j) \notin E$, 可以在节点v_i,v_j之间添加一条边(v_i,v_j), 两个节点的度同时增加 1; 若$(v_i,v_j) \in E$, 此时需要找到节点v_i的不相邻节点集合$\overline{\Gamma}(v_i)$, 以及v_j的不相邻节点集合$\overline{\Gamma}(v_j)$, 在$\overline{\Gamma}(v_i)$和$\overline{\Gamma}(v_j)$中分别找到两个节点v_p,v_q满足$(v_p,v_q) \in E$, 删除边(v_p,v_q),同时增加边(v_p,v_i)与边(v_q,v_j),此时我们可以看出两个节点v_i,v_j的度同时增加 1, 而节点v_p,v_q的度没有变化, 并且增加了一条边.

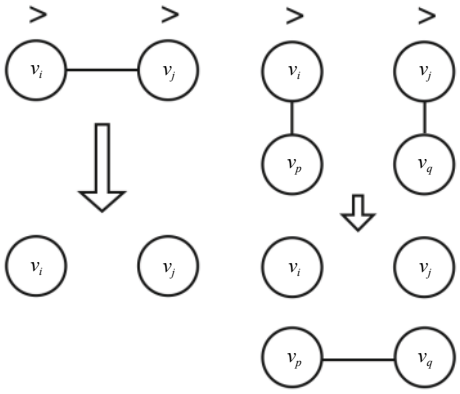
(2) 边删除策略: 如图 3 所示, 我们选择两个不同节点$v_i,v_j \in V$, 若$(v_i,v_j) \in E$, 此时我们可以在节点v_i,v_j

之间删除边(v_i,v_j), 两个节点的度同时减 1; 若$(v_i,v_j)\notin E$, 此时我们需要找到节点v_i的相邻节点集合$\Gamma(v_i)$, 以及v_j的相邻节点集合$\Gamma(v_j)$. 在$\Gamma(v_i)$和$\Gamma(v_j)$中分别找到两个节点v_p,v_q满足$(v_p,v_q)\notin E$, 增加边(v_p,v_q), 同时删除边(v_p,v_i)与边(v_q,v_j). 此时节点v_i,v_j的度同时减少 1, 而v_p,v_q的度没有变化, 并且减少了一条边.



(a) 不存在边的情况下 (b) 存在边的情况下

图 2 边增加策略的方式



(a) 存在边的情况下 (b) 不存在边的情况下

图 3 边删除策略的方式

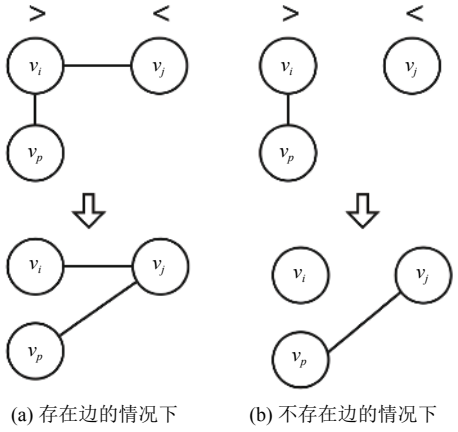
(3) 边交换策略: 如图 4 所示, 在边交换策略中, (v_j,v_p)与$(v_i,v_j)\notin E$这两种情况的边操作是一样的, 需要同时对 3 个点进行操作. 如果$v_i,v_j,v_p\in V$, 且同时满足$(v_i,v_p)\in E$和$(v_j,v_p)\notin E$, 此时删除边(v_i,v_p), 增加边(v_j,v_p), 节点v_i的度减少 1, 而节点v_j的度增加了 1, v_p的度和边数没有变化.

原始图中节点v的度数$d(v)$与其所属分组的平均度数$d(a)$之间可能存在的大小关系有如下 3 种情况:

(1) 当$d(v)<d(a)$时, $\Delta d(v)>0$, 节点v需要通过执行边增加策略使节点度数增加.

(2) 当$d(v)>d(a)$时, $\Delta d(v)<0$, 节点v需要通过执行边删除策略使节点度数减少.

(3) 当$d(v)=d(a)$时, $\Delta d(v)=0$, 节点v满足匿名化, 不需要进行任何操作.



(a) 存在边的情况下 (b) 不存在边的情况下

图 4 边交换策略的方式

我们在进行图结构修改时只需考虑还未满足度匿名化的节点, 对于满足度匿名化的节点可直接跳过无需进行任何操作. 对于还未满足度匿名化的节点, 需要根据两个点间的度数大小关系以及有无连边的情况选择相应的图修改策略.

任意两个节点v_i,v_j, 两者都满足$\Delta d(v)<0$, 则需要执行图 2 中的边增加策略, 还需根据两个节点之间是否存在边选择相应的操作方式. 当两个节点之间不存在边时, 则选择操作方式如图 2(a); 当两个节点之间存在边时, 则选择操作方式如图 2(b).

任意两个节点v_i,v_j, 两者都满足$\Delta d(v)>0$, 则需要执行图 3 中的边删除策略, 还需根据两个节点之间是否存在边选择相应的操作方式. 当两个节点之间不存在边时, 则选择操作方式如图 3(a); 当两个节点之间存在边时, 则选择操作方式如图 3(b).

任意两个节点v_i,v_j, 其中一个满足$\Delta d(v)>0$, 另一个满足$\Delta d(v)<0$时, 则需要执行图 4 中的边交换策略, 无论两个节点之间是否存在边, 操作方式都是一样的.

在边操作中选择边的时候, 要考虑保留重要的边, 方案中我们利用了邻域中心性 (Neighbourhood Centrality, *NC*) 值来量化大型网络的边缘相关性[20]. 边(v_i,v_j)的邻域中心性定义为同时与v_i或v_j相邻, 但不同时与v_i和v_j相邻的节点的比例, 由式 (8) 计算得到:

$$NC\{v_i,v_j\}=\frac{\left|\Gamma(v_i)\cup\Gamma(v_j)\right|-\left|\Gamma(v_i)\cap\Gamma(v_j)\right|}{2\max(deg)} \quad (8)$$

NC 值越小, 说明该边的边相关程度就越低, 该边的重要程度比较低; *NC* 值越大, 说明该边的边相关程度就越高, 该边的重要程度比较低. 为了降低图修改前后的信息损失量, 本方案在进行边操作时选择 *NC* 值较低的边.

本方案的算法在执行时会多次遍历, 直达图结构修改完成, 具体的图结构修改算法如算法 2 所示.

算法 2. 图结构修改算法

输入: 原始图 G, 原始图度序列 d, k-度匿名序列
输出: k-度匿名图 $\hat{G}$

1. **while** True **do**
2. $\Delta d=\hat{d}-d$
3. $sort(|\Delta d|)$
4. pick node $v_i=|\Delta d|_{\max}$
5. pick node v_j randomly, $|\Delta d|_{v_j}\neq 0$
6. **if** $\Delta d_{v_j}>0$ and $\Delta d_{v_j}>0$ **then**
7. **if** there exits (v_i,v_j) **then**
8. pick two nodes v_p,v_q randomly (exits (v_p,v_q) and $v_p,v_q\notin\Gamma(v_j)$ and $v_p,v_q\notin\Gamma(v_j)$)
9. delete (v_p,v_q), add (v_i,v_p), add (v_j,v_p)
10. **else**
11. add (v_i,v_j)
12. Δd_{v_i}++, Δd_{v_j}++
13. **end if**
14. **if** $\Delta d_{v_i}<0$ and $\Delta d_{v_j}<0$ **then**
15. **if** there exits (v_i,v_j) **then**
16. delete (v_i,v_j)
17. **else**
18. pick two nodes v_p,v_q randomly (not exits (v_p,v_q) and $v_p\in\Gamma(v_i)$ and $v_q\in\Gamma(v_j)$)
19. delete (v_i,v_p), delete (v_j,v_q), add (v_p,v_q)
20. Δd_{v_i}−−, Δd_{v_j}−−
21. **end if**
22. **if** $\Delta d_{v_i}<0$ and $\Delta d_{v_j}>0$ **then**
23. pick node v_p randomly ($v_p\in\Gamma(v_i)$and$v_p\notin\Gamma(v_j)$)
24. delete (v_i,v_p), add (v_j,v_p)
25. Δd_{v_i}−−, Δd_{v_j}++
26. **if** $\Delta d_{v_i}>0$ and $\Delta d_{v_j}<0$ **then**
27. pick node v_q randomly ($v_q\notin\Gamma(v_i)$ and $v_q\in\Gamma(v_j)$)
28. add (v_i,v_p), delete (v_j,v_p)
29. Δd_{v_i}++, Δd_{v_j}−−
30. **end while**

在上述算法中, 将原始图 G 和匿名前后的度序列作为输入, 首先计算每个节点需增加的度数$|\Delta d(v)|$, 这个过程的时间复杂度为 O(n), 并对其进行降序排序, 排序过程的时间复杂度 O($n\log_2 n$). 然后选取$|\Delta d(v)|$值最大的节点v_i和$|\Delta d(v)|$值非零的节点v_j, 判断v_i与v_j之间的度数关系以及有无连边, 执行相应的边操作, 并更新两个节点相应的$\Delta d(v)$值, 这个过程的时间复杂度为 O(1). 根据更新后的$\Delta d(v)$值继续选择操作的节点, 重复以上步骤, 直到所有节点的$\Delta d(v)$值为 0, 则图结构修改完成, 总重复次数为 O(n) 级别. 因此, 该算法的总时间复杂度为 O(n^2) 级别. 但是在真实社交网络中 算法仍有较好的执行效率, 能够满足实际需求.

3 实验结果分析

本方案使用 Facebook 数据集进行仿真实验, 来源于 Stanford 大学的一个公开数据库 SNAP[26], 该数据集说明了 Facebook 社交网站上的各个用户之间的关系, 包含节点数 4039 个, 无向边 88234 条, 节点的总度数为 176468 度, 平均度数为 43 度, 且节点的度服从幂律分布. 算法代码用 Python 编程实现, 实验环境为 Intel Core i5 CPU 1.4 GHz, 16 GB 内存, 操作系统为 MacOS.

对于社会网络中的图数据, 在进行匿名隐私保护的同时保持其可用性是非常重要的[10]. 为了说明本文提出的 k-度匿名隐私保护方案的有效性, 我们通过计算边的变化率来说明数据的信息损失量, 其中边的变化率为匿名前后边的变化数与原始图中的边总数之比. 数据的信息损失量越小, 则数据的可用性越好. 另外我们还考虑了图结构的一些基本属性, 主要测试平均聚类系数、平均最短路径、节点平均度这 3 个指标. 单个节点的聚类系数是它所有相邻节点之间连边的数目占可能的最大连边数目的比例, 而整个网络的平均聚类系数就是所有节点簇系数的平均值; 平均最短路径是网络中所有结点对的距离的平均值; 节点平均度是网络中所以节点的度数之和与节点总数之比. 我们将实验前的数据与进行 k-度匿名后的数据进行对比, 同时与文献 [16,20] 的方案进行对比来, 以验证本方案的有效性.

图 5 展示的是本方案和文献 [16,20] 的方案在匿名前后信息损失量的变化结果, 表 2 是具体的实验数据. 通过对比分析在匿名前后网络的边变化率来衡量信息损失量. 如图 5 所示, 随着 k 值的增大, 3 种方案造成的信息损失量也都跟着变大, 但是相比于文献 [16,20] 两种方案, 本方案信息损失量是最小的, 更好地保持了数据的高可用性.

图 6 展示了本方案和文献 [16,20] 方案在匿名前后聚类系数的变化结果, 表 3 是具体的实验数据. 如图 6 所示, 与文献 [16,20] 两种方案相比, 本方案使得匿名后的网络在不同的 k 值下始终最接近于原始网络的平均聚类系数值, 对网络平均聚类系数的影响明显小于另外两种方案. 文献 [16] 的方案在图结构修改时没有考虑保留重要的边, 使得匿名后网络的平均聚类系数与原始网络相比有较大改变, 对图结构的修改较大. 而本方案在图结构修改时引入了 NC 值来保留重要的边, 匿名前后图结构变化非常小. 由此可见, 本方案在实现 k-度匿名保护用户隐私同时, 还能保持数据具有较高的可用性.

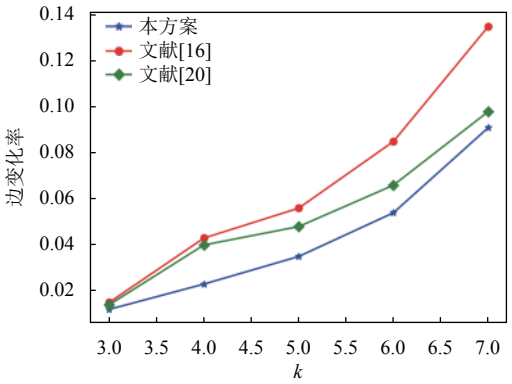


图 5　信息损失量对比图

表 2　信息损失量对比

k值	本方案	文献[16]	文献[20]
k=3	0.012	0.015	0.014
k=4	0.023	0.043	0.040
k=5	0.035	0.056	0.048
k=6	0.054	0.085	0.066
k=7	0.091	0.135	0.098

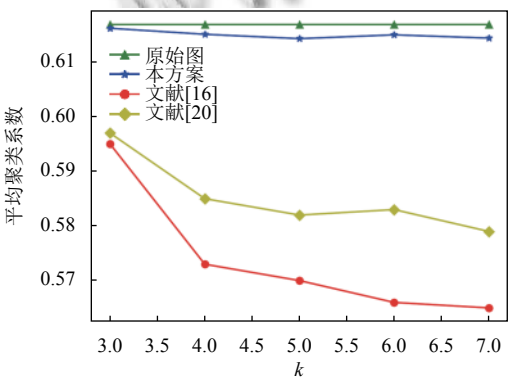


图 6　平均聚类系数对比图

表 3　平均聚类系数对比

k值	原始图	本方案	文献[16]	文献[20]
k=3	0.6169	0.6162	0.5954	0.5973
k=4	0.6169	0.6151	0.5735	0.5852
k=5	0.6169	0.6143	0.5701	0.5821
k=6	0.6169	0.6150	0.5662	0.5833
k=7	0.6169	0.6144	0.5653	0.5791

图 7 展示的是本方案和文献 [16,20] 的方案在匿名前后平均最短路径的变化结果, 表 4 是具体的实验数据. 如图 7 所示, 随着 k 值的增加, 3 种方案的平均最短路径都在减小, 但是文献 [16] 的方案使得匿名后网络的平均最短路径始终大于原始网络, 较大程度地破坏了网络结构. 当 k 值较小时, 文献 [20] 的方案使得匿名后网络的平均最短路径始大于原始网络, 对网络结构的破坏较大; 当 k 值较大时, 文献 [20] 的方案使得匿名后网络的平均最短路径始小于原始网络, 对网络结构的破坏较小. 而本方案使得匿名后网络的平均最短路径始终小于原始网络, 较好地保持了网络结构的稳定性.

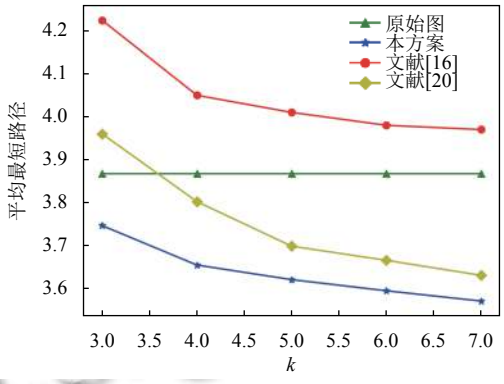


图 7　平均最短路径对比图

表 4　平均最短路径对比

k值	原始图	本方案	文献[16]	文献[20]
k=3	3.8674	3.7462	4.2251	3.9601
k=4	3.8674	3.6541	4.0501	3.8023
k=5	3.8674	3.6203	4.0121	3.6984
k=6	3.8674	3.5941	3.9832	3.6651
k=7	3.8674	3.5702	3.9723	3.6302

图 8 展示的是本方案和文献 [16,20] 的方案在匿名前后节点平均度的变化结果, 表 5 是具体的实验数据. 如图 8 所示, 随着 k 值的增加, 本方案对原始网络节点平均度的改变量最小, 匿名后网络的节点平均度数与原始网络基本相同. 文献 [20] 的方案对原始网络节点平均度的改变程度略高于本方案. 文献 [16] 的方案使得匿名前后网络的节点平均度有较大改变, 对原

始网络结构的破坏较为严重.

图 9 展示了本方案和文献 [16,20] 的方案在运行时间上的比较结果, 表 6 是具体的实验数据. 如图 9 所示, 当 k 值较小时, 3 个方案的算法运行时间大致相同, 当 k 值较大时, 本方案的运行时间要高于文献 [16,20] 的方案. 但总体来说, 本方案的运行时间不会比另外两个方案高出很多, 且本方案使得社会网络在抵御度攻击方面和保持图数据可用性方面均有了较好的改进, 因此这样稍高的时间复杂度还是在可接受范围内.

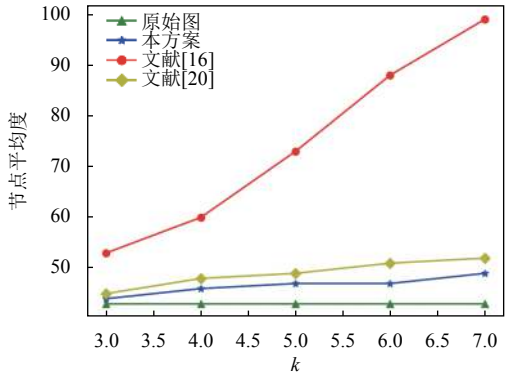


图 8 节点平均度对比图

表 5 节点平均度对比

k值	原始图	本方案	文献[16]	文献[20]
k=3	43	44	53	45
k=4	43	46	60	48
k=5	43	47	73	49
k=6	43	47	88	51
k=7	43	49	99	52

4 结束语

针对社会网络数据的发布可能遭到度攻击进而导致用户隐私泄露的问题, 本文提出一种基于节点平均度的 k-度匿名隐私保护方案, 本方案在保护用户隐私的同时保证了发布的数据具有较高可用性. 首先利用基于平均度的贪心算法对社会网络节点度序列进行划分, 使得同一分组中节点的度都修改成平均度, 生成 k-度匿名序列, 极大地减少了与原始度序列的距离; 然后使用边增加、边删除、边交换 3 种边操作方式对原始图进行图结构修改, 由于对边进行操作时考虑了 NC 值, 保留了网络中重要的边, 匿名后的网络保持了较好的连通性和关系结构, 从而提高了发布数据的可用性. 本方案不仅能有效提高社会网络抵御度攻击的能力, 还能保持网络结构的高稳定性和发布数据的高可用性. 但是在算法的时间复杂度方面, 与其它方案相比优势不够明显, 因此还需要在未来进一步研究如何减小算法的时间复杂度.

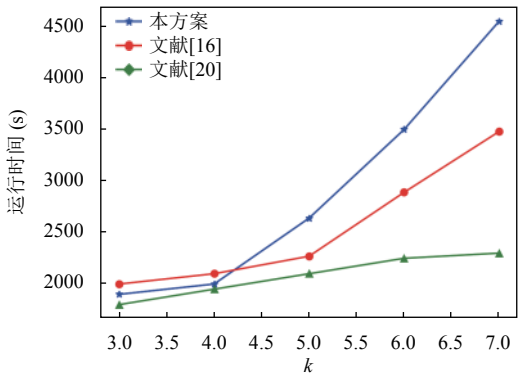


图 9 执行时间对比图

表 6 执行时间对比 (s)

k值	本方案	文献[16]	文献[20]
k=3	1900	2000	1800
k=4	2000	2100	1950
k=5	2640	2270	2100
k=6	3500	2890	2250
k=7	4550	3480	2300

参考文献

1 Rathore S, Sharma PK, Loia V, *et al*. Social network security: Issues, challenges, threats, and solutions. Information Sciences, 2017, 421: 43–69. [doi: 10.1016/j.ins.2017.08.063]

2 Li HX, Chen QR, Zhu HJ, *et al*. Privacy leakage via de-anonymization and aggregation in heterogeneous social networks. IEEE Transactions on Dependable and Secure Computing, 2020, 17(2): 350–362. [doi: 10.1109/TDSC.2017.2754249]

3 方跃坚, 朱锦钟, 周文, 等. 数据挖掘隐私保护算法研究综述. 信息网络安全, 2017, (2): 6–11. [doi: 10.3969/j.issn.1671-1122.2017.02.002]

4 刘向宇, 王斌, 杨晓春. 社会网络数据发布隐私保护技术综述. 软件学报, 2014, 25(3): 576–590. [doi: 10.13328/j.cnki.jos.004511]

5 Ji SL, Mittal P, Beyah R. Graph data anonymization, de-anonymization attacks, and de-anonymizability quantification: A survey. IEEE Communications Surveys & Tutorials, 2017, 19(2): 1305–1326.

6 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]
7 何泾沙, 杜晋晖, 朱娜斐. 基于 *k* 匿名的准标识符属性个性化实现算法研究. 信息网络安全, 2020, 20(10): 19–26. [doi: 10.3969/j.issn.1671-1122.2020.10.003]
8 Machanavajjhala A, Kifer D, Gehrke J. *L*-diversity: Privacy beyond *k*-anonymity. ACM Transactions on Knowledge Discovery from Data, 2007, 1(1): 3–es. [doi: 10.1145/1217299.1217302]
9 Li NH, Li TC, Venkatasubramanian S. *t*-Closeness: Privacy beyond *k*-anonymity and *l*-diversity. 2007 IEEE 23rd International Conference on Data Engineering. Istanbul: IEEE, 2007. 106–115.
10 Qian JW, Li XY, Zhang CH, *et al*. Social network de-anonymization and privacy inference with knowledge graph model. IEEE Transactions on Dependable and Secure Computing, 2019, 16(4): 679–692. [doi: 10.1109/TDSC.2017.2697854]
11 Fang JB, Li AP, Jiang QY, *et al*. A structure-based de-anonymization attack on graph data using weighted neighbor match. 2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC). Hangzhou: IEEE, 2019. 480–486.
12 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]
13 Hay M, Miklau G, Jensen D, *et al*. Resisting structural re-identification in anonymized social networks. Proceedings of the VLDB Endowment, 2008, 1(1): 102–114. [doi: 10.14778/1453856.1453873]
14 Skarkala ME, Maragoudakis M, Gritzalis S, *et al*. Privacy preservation by *k*-anonymization of weighted social networks. 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. Istanbul: IEEE, 2012. 423–428.
15 姜火文, 占清华, 刘文娟, 等. 图数据发布隐私保护的聚类匿名方法. 软件学报, 2017, 28(9): 2323–2333. [doi: 10.13328/j.cnki.jos.005178]
16 Liu K, Terzi E. Towards identity anonymization on graphs. Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. Vancouver: ACM, 2008. 93–106.
17 Chester S, Kapron BM, Ramesh G, *et al*. *k*-anonymization of social networks by vertex addition. ADBIS, 2011, (2): 107–116.
18 Yuan MX, Chen L, Yu PS, *et al*. Protecting sensitive labels in social network data anonymization. IEEE Transactions on Knowledge and Data Engineering, 2013, 25(3): 633–647. [doi: 10.1109/TKDE.2011.259]
19 Zhou B, Pei J. Preserving privacy in social networks against neighborhood attacks. 2008 IEEE 24th International Conference on Data Engineering. Cancun: IEEE, 2008. 506–515.
20 Casas-Roma J, Herrera-Joancomartí J, Torra V. *k*-Degree anonymity and edge selection: Improving data utility in large networks. Knowledge and Information Systems, 2017, 50(2): 447–474. [doi: 10.1007/s10115-016-0947-7]
21 周克涛, 刘卫国, 施荣华. 基于邻居度序列相似度的 *k*-度匿名隐私保护方案. 计算机工程与应用, 2017, 53(19): 102–108. [doi: 10.3778/j.issn.1002-8331.1604-0380]
22 Macwan KR, Patel SJ. *k*-Degree anonymity model for social network data publishing. Advances in Electrical and Computer Engineering, 2017, 17(4): 117–124. [doi: 10.4316/AECE.2017.04014]
23 Macwan KR, Patel SJ. *k*-NMF anonymization in social network data publishing. The Computer Journal, 2018, 61(4): 601–613. [doi: 10.1093/comjnl/bxy012]
24 Kiabod M, Dehkordi MN, Barekatain B. TSRAM: A time-saving *k*-degree anonymization method in social network. Expert Systems with Applications, 2019, 125: 378–396. [doi: 10.1016/j.eswa.2019.01.059]
25 张晓琳, 刘娇, 毕红净, 等. 大规模社会网络 K-出入度匿名方法. 计算机工程, 2020, 46(11): 164–173.
26 Stanford large network dataset collection. http://snap.stanford.edu/data/index.html. (2014-06-01).