Open access • Journal Article • DOI:10.1007/S10115-016-0947-7

k-Degree anonymity and edge selection: improving data utility in large networks — Source link

Jordi Casas-Roma, Jordi Herrera-Joancomartí, Vicenç Torra

Institutions: Open University of Catalonia, Autonomous University of Barcelona, University of Skövde

Published on: 01 Feb 2017 - Knowledge and Information Systems (Springer London)

Topics: k-anonymity, Centrality and Cluster analysis

Casas-Roma, J., Herrera-Joancomartí, J. & Torra, V. (2017). k-Degree Anonymity And Edge Selection: Improving Data Utility In Large Networks. Knowledge and Information Systems, 50(2), 447-474.

DOI

https://doi.org/10.1007/s10115-016-0947-7

Knowledge and Information Systems manuscript No.
(will be inserted by the editor)

k-Degree Anonymity And Edge Selection: Improving Data Utility In Large Networks

Jordi Casas-Roma · Jordi Herrera-Joancomartí · Vicenç Torra

Received: Oct 03, 2014 / Revised: Nov 02, 2015 / Accepted: Mar 19, 2016

Abstract The problem of anonymization in large networks and the utility of released data are considered in this paper. Although there are some anonymization methods for networks, most of them cannot be applied in large networks because of their complexity. In this paper we devise a simple and efficient algorithm for *k*-degree anonymity in large networks. Our algorithm constructs a *k*-degree anonymous network by the minimum number of edge modifications. We compare our algorithm with other well-known *k*-degree anonymous algorithms and demonstrate that information loss in real networks is lowered. Moreover, we consider the edge relevance in order to improve the data utility on anonymized networks. By considering the neighbourhood centrality score of each edge, we preserve the most important edges of the network, reducing the information loss and increasing the data utility. An evaluation of clustering processes are performed on our algorithm, proving that edge neighbourhood centrality increases data utility. Lastly, we apply our algorithm to different large real datasets and demonstrate their efficiency and practical utility.

Keywords Privacy · K-Anonymity · Social networks · Information loss · Data utility · Edge measures

Jordi Casas-Roma
Internet Interdisciplinary Institute (IN3)
Faculty of Computer Science, Multimedia and Telecommunications
Universitat Oberta de Catalunya (UOC), Barcelona, Spain.
E-mail: jcasasr@uoc.edu

Jordi Herrera-Joancomartí
Department of Information and Communications Engineering
Universitat Autònoma de Barcelona (UAB), Bellaterra, Spain.
E-mail: jherrera@deic.uab.cat

Vicenç Torra
School of Informatics
University of Skövde, Sweden
E-mail: vtorra@his.se

1 Introduction

In recent years, an explosive increase of network data has been made publicly available. Embedded within this data, there is private information about users who appear in it. Therefore, data owners must respect the privacy of users before releasing datasets to third parties. In this scenario, anonymization processes become an important concern. The study of Ferri et al. [19] reveals that though some user groups are less concerned about data owners sharing data about them, up to 90% of members in others groups disagree with this principle. Backstrom et al. [2] point out that the simple technique of anonymizing networks by removing the identities of the vertices before publishing the actual network does not always guarantee privacy. They show that there exist adversaries that can infer the identity of the vertices by solving a set of restricted graph isomorphism problems. Some approaches and methods have been imported from anonymization on relational data [16], but the peculiarities of anonymizing network data avoid these methods to work directly on graph-formatted data. In addition, divide-and-conquer methods do not apply to anonymization of network data due to the fact that registers are not separable, since removing or adding vertices and edges may affect other vertices and edges as well as the properties of the network [51].

Although some methods have been developed for graph anonymization, they are only applicable to small and medium networks of a few thousands of vertices and edges, at most. Anonymization in large networks is still an open problem. In addition, it is very important to minimize the information loss during the anonymization process in order to maximize the data utility. The process is lossless, and thus data utility is maximized, when the analysis performed on the anonymous data should produce the same results as the ones the original data would have led to.

In this paper we introduce an algorithm for anonymization in large networks based on the concept of k-degree anonymity. It works with simple, undirected and unlabelled networks. Because these networks have no attributes nor labels in the edges, information is only in the structure of the network itself and, due to this, the adversary can use this structural information to attack the privacy. k-Degree anonymity ensures the user's privacy when the attacker has degree-based knowledge about some target vertices. Moreover, we introduce the concept of edge relevance in order to minimize information loss. The quality of anonymized data can be improved by evaluating how to modify network structure, i.e., how to preserve the most important edges.

