# 抵制反匿名查询：基于H-index的社会网络k-匿名隐私保护

## 摘要

在本文中，我们提出了一种新的k-匿名方案来抵制社交网络上的匿名查询。使用这种方案，所有实体都受到k-匿名化保护，这意味着攻击者无法以高于1 / k的置信度重新识别目标。所提出的方案最小化了对原始网络的修改，并因此最大化公布的数据的效用保留，同时实现k-匿名隐私保护。在实际数据集上的大量实验证明了所提出的方案的有效性，其中k-匿名网络的效率用pagerank（网页排名），betweenness（中介中心性）和它们的Kolmogorov-Smirnov（K-S）测试的分布进行验证。

**关键字**

隐私保护；匿名化查询；H-index

## 介绍

如今，如何处理隐私风险在信息检索（IR）领域[1]是一个具有挑战性的问题，特别是对于社交媒体数据。近期研究驱动型的数据挖掘和查询已经发展到那些发送到第三方用户[2]的社交网络数据。在发布这些数据之前，通常会在隐私泄露的情况下删除社会实体的身份， 这种匿名方法可能对关系型数据有效，但对社交网络数据无效，在社交网络数据中，社交关系以图为模型，顶点表示单个社交实体，边指示它们之间的关系。因此，攻击者有可能通过查询图结构泄露个人隐私，即使删除了身份识别。 例如，86.9％-95.5％的Google+用户是可匿名化的，这表明基于结构的匿名化在实践中非常强大[3]。

因此，如果在发布社交网络数据时只删除实体的身份将仍然面临隐私风险[4]，邻居攻击的方法首先在参考文献[5]中提出，攻击者可以根据其1-邻局图的特征成功查询受害者的隐私，该邻居图是包括目标顶点，它的邻居顶点和它们之间的边。此外，王等人提出了一个更具体的假设，即攻击者知道目标的1-邻居图和邻居顶点的所有度，称为1 \*-邻居攻击[6]。最近，在[7]中提出了加权1 \*-邻居攻击，要求攻击者获取所有边上权重相关内容。这些关于边的权重的假设导致匿名化成本非常高。例如，当k = 25时，Facebook数据集上超过20％的边会在[7]的k-匿名化方法中进行修改。如果发布的社交网络过度修改，匿名网络的效用会降低甚至会完全失去。 此外，攻击者获得邻居度分布的知识往往更为实用（比如说h-index[8]），而不需要知道所有邻居的确切的度。

在本文中，我们确定了一种新的基于h-index的匿名化查询攻击模型，它在社会网络领域更加现实，而不需要完全了解精确的邻居度。为了应对这种查询攻击，我们提出了一种k-匿名隐私保护方案，该方案可以保护社交网络中所有实体的隐私，同时具有实现有效保护和高网络实用程序保留的显着优势。

本文的其余部分组织如下：第2节介绍问题的表述。 所提出的方案在第3节中介绍。实验结果在第4节中提供。最后，第5节总结了本文。

## 问题的表述

在启动去匿名查询之前，攻击者被假定拥有一些关于目标的背景知识。 在本文中，我们确定了一个新的反匿名查询模型，攻击者可以获得目标的h-index。h-index定义如下：

定义2.1. **H-index.**给定一个社交网络图，顶点的h-index 是

 （1）

其中是身份已被删除的社交网络中的实体集合；是实体之间的关系集合；是与实体相关的信息的集合；是满足的公式，， 返回u的度。

定义2.2.**匿名化查询.**给定一个匿名社交网络图，攻击者可以根据某些背景知识（如h-index）重新识别目标实体。这种攻击是一个反匿名查询。

虽然实体的身份已经从匿名网络中移除，但反匿名查询仍然能够破坏对隐私信息的保护。 在本文中，我们关注基于h-index的反匿名查询攻击，并提出了一个k-匿名方案来避免隐私泄漏。 K-匿名化定义为：

定义2.3**.K-匿名.**如果至少存在（k-1）个其特征（如h-index）相同的其他顶点，则称顶点为k-匿名。如果社交网络的所有顶点都是k-匿名的，则社交网络受k-匿名保护。

在k-匿名化社交网络中，基于h-index的反匿名化查询的结果至少包括k个顶点。 因此，攻击者无法以高于1 / k的可信度重新识别目标。为了实现k-匿名化，我们将问题表述为：

**问题的制定：**给定一个社交网络图，问题是获得k-匿名图，

，它满足（1） ；（2）对于 ，至少k-1个其他顶点满足 ；（3）最大化G'保存G的效用。

## k-匿名法的建议方案

在提出的方案中，顶点首先被分成不同的组，然后通过修改原始网络在每个组中实现k-匿名化。

### 3.1、合并分组的分类

根据第2节中对问题的定义，一个组中顶点的数量必须大于或等于给定的k。我们将桶作为容器引入分组顶点。 每个桶对应一个h-index，并且具有相同h-index的顶点被分配给相应的桶。

