Hans Journal of Data Mining 数据挖掘, 2020, 10(2), 143-151
Published Online April 2020 in Hans. http://www.hanspub.org/journal/hjdm
https://doi.org/10.12677/hjdm.2020.102015

Hans汉斯

An Privacy Attribute (a, k)-Anonymous Algorithm Based on Node Segmentation

Xiuqin Deng*, Yifei Zhang, Zhihua Jiang, Lihui Tan

School of Applied Mathematics, Guangdong University of Technology, Guangzhou Guangdong
Email: *dxq706@gdut.edu.cn

Received: Mar. 10th, 2020; accepted: Apr. 14th, 2020; published: Apr. 21st, 2020

Abstract

With the development of network technology, the information contained in various social networks is constantly increasing. But the increase in data information also means that the possibility of leakage of private information increases. Therefore, the protection of sensitive information should be considered when uploading and extracting user information. The (a, k)-anonymous algorithm derived from the k-anonymity algorithm is a classic privacy protection model, but with the complexity of social networks increasingly, the traditional (a, k)-anonymity algorithm is insufficient to meet the requirements of information hiding in social networks. In social networks, structural information and non-privacy attribute information of nodes may also be attacked, increasing the risk of privacy attribute disclosure. A privacy attribute (a, k)-anonymous algorithm based on node segmentation is proposed in this paper. In this algorithm, the nodes with privacy attribute value in the social network are segmented, so that the features of the nodes are divided into two nodes, and the possibility of the nodes being attacked is reduced. Experimental results demonstrate that this algorithm can protect the privacy data from partial attacks and ensure the availability of data.

Keywords

Privacy Property, Privacy Protection, Node Split, Anonymous, Social Networks

一种基于节点分割的隐私属性(a, k)-匿名算法

邓秀勤*，张翼飞，江志华，谭立辉

广东工业大学应用数学学院，广东 广州
Email: *dxq706@gdut.edu.cn

收稿日期：2020年3月10日；录用日期：2020年4月14日；发布日期：2020年4月21日

*通讯作者。

文章引用: 邓秀勤, 张翼飞, 江志华, 谭立辉. 一种基于节点分割的隐私属性(a, k)-匿名算法[J]. 数据挖掘, 2020, 10(2): 143-151. DOI: 10.12677/hjdm.2020.102015

摘　要

伴随着网络技术的发展，各类社交网络所包含的信息也在不断地增大。在数据信息增加的同时也意味着隐私信息泄露的可能性增大。因此在上传和提取用户信息的时候应该考虑到敏感信息的保护，在k-匿名算法的基础上衍生的(a, k)-匿名算法是经典的隐私保护模型，但是随着社交网络的复杂性不断增加，传统的(a, k)-匿名算法不足以满足社交网络中信息隐匿的要求。针对在社交网络中，节点的结构信息和非隐私属性信息等也可能会受到攻击，本文提出一种基于节点分割的(a, k)-匿名算法。该算法对社交网络中带有隐私属性值的节点进行分割，使得节点特征被分割到两个节点里，降低了节点被攻击识别的可能性。实验结果表明，该算法可以有效防御部分攻击造成的隐私属性泄露，同时保证数据保持一定的可用性。

关键词

隐私属性，隐私保护，节点分割，匿名，社交网络

1. 引言

随着计算机技术的不断更新换代，人们交流的方式和工具也是层出不穷。网络社交成为人们获取信息的重要途径，但是在获取信息的同时，与用户相关的敏感信息也存在泄露的可能。根据 Ferri [1]等人的研究，绝大多数的人都比较关注其隐私。2019 年，数据泄露的问题随着 Facebook 的丑闻，吸引了许多人的眼球，据《纽约时报》的新闻透露，Facebook 和许多公司有多年的数据交易关系，并为一些大公司提供了用户隐私访问规则以外的更多接口。这些事件表明，现在数据的价值利益已经十分巨大，许多公司为了利益枉顾安全，对于隐私信息发布的保护已经变得至关重要了。