1.1 Our contributions

In this paper[1] we present an algorithm for privacy-preserving based on the k-anonymity concept. We also consider edge relevance in order to reduce information loss associated with the anonymization process, which leads to an increase of data utility. We offer the following results:

- We introduce a polynomial time k-degree anonymous algorithm based on univariate micro-aggregation for undirected and unlabelled graphs.
- We demonstrate that, modifying the network structure, edge relevance helps us reduce information loss and increase data utility.
- We conduct an empirical evaluation of our algorithm on several well-known networks. Additionally, we compare our algorithm with other well-known k-degree anonymous

[1] A preliminary, short version [9] of this paper appeared at ASONAM 2013.

algorithms and demonstrate that ours achieves lower information loss, and consequently, better data utility.

- We prove that we are able to reduce the information loss and increase the data utility by using the edge relevance measure to preserve important edges during the network modification process. We perform an analysis using well-known clustering algorithms and we also show that using edge neighbourhood centrality our algorithm leads the anonymization process to better clustering results, i.e., results closer to the ones performed on original data.
- Finally, we conduct an empirical evaluation of our algorithm on three large real networks, demonstrating that it achieves the desired *k*-anonymity level on large networks with thousands and millions of vertices and edges.

1.2 Notation

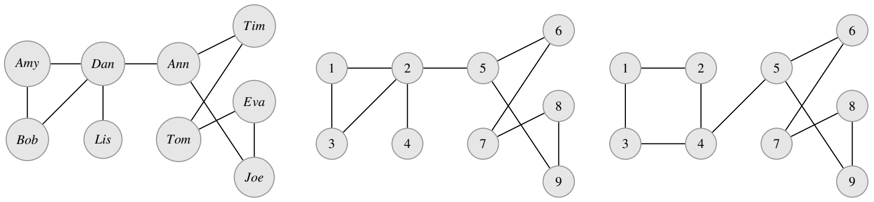
Let $G = (V,E)$ be a simple graph, where V is the set of vertices and E the set of edges. We use $v_i \in V$ to refer to vertex i and $\{v_i, v_j\}$ to refer to an undirected edge between vertices v_i and v_j. We define $n = |V|$ to denote the number of vertices and $m = |E|$ to denote the number of edges. We use d to define the degree sequence of G, where d is a vector of length n and d_i is the value of i-th element, that is, the degree of vertex $v_i \in V$. We refer to the ordered degree sequence as a monotonic non-decreasing sequence of the vertex degrees, that is $d_i \leq d_j \; \forall i < j$. We denote the average degree of the network as $\langle deg \rangle = \frac{2m}{n}$, the maximum degree as $max(deg)$ and the set of 1-neighbourhood of vertex v_i as $\Gamma(v_i)$.

1.3 Roadmap

This paper is organized as follows. The problem tackled in this paper is described in Section 2. In Section 3, we review the state of the art of anonymization in networks, specifically the *k*-anonymity based methods. Our algorithm for *k*-degree anonymity in large networks is introduced in Section 4. Then, experiments related to information loss, clustering assessment and large scale networks are introduced in Section 5. Specifically, we compare our algorithm with other two well-known *k*-degree based algorithms and discuss the results in 5.1. An evaluation of our algorithm on clustering-specific graph mining processes is presented in Section 5.2. Next, in Section 5.3, we conduct an empirical analysis for large networks. Last but not least, we present the conclusions and the future work in Section 6.

2 Problem definition

Currently, large amounts of data are being collected on social and other kinds of networks, which often contain personal and private information of users and individuals. Although basic processes are performed on data anonymization, such as removing names or other key identifiers, remaining information can still be sensitive, and useful for an attacker to re-identify users and individuals. To solve this problem, methods which introduce noise to the original data have been developed in order to hinder the subsequent processes of re-identification. A natural strategy for protecting sensitive information is to replace identifying attributes with synthetic identifiers. We refer to this procedure as simple or *naïve*



(a) G (b) G^* (c) $\widetilde{G}$

Fig. 1: Simple anonymization example, where G is the original graph, G^* is the naïve anonymous version and $\widetilde{G}$ is an anonymous version of the network.

anonymization. This common practice attempts to protect sensitive information by breaking the association between the real-world identity and the sensitive data.