算法1中给出了k-匿名化过程。算法1的第1-3行是分桶的初始化过程。首先，将所有

顶点的h-index放入集合中。然后，根据h-index将的元素分配到不同的桶。 在第三行中，是桶的集合，hist是根据它们的h-index将顶点分配到 桶中的函数。桶按照h-index的升序排序。第6行到第15行展示了合并桶和将每个接近相同h-index的顶点迭代过程。在此过程中，是记录组中顶点数量的变量，是结束合并过程的条件。是经处理的桶的集合，并且其大小，即属于桶的顶点的数量不被允许改变为小于k的值。

|  |
| --- |
| 算法1：基于H-index的k-匿名化 |
| **Input：**  **Output：**  1 Initiate ;  2 ()  3 hist;sort;  4 ;;  5 ;  6 **foreach** **do**  7 ;  8 ;  9 **if**  **then**  10 ;  11 ;  12 ;  13 ;  14 **end**  15 **end**  16 **return** |

当一个组成立时，下一步就是统一组中的h-index。 如果组中有多个桶，则属于该组的顶点具有多个指数，因为每个桶对应一个h-index。 因此，每个组需要一个目标h-index，并且我们提出以下指标来确定目标h-index：

 （2）

其中和是将v的h-index改变为h的加边数和删除边数。然后，一组中的顶点的各种h-index将被修改，这种修改以通过逼近方法接近目标h-index。

### 3.2、统一组中的H-index

在本节中，我们将介绍一个组中h-index统一的详细方法（算法1的第10行）。如果桶的相应h-index与组的目标h-index不相同，则桶中的每个顶点增加或减少其h-index至目标h-index。无论是增加还是减小h-index都会导致度数修改，从而改变当前顶点及其邻居顶点的h-index。因此，决定如何修改图以实现k-匿名化非常重要。在不失一般性的情况下，我们在以下两种情况下将顶点u的h-index逼近目标h-index。

**3.2.1案例1：**. 如果你的h-index小于目标h-index，我们需要增加h-index以实现k匿名化。有两种方法可以增加顶点的h-index：首先，将u连接到更大度数的其他顶点；其次，增加u的邻居顶点的度数。

首先，我们从定义2.1中推导出两个h-index条件。一个是，在u的邻居中，至少个顶点的度数不小于，另一个是邻居中大于度数的顶点不能超过+1个。 否则，你的H指数不会是。 因此，我们可以提出如下建议：

建议：3.1. 你的h-index满足两个条件：

（3）

函数与公式（1）中定义的相同。它返回第一个参数的邻居的数量，那些邻居的度数不小于第二个参数给定值。

定理1. 为了将u的h-index增加到，如果直接将u直接连接到度数大于-2的其他顶点，则增加边的最少数量为：

 （4）

定理1显示了通过第一种方法增加u的h-index所需的最少边数。然而，并非所有的顶点都可以连接到u。将u连接到称为目标顶点的其他顶点需要满足三个条件：

1. 目标顶点的度数大于-2；
2. 该过程（算法1的第11行中的）桶仍然满足k大小的要求；
3. u和目标顶点之间不存在边。

如果没有满足这些条件的个顶点，则第二种方法将被用于通过增加u的邻居度的度数来贡献于u的h-index，其度数小于目标h-index。

定理2. 通过向度数小于的u的邻居添加边来增加顶点u的h-index为，增加的边的数量为：

 （5）

表 1：修改边的百分比

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *K* | 5 | 10 | 15 | 20 | 25 |
| Facebook | 0.13% | 0.38% | 0.53% | 0.87% | 0.99% |
| Wiki | 0.53% | 0.64% | 1.02% | 1.13% | 1.47% |

，其中是的子集。定理2显示了增加h-index的另一种方法。为了最小化修改，中具有较大度的顶点优先被选择。

**3.2.2案例2：**. 如果u的h-index大于目标h指数，我们需要将u的h-index减小到，这里有两种方法可以实现：删除u和它的那些度数不小于的邻居之间的边；或者减少度数不小于的u个邻居的度数。

定理3. 通过删除u连接到那些度不小于的边来减少u的h-index并且逼近，删除的边的数量是满足的：

 （6）

减少u的h-index的第二种方法是减少邻居顶点的度数，如下所示：

定理4. 通过减少度数不小于的u个邻居的度数来减少u的h-index为，删除的边数为：

 （7）

无论哪种情况都需要考虑是否可以添加/删除边，因为度的变化不仅可能导致当前顶点和目的顶点的h-index发生变化，而且还会导致目标顶点的邻居的h-index发生变化。如果更改导致任何处理后的桶的大小小于k，则不允许添加/删除边缘。在逼近度的过程中，对于这两种情况，主要采用修改当前顶点u的直接边的第一种方法。第二种方法只有在先前的修改仍不能达到目标h-index时才采用。

## 实验结果

在本节中，我们将显示边的变化和评估结果的效用。评估是在两个真实世界的数据集上进行的，其中Facebook网络数据有4039个顶点和88234个边，Wiki网络数据包含7115个顶点和103689个边。

### 4.1、边缘的修改

在表1中，记录Facebook和Wiki数据集中修改边的百分比。边缘的修改随着匿名强度k的增加而增加。越大的k意味着更强的保护，但也意味着更大数量的边需要修改。在此次的实验中，Facebook网络和Wiki网络的修改边缘的百分比最多只有0.99％和1.47％。 另外，我们通过进一步评测来估计修改的社交网络的效用。

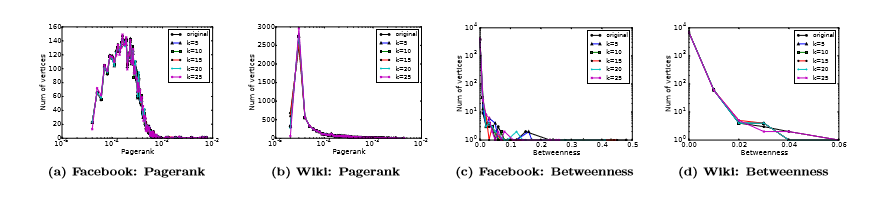


图 1：效用评估：比较Pagerank分布和Betweenness分布

表 2：K-S测试的P-值

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *K* |  | 5 | 10 | 15 | 20 | 25 |
| Pagerank | Facebook | 0.99160 | 0.99160 | 0.99993 | 0.99999 | 0.99160 |
| Wiki | 0.99995 | 0.99995 | 0.99995 | 0.99995 | 0.99995 |
| Betweenness | Facebook | 0.99999 | 0.99999 | 0.99999 | 0.99991 | 0.99952 |
| Wiki | 0.99820 | 0.97666 | 0.99024 | 0.96887 | 0.90549 |

**4.2、效用**

公开数据的效用可以评估匿名过程中信息的丢失和失真。丢失或失真的信息越多，公布的数据就越少。本文评估匿名网络的效用与pagerank（网页排名）和betweenness（中介中心性）的分布。

图1a-1b和图1c-1d显示了k匿名Facebook和Wiki网络对各种k的网页排名和中介性的分布。可以看出，对于从5到25的k，分布是非常相似的。我们首先分别使用长度0.00001和0.01来汇合网页排名和中介性。然后，这两个样本通过K-S检验进行验证。K-S检验的P值介于0和1之间，表明显著水平拒绝零假设。较大的p值意味着在相同分布后两个样本的可能性较高。在我们的实验中，如表2所示，在不同k水平上所有指标的p值都大于0.9，其中大多数都超过了0.99，这意味着就这些通过K-S测试的指标来说，所有修改都被认为是失真了原始网络中的一小部分。

## 结论

隐私保护是信息检索中的关键挑战。本文调查了在社交网络上防止匿名化查询的隐私保护问题，确定了一个非常实用的名为h-index查询的反匿名攻击。为了对付这样的查询，本文提出了一个有效的k-匿名保护方案。所提出的方案对所有实体实现k-匿名保护。实验结果表明，使用所提出的方案的匿名化社交网络需要非常小的边的修改，并且在相关关键评分分布方面保留相当大的效用。

## 致谢

这项工作得到了国家自然科学基金（No. 1646955），国家自然科学基金（No. 61532008）和湖北省国际合作项目（No. 2014BHE0017）的部分支持。

## 参考文献

1. H. Yang, I. Soboroﬀ. Privacy-preserving IR 2015: When information retrieval meets privacy and security. In Proc. SIGIR, pp. 1157-1158, 2015.
2. H. Hsieh, C. Li, R. Yan. I See You: Person-of-Interest Search in Social Networks. In Proc. SIGIR, pp. 839-842, 2015.
3. S. Ji, W. Li, M. Srivatsa, R. Beyah, Structural data deanonymization: theory and practice, IEEE/ACM Transactions on Networking, 24(6), pp. 3523-3536, 2016
4. H. Fu, A. Zhang, X. Xie. Eﬀective Social Graph Deanonymization Based on Graph Structure and Descriptive Information. ACM Trans. Intell. Syst. Technol., 6(4), pp. 49:1–49:29, 2015.
5. B. Zhou, J. Pei. Preserving Privacy in Social Networks Against Neighborhood Attacks. In Proc. IEEE International Conference on Data Engineering, pp. 506-515, 2008.
6. G. Wang, Q. Liu, F. Li, S. Yang, and J. Wu, Outsourcing Privacy Preserving Social Networks to a Cloud, In Proc. IEEE INFOCOM, pp. 2886-2894, 2013.
7. Q. Liu, G. Wang, F. Li, S. Yang, J. Wu. Preserving privacy with probabilistic indistinguishability in weighted Social networks, IEEE Transactions on Parallel and Distributed Systems, 28(5), pp. 1417-1492, 2017.
8. L. Lu, T. Zhou, Q. Zhang, H. Stanley. The h-index of a network node and its relation to degree and coreness, Nature Communications, 7:10168, DOI: 10.1038/ncomms10168, 2016.