传统的数据隐私保护方法自 Sweeney 提出 k-匿名法[2]后开始迅速发展起来，衍生出了 p-敏感 k-匿名[3]和(a, k)-匿名[4]等各种各样的隐私保护模型，文献[5]在 p-敏感 k-匿名模型的基础上提出针对敏感属性的个性化隐私保护算法，实现了对敏感属性进行个性化隐私保护的目的。上述方法都是面对关系型数据和属性数据的，针对具有图结构的数据，也开始采取 k-匿名的思想来进行匿名隐私保护[6]。Liu 和 Terzi [7]等人提出了基于度的匿名化方法，由此出现了许多新颖的图结构数据隐私匿名方法[8] [9] [10] [11] [12]。Casas-Roma [8] [9]等人通过优化边的选择策略来改进 k 度匿名，降低边的修改数目，在图的节点和边缘建立新的节点和边缘，以此来防止敏感信息的泄露。金叶[11]等人提出了一种改进的 k 度匿名保护方法，不依靠度序列来重构图，而是从节点本身出发来修改节点的度并同时修改图的结构，该方法在满足 k 度匿名隐私保护需求的同时，也对图结构进行了保护。

在上述诸多的匿名算法中，较少针对社交结构和属性分布之间的关联，而且很少将社交网络和数据表结合起来研究，所以这些算法很难抵御例如攻击者通过图的社交结构或通过属性联合分布特征对某用户隐私属性所进行的推测攻击，通过联立数据表和图进行的联合攻击。此外，许多匿名方案在保持数据可用性时权衡不足，存在过度匿名。同一种属性，在社交网络中取值可以是多样的，有些可能只有部分

取值是敏感的隐私。现有的大部分方案模型普遍默认隐私属性列的所有取值均为隐私属性，带有相同的敏感性，即对同一隐私属性列的所有取值都进行匿名化，显然这方便了算法设计，但也不合理，因为现实应用中每个人对某个值是不是隐私都有不同的判断。

为了应对社交网络属性隐私保护的需求和问题，本文在保持一定的数据可用性的情况下，提出了一种基于图节点分割和数据表(a, k)-匿名算法，该算法可以实现带有隐私属性的用户的身份匿名、属性匿名的隐私保护。

2. 基于节点分割的隐私属性匿名算法

原始的 k-匿名能抵御身份泄露，却难以抵御属性泄露，l-多样性能增加带隐私属性多样性[3]，但也难以抵御概率推测攻击，而且随着近年来在线社交媒体的迅速发展，许多社交软件应运而生，如微信、微博等，里面不仅包含了许多个人属性，还新增了各个用户之间的社交连接属性，产生了社交连接结构。不怀好意的攻击者也开始研究出各种针对社交结构的攻击手段。为了使攻击者难以通过链接攻击、同质性攻击、背景知识攻击、近似性攻击、邻域攻击等攻击方式推断出目标用户与匿名发布图节点之间的真实对应关系，本文在经典的(a, k)-匿名模型的基础上，新增对网络图社交结构的操作，并与数据匿名操作进行融合，提出了基于节点分割的(a, k)-匿名算法，该算法以属性－社交网络模型为基础，对所有拥有隐私属性值的用户节点进行分割，处理了带有隐私属性值的用户节点的子图结构，改变了部分社交连接和节点度数，成功防止了社交结构类的攻击。算法具体步骤描述如下：

步骤 1：预处理目标属性－社交网络及其转化的数据表

1) 确定目标数据表里需要用到的数据记录；2) 对目标数据表里的属性依据之前提到的常用分类，将属性分为标识符、准标识符、带隐私属性和非隐私属性等；3) 依照不同的现实应用需求设定出 a 和 k 两个值分别用于满足 a-非相关性约束和 k-匿名；4) 清理或修改目标数据表里不合规则的各条记录；5) 构建所有准标识符 QI 和隐私属性值的泛化层次。

步骤 2：获得需要匿名处理的节点的优先级队列

在匿名化处理之前，要统计出原始图或表中被确定的隐私属性值，然后对拥有这些值的节点定为处理目标，加入到队列里，用于确定节点分割的顺序。

处理顺序基于如下考虑：1) 用户节点包含的隐私属性值越多，应当更早处理；2) 其邻居包含的隐私属性值越多，应当更早处理。因为其需要匿名的属性值越多，或者与隐私属性值的关联越多，则越需要匿名处理；3) 节点度越高，优先级越高，因为高度数节点的邻域子图通常较大，分割后对总体的结构影响也较大，优先处理可以尽早确定图的总体结构；4) 若某节点对于前三个规则都相同，则让其插入到其所在的优先级里的任意位置。

通过序列里的节点个数，可以确定需要分割的次数，每一次分割将多一个新的节点出来，则可以确定新的节点个数，通过确定个数，可以联系(a, k)-匿名，将新节点的 QI 属性加进需要的 k 匿名组里面，减少泛化的程度，提高数据的可用性，还可以减低敏感属性出现的频率，降低 a-非相关性泛化的程度。

步骤 3：提取当前需要分割的节点的 1-邻域子图

节点邻域查询是指获取某节点的邻域子图及其属性所做的方式，通过常见的广度优先策略查询出输入图中每个用户的邻居集合，然后合并到输入图，且输入该用户与邻居的社交连接。因为处理节点的社交关系，所以暂时不需考虑节点的属性，于是可将无向图简化为 $G=\{V,E\}$，即只包含节点和节点间的连接边。节点的 1-邻域查询表示为 $LY_i(v)$，令节点的初始查询 $LY_0(v)$ 返回节点自身，即 $LY_0(v)=\{v\}$，则有：

$$LY_i(v) = LY_{i-1}(v) \cup R(V_i, E_i)$$

其中， $R(V_i, E_i)$ 表示第 i 次查询的添加子图，并且有

$$V_i = V_{i-1} \cup JH(V_{i-1})$$

$$E_i = E_{i-1} \cup \{e | e(V_{i-1}, JH(V_{i-1})) \cup e(JH(V_{i-1}), JH(V_{i-1}))\}$$

$i = 1$ 时， $LY_1(v) = Neighbor_1(v)$ ，表示该查询的结果为节点的邻居子图。

按照优先级序列顺序，取出当前节点，并按照上述的节点邻域查询进行邻域子图的提取，算法伪代码详见算法一。

算法一 提取节点邻域子图算法(getSv)

输入： 属性－社交网络 G，节点 v
输出： 节点邻接子图 Sv

```
1.  set List nodeList, List attrList, List edgeList; //分别新建用于存放节点、属性值、边的数组
2.  nodeList.add(v); //将 v 加进数组里
3.  for each attr in v.getAttrs//迭代节点 v 的属性值
4.      edgeList.add(edge(v,attr)); //将 v 和 v 所拥有的属性值的边加进数组
5.  end for;
6.  for each node in v.getNeighbors//迭代每一个邻居节点
7.      nodeList.add(node); //将邻居节点加进数组里
8.      edgeList.add(edge(v,node)); //将 v 和邻居节点的边加进数组
9.      for each attr in node.getAttr //迭代当前邻居节点的属性值
10.         attrList.add(attr); //将当前邻居节点的属性值加进数组
11.         edgeList.add(edge(node,attr) //将当前邻居节点和其属性值的边加进数组
12.     end for;
13. end for;
14. Sv←buildPic(nodeList,attrList,edgeList);
15. return Sv;
```

步骤 4： 子图内的节点分割处理

分割当前节点，构造两个新节点，按一定规则分别继承其拥有的邻居和属性，规则如下：

1) 处理准标识符列的属性分割：分割出的两个新节点完全继承原节点相同的准标识符属性值的属性值。

2) 处理带隐私属性的属性分割：将原节点所有的隐私属性值随机平均分配到两个节点中，没有继承到属性值的将该带隐私属性值置为空值，若不允许为空值则继承该隐私属性值对应的泛化树中距离最近的非隐私泛化值。

3) 处理非隐私属性的属性分割：将原节点的所有属性值随机平均分配到两个节点中，没有继承到属性值的将属性值置为空值，若不允许为空值则强制两个节点都继承该属性值。

4) 属性分割后，原节点的邻居节点按照和新节点的带隐私属性和非隐私属性的共同属性值的数量，选择数量多的新节点来建立社交连接，移除原节点的社交连接。若共同属性数量一样多，则按照共同邻居数量多少进行选择，若数量仍相同，则按继承属性值数量少的继承，仍相同，则随机继承。

在分割完当前节点后，将原节点从优先级序列中删除，更新匿名图，若序列里仍有待分割的节点，返回步骤 3。具体流程详见算法二。

步骤 5： 将匿名图转化的数据表(a, k)-匿名化

分割完所有需要分割的节点后，得到了最终的匿名发布图，此时可以转化为数据表进行处理。

先考虑 k-匿名，在等价组里，需要得到至少 k 个不可识别的准标识符值，则分割后的两个新节点完全继承原节点相同的准标识符属性的属性值，可以增加等价组所需的节点，降低 k-匿名的泛化次数。分

割完节点后，使用最为经典的 Datafly 算法[12]进行泛化处理。

步骤 6：将(a, k)-匿名化后的数据表所变更的属性修正到匿名图上，最后发布匿名图和匿名表。详见算法三和算法四。

算法二 节点分割算法(nodeAnatomy)

输入：代分割节点 v，v 对应属性－社交网络局部子图 Sv，隐私属性值集 P，属性集 J，隐私属性值集对应的泛化树集 Tree
输出：新的局部子图 SPv

```
//先分割属性值
1. set int num = 1;      //平均分配隐私属性值时用到的数字
2. set v1,v2;        //建立两个新的空节点属性组
3. for each attr in v        //迭代节点的所有属性
4.   if (attr.kind==J.QI)   //对属性值是否属于准标识符作判断
5.     v1,v2←add(attr)        //将相应属性分别加入 v1，v2
6.   else if (attr.kind==J.S)   //对属性值是否属于隐私属性作判断
7.   if (num==1)
8.      v1←add(attr)；num=2;
9.    if(J.S can't null)      //该属性不能为空值
10.v2←add(tree.attr)      //从该隐私属性值的泛化树中取出泛化后的非隐私属性值结果，并存入 v2 的属性值数组
11.      end
12.   else
13.v2←add(attr)；num=1;
14.    if(J.S can't null)     //该属性不能为空值
15.v1←add(tree.attr)      //从该隐私属性值的泛化树中取出泛化后的非隐私属性值结果，并存入 v1 的属性值数组
16.end
17.    end
18.   else if(attr.kind==J.IS)   //对属性值是否属于非隐私属性作判断
19.重复 7-17
20.   end          //此时 v1,v2 以获得所分割的节点
//  分割社交连接
21.   set   edgeListOfv1, edgeListOfv2//分别建立两个节点存放边的数组
22.   for each node in v.getNeighbors //迭代原节点的邻接节点
23.a,b←count(node.getAttrs)          //得到 v1 和 v2 分别于当前节点共同属性值数量
24.   if(a=b)      //如果共同属性值一样多，进行随机化操作
25.      random(a+x)，x!=0;
26.   end
27.   if(a>b)          //v1 的共同属性值多
28.      e1←add(v1,node)   //将当前节点和 v1 的边加入 e2
29.else
30.      e2←add(v2,node)
31. end
32.   end for
33. 分别建立 v1，v2 和当前节点的连接
34.   remove(v)    //移除原节点和其属性值及社交连接
35.   return SPv;
```

算法三 匿名发布图生成流程算法

输入：属性－社交网络 G，节点优先级队列 L，隐私属性值集 P，属性集 J，隐私属性值集对应的泛化树集 Tree
输出：匿名后的属性－社交网络 Gp

```
1.   for each node in L        //迭代队列 L 的每个节点，进行分割操作
2.      Sv←getSv(node);        //得到当前节点的子图
3.      Sv' ←nodeAnatomy(Sv,v,P,J,Tree); //将当前子图进行节点分割处理，得到处理后的新子图
4.      G←update(G,Sv',Sv);                  //更新属性-社交网络图
5.      end for
6.      Gp←G;                                  //得到经过节点分割的匿名图
7.      Return Gp;
```

算法四 匿名发布表生成流程算法

输入：匿名后的属性－社交网络 Gp，属性集 J，隐私属性值集对应的泛化树集 Tree
输出：匿名数据表 anonymityChart
1. **Chart←Gp.getChart** //匿名图生成待发布的数据表
2. **Chart' ←Datafly(Chart,Tree,J);** //对数据表使用 Datafly 算法进行 k-匿名处理
3. **anonymityChart← a-Anony(Chart');**//对数据表进行 a-非相关性约束
4. **return anonymityChart;** //返回(a, k)-匿名处理后的数据表

3. 算法复杂度分析

步骤 1 为初始化预处理工作，不计入算法时间复杂度；步骤 2 为生成优先级队列，需要对节点遍历一遍，确定分割的节点和顺序，时间复杂度为$O\left(n^2\right)$；步骤 3 是以广度优先策略查询节点邻居子图，该步骤中存储图的数据结构是邻接表，所以时间复杂度为$O(n)$；步骤 4 是属性-社交网络的节点分割，时间复杂度为$O\left(n^2\right)$；步骤 5 按照参考文献中的(a, k)-匿名算法执行，时间复杂度为$O\left(n^2\right)$，则该算法的总时间复杂度为$O\left(3n^2+n\right)$，即为$O\left(n^2\right)$。

4. 实验结果与分析

本实验采用 MovieLens 1M 的数据集的部分数据进行实验，数据集包含三个文件：用户数据 users.dat，电影数据 movies.dat 和评分数据 ratings.dat。其中 users.dat 的字段包含了标识符 UserID、准标识符 sex、准标识符 age、带隐私属性值职业 ID、准标识符 zipcode；movies.dat 的字段包含了 MovieID、Title、Genres。由于软件限制图节点不能过多，所以本文选取了 users.dat 文件里的 50 个用户节点、并将部分职业 ID 定义为隐私属性值，并且增加原数据集的社交连接 users-friends.dat。根据节点分割处理前后的原始图和匿名图的对比，分析出图结构变化的程度，从而得出图结构变化情况判断数据可用性。本文使用图的聚类系数和接近中心势来衡量图结构变化，变动小则数据可用性高。其中，聚类系数是衡量处理前后图结构的聚集特征变动，接近中心势是衡量图的节点接近趋势变化。

为了对比出隐私属性值的数量对图数据可用性的影响，将带隐私属性“职业 ID”定义 10%、20%、30%隐私属性值比例，分别分析匿名前后的图变化情况。图 1 是原始数据图，分别在隐私属性值占比 10%，20%，30%的时候进行节点分割算法处理，分割结果如图 2、图 3、图 4 所示。

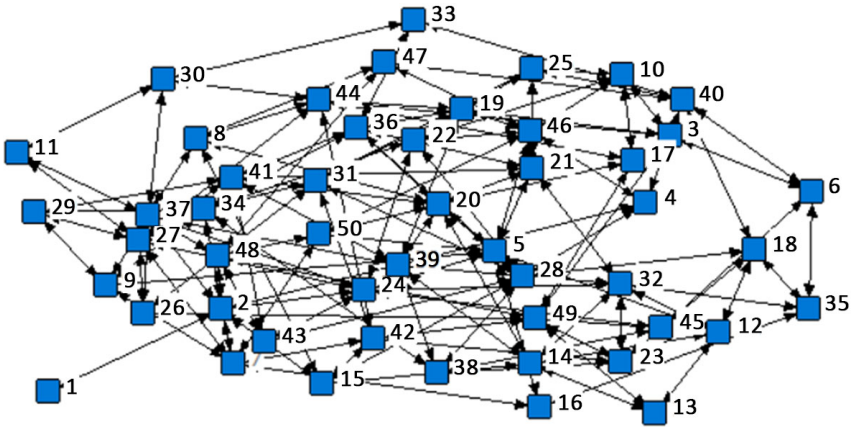

Figure 1. Original data map
图 1. 原始数据图

使用 Ucinet 软件得出聚类系数和接近中心势，最终得出的数据如图 5 所示(0 为未处理情况)：

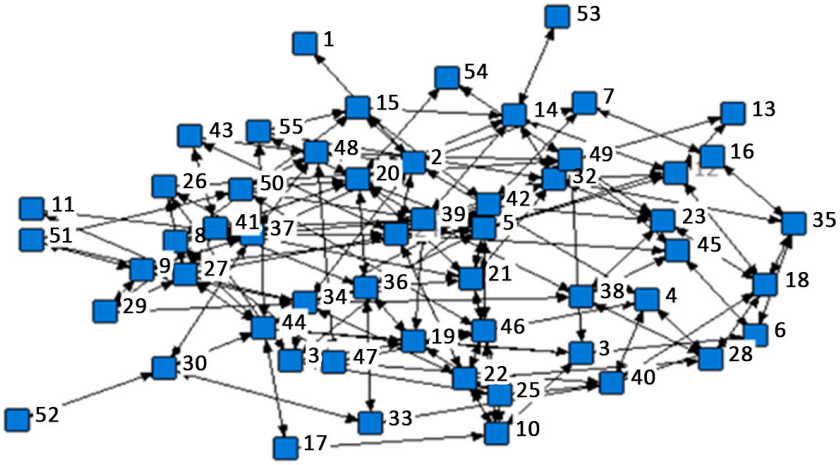

Figure 2. An anonymous processing graph with privacy value accounting for 10%
图 2. 隐私值占比 10%的匿名处理图

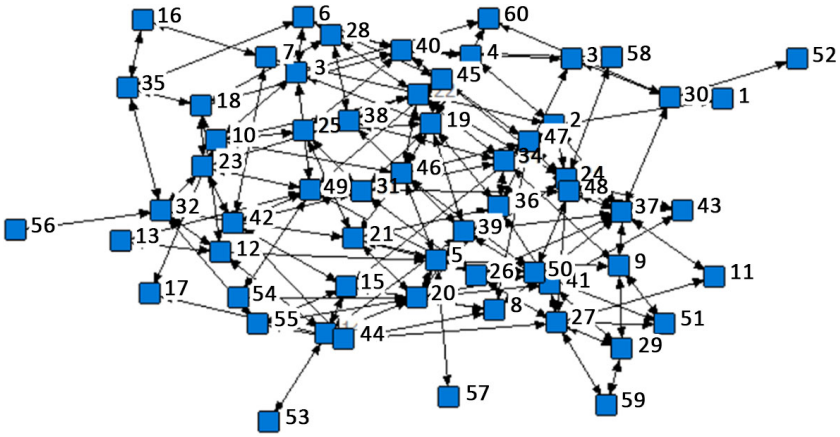

Figure 3. An anonymous processing graph with privacy value accounting for 20%
图 3. 隐私值占比 20%的匿名处理图

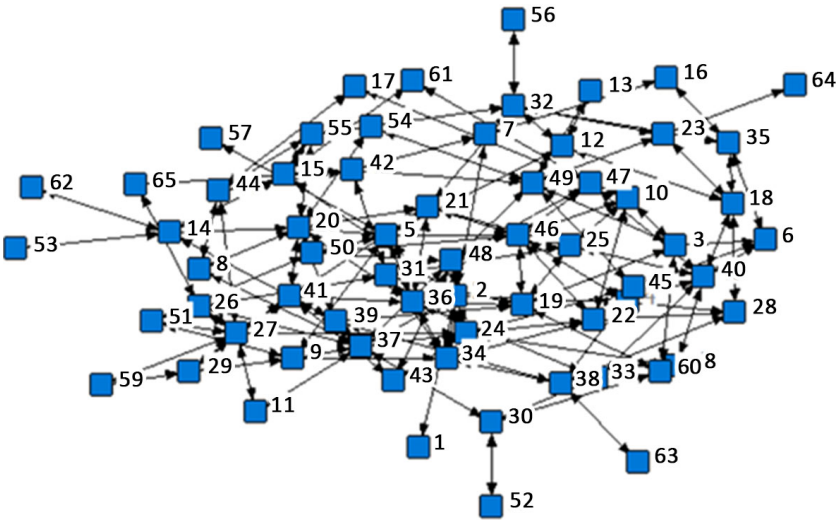

Figure 4. An anonymous processing graph with privacy value accounting for 30%
图 4. 隐私值占比 30%的匿名处理图

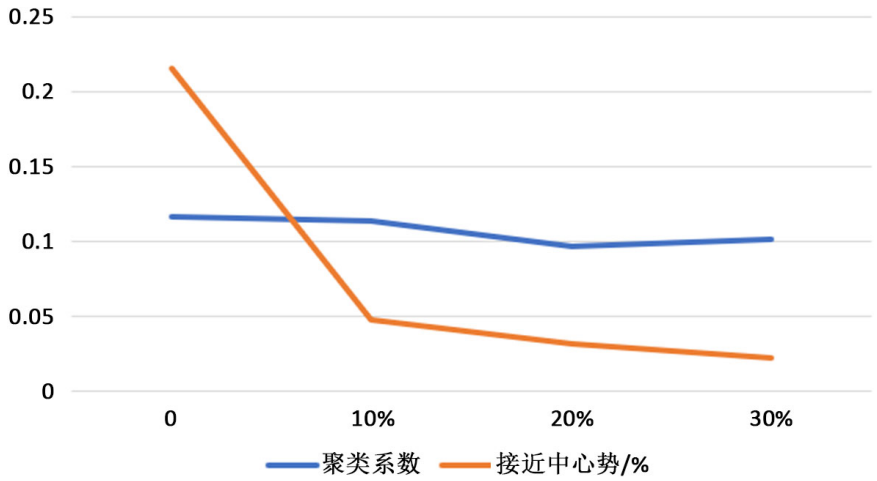

Figure 5. Clustering coefficient and statistics of near center potential before and after anonymity
图 5. 匿名前后的聚类系数和接近中心势统计值

通过图 5，可以看到匿名前后图的聚类系数只有少量降低，而且随着所选的隐私属性值的增加，即需要分割的节点增加，该系数的变化也不大，表明图的聚集结构并没有发生明显变化，扰动较小；而接近中心势匿名前后有一定降低，原因是由于节点分割降低了部分节点的度数，各节点的中心性也有所减少。但总体而言，该分割算法较好的保持了匿名前后图的结构，保持了数据的可用性。

即使在外部攻击者通过一些方式得知了用户的邻居或属性数量，并以此在发布图中进行查询，企图定位出节点，获得相关隐私，但是算法将图匿名处理后，能很好的拓展目标的可能范围，防止定位成功。这是因为算法分割了包含隐私属性值的用户节点，使得两个新节点的度数和子图结构与原节点有所差异，攻击者难以从度数或属性数量改变过的范围里定位出目标用户。可以有效抵御基于邻居或属性值数量的单独或组合攻击和基于邻居子图结构的攻击。

5. 结束语

本文根据(a, k)-匿名算法可以有效防止因隐私属性值分布不均导致泄露的优点以及现代的属性－社交网络结构图，针对传统模型无法应对社交结构攻击的缺陷，新增对网络图社交结构的操作，并与数据匿名操作进行融合，提出了一种基于节点分割的(a, k)-匿名算法。该算法对社交网络中带有隐私属性值的节点进行分割，使得节点特征被分割到两个节点里，降低了节点被攻击识别的可能性，改变了部分社交连接和节点度数，成功防止了社交结构类的攻击。保持了较好的数据可用性的同时，抵御了对节点度数或属性度数的重识别攻击、邻居子图查询攻击等新型攻击。

基金项目

广东工业大学教育教学改革项目“基于产出导向教育理念的离散数学课程教学改革研究与实践”(广工大教字[2017]101 号)，“以案例和实践为主导的模式识别教学改革”(广工大教字[2018]132 号)。

参考文献

[1] Ferri, F., Grifoni, P. and Guzzo, T. (2012) New Forms of Social and Professional Digital Relationships: The Case of Facebook. *Social Network Analysis and Mining*, **2**, 121-137. https://doi.org/10.1007/s13278-011-0038-4

[2] Sweeney, L. (2002) K-Anonymity: A Model for Protecting Privacy. *International Journal on Uncertainly, Fuzziness, and Knowledge Based Systems*, **10**, 557-570. https://doi.org/10.1142/S0218488502001648

[3] Ford, R., Truta, T.M. and Campan, A. (2009) P-Sensitive K-anonymity for social networks. In: Stahlbock, R., Crone, S.F., Lessmann, S., eds., *International Conference on Data Mining*, CSREA Press, Las Vegas, 403-409.

[4] 吴宏伟. 社会网络数据发布中的隐私匿名技术研究[D]: [硕士学位论文]. 黑龙江: 哈尔滨工程大学, 2013.

[5] 贾俊杰, 闫国蕾. 一种个性化(p, k)-匿名隐私保护算法[J]. 计算机工程, 2018, 44(1): 176-181.

[6] 姜火文, 占清华, 刘文娟, 马海英. 图数据发布隐私保护的聚类匿名方法[J]. 软件学报, 2017, 28(9): 2323-2333.

[7] Liu, K. and Terzi, E. (2008) Towards Identity Anonymization on Graphs. *Proceedings of ACM SIGMOD International Conference on Management of Data* (*SIGMOD* 2008), **2008**, 93-106. https://doi.org/10.1145/1376616.1376629

[8] Casas-Roma, J., Herrera-Joancomartí, J. and Torra, V. (2017) k-Degree Anonymity and Edge Selection: Improving Data Utility in Large Networks. *Knowledge and Information Systems*, **50**, 447-474. https://doi.org/10.1007/s10115-016-0947-7

[9] Casas-Roma, J., *et al*. (2019) k-Degree Anonymity on Directed Networks. *Knowledge and Information Systems*, **61**, 1743-1768. https://doi.org/10.1007/s10115-018-1251-5

[10] Kiabod, M., Dehkordi, M.N. and Barekatain, B. (2019) TSRAM: A Time-Saving k-Degree Anonymization Method in Social Network. *Expert System*, **125**, 378- 396. https://doi.org/10.1016/j.eswa.2019.01.059

[11] 金叶, 丁晓波, 龚国强, 吕科. 基于节点分类的 k 度匿名隐私保护方法[J/OL]. 计算机工程, 1-7. https://doi.org/10.19678/j.issn.1000-3428.0054407, 2020-03-26.

[12] Portela, J., García Villalba, L.J., Silva Trujillo, A.G., Sandoval Orozco, A.L. and Kim, T.-H. (2016) Estimation of Anonymous Email Network Characteristics through Statistical Disclosure Attacks. *Sensors*, **16**, 1832-1847. https://doi.org/10.3390/s16111832