Figure 1a shows a simple example of a social network, where each vertex represents an individual and each edge indicates the friendship relation between two individuals. Figure 1b presents the same graph after a naïve anonymization process, where vertex identifiers have been removed and the graph structure remains the same. Its degree sequence is $d_{G^*} = \{2,4,2,1,3,2,2,2,2\}$. Even though users' privacy is supposed to be secure, an attacker could still break the privacy and re-identify a user in the anonymous graph. For instance, if an attacker knows that Dan has four friends, then he is able to re-identify user Dan in anonymous graph, since he is the only user with four friends in graph G^*, i.e., unique vertex with degree equal to four. Consequently, user's privacy has been broken by the attacker.

Therefore, anonymization processes become an important concern in this scenario. These methods usually modify original graph structure to hinder re-identification processes. One of the most widely used technique to modify the graph structure is by adding or removing edges or vertices. Figure 1c depicts the same network after an anonymization process, where edge $\{v_2, v_3\}$ was removed and $\{v_3, v_4\}$ was created. The new degree sequence is $d_{\widetilde{G}} = \{2,3,2,2,3,2,2,2,2\}$. After this stage, there is no vertex with degree equal to four, and consequently no adversary with degree-based knowledge is able to re-identify user Dan in the anonymous network $\widetilde{G}$.

Nevertheless, the noise introduced by the anonymization steps may also affect the data, reducing its utility for subsequent data mining processes. Usually, the larger the data modification, the harder the re-identification but also the less data utility. Thus, it is necessary to preserve the integrity of the data to ensure that the data mining step is not altered by the anonymization step. The analysis performed on the obfuscated data should produce results as close as possible to the ones the original data would have led to. For instance, the average distance in G (and also G^*) is 2.278, while it is 2.416 in $\widetilde{G}$. Hence, this process has altered this measure, producing information loss and reducing data utility. A trade-off between data privacy and data utility must be reached; therefore, we developed our suggested algorithm in order to accomplish this aim. Our main goal is to develop a method which allows data publishing without breaking the privacy of users who appear in the network, and permits the analysis of published data minimizing the bias from the original data.

3 *k*-Anonymity on networks

Generally speaking, there are four families of methods to address network data privacy [35, 26,53]:

- The first family encompasses "graph modification" approaches. These methods first transform the data by edges or vertices modifications (adding and/or deleting) and then release the perturbed data. The data is thus made available for unconstrained analysis.
- The second family, called "uncertain graphs", was introduced by Boldi et al. [4]. This approach adds or removes edges partially by assigning a probability to each edge in anonymous network. Instead of creating or deleting edges, the set of all possible edges is considered and a probability is assigned to each edge. More recent works, such as [36], attempted to achieve a better trade-off between privacy and data utility by keeping some graph properties as close as possible to those in the original graph.
- The third family envelops "generalization" or "clustering-based" approaches [24,8]. These can be essentially regarded as grouping vertices and edges into partitions called super-vertices and super-edges. The details about individuals can be hidden properly, but the graph may be shrunk considerably after anonymization, which may not be desirable for analysing local structures. Some authors used the size of a partition to ensure node anonymity, where each super-vertex represents at least *k* vertices, and each super-edge represents all the edges between vertices in two super-vertices.
- Finally, the fourth family encloses "privacy-aware computation" methods [18,38,25, 26], which do not release data, but the output of an analysis computation. The released output is such that it is very difficult to infer from it any information about an individual input datum. Some of these methods refer to algorithms which ensure that individuals are protected under the definition of differential privacy [17], which imposes a guarantee on the data release mechanism rather than on the data itself. Differential privacy emphasizes that the structure of allowable queries on a statistical database must be designed in a way that a malicious attacker with the ability to query the database, but without direct access to the full database, cannot determine the unique characteristics of a specific individual. Hence, the goal is to provide statistical information about the data while preserving the privacy of users.

Among these families, "graph modification", "uncertain graphs" and "clustering-based" approaches first transform the data by different types of graph's modifications and then, release the perturbed data. The data is thus made available for unconstrained analysis. Nonetheless, it is important to underline that several graph metrics and algorithms must be re-defined to be applied on uncertain graphs, since almost all of them were designed to work with binary-edge graphs. Alternatively, "clustering-based" approaches do not enable local structure data analysis. A comparison between a clustering-based algorithm [8] and a graph modification algorithm [34] were conducted in [7]. The results show that the graph modification anonymous algorithm better preserves the communities on the released graphs, since the local structures are better preserved during anonymization process.

On the contrary, "privacy-aware computation" methods do not release data, but the output of an analysis computation. We do not consider these methods in our work, since they do not allow us to release the entire network, which provides the widest range of applications for data mining and knowledge extraction.

Thus, if we want to analyse the structure of the network, both the local and the global, we have to use a "graph modification" approach which enables us to deliver the entire network structure.

The first “graph modification” approaches emerged from the concept of randomization. Randomization methods are based on adding random noise in original data in order to hinder re-identification process. Hay et al. [23] suggested a method to anonymize unlabelled networks, called Random Perturbation, which is based on removing and then adding p edges at random. Ying and Wu [46] developed two algorithms designed to preserve spectral characteristics of the original networks: Spctr Add/Del and Spctr Switch. Ying et al. [47] presented a method, called Blockwise Random Add/Delete strategy or simply Rand Add/Del-B, which divides the network into blocks and implements modifications on the vertices at high risk of re-identification. Notice that the randomization approaches protect data against re-identification in a probabilistic manner.

Other approaches consider “graph modification” to meet desired privacy constraints. The notion of k-anonymity is included in this group. The k-anonymity model was introduced in [41] and [42] for privacy preservation on structured or relational data. The k-anonymity model indicates that an attacker cannot distinguish between different k records or individuals although he manages to find external knowledge related to the users who appear in the anonymous datasets. Therefore, an attacker cannot re-identify an individual with a probability greater than $\frac{1}{k}$.

The k-anonymity model can be applied using different adversaries’ knowledge when dealing with networks rather than relational data. A widely used option is to consider an adversary who knows the degree of some target vertices. If the attacker identifies a single vertex with the same degree in the anonymous graph, then he has re-identified this vertex. That is to say, $deg(v_i) \neq deg(v_j)\ \forall j \neq i$. This model is called k-degree anonymity [33] and these methods are based on modifying the graph structure (by edge modifications) to ensure that all vertices satisfy k-anonymity of their degree. In other words, the main objective is that all vertices have at least $k-1$ other vertices sharing the same degree. Furthermore, Liu and Terzi [33] developed a method based on integer linear programming in order to construct k-degree anonymous graph, where $V = \widetilde{V}$ and $E \cap \widetilde{E} \approx E$. Liu and Terzi’s work inspired many other authors who tried to improve solutions with different kinds of heuristics [34, 22]. Returning to our previous example in Figure 1, we can see that graph $\widetilde{G}$ is 2-degree anonymous, while the naïve anonymous version is only 1-degree anonymous. Hence, an attacker with degree-based knowledge is not able to re-identify a user in $\widetilde{G}$ with a probability greater than $\frac{1}{2}$.

Chester et al. [12, 14] also considered the k-degree anonymity problem, but they modified the network structure by adding new edges between fake and real vertices or between fake vertices. Under the constraint of minimum vertex additions, they show that on vertex-labelled networks, the problem is NP-complete. Following the same path, Bredereck et al. [5] studied the problem of making an undirected graph k-degree anonymous by adding vertices, together with incident edges. The authors explored three variants of vertex addition and studied their computational complexity. Chester et al. [13] introduced the concept of k-subset-degree anonymity, which produces an output network $\widetilde{G} = (V, E \cup \widetilde{E})$ such that $X \subseteq V$ is k-degree-anonymous and $|\widetilde{E}|$ is minimized. They also presented an algorithm to k-subset-degree anonymity which is based on the use of the degree constrained sub-graph satisfaction problem.

Instead of using a vertex degree, Zhou and Pei [51] considered an attacker with knowledge based on the 1-neighbourhood sub-graph of the target vertices. For a vertex $v_i \in V$, v_i is k-anonymous in G if there are at least $k-1$ other vertices $v_1, \dots, v_{k-1} \in V$ such that $\Gamma(v_i), \Gamma(v_1), \dots, \Gamma(v_{k-1})$ are isomorphic. Then, G is called k-neighbourhood anonymous if every vertex is k-anonymous considering the 1-neighbourhood. However, Tripathy and Panda [43] noted that their algorithm could not handle the situations in which an adversary has knowledge about vertices in the second or higher hops of a vertex, in addition to its immediate neighbours. To handle this problem, they proposed a modification of the algorithm to handle such situations. Zhou et al. [54] and Zhou and Pei [52] considered all structural information about a target vertex and introduced a new model called k-automorphism. Hay et al. [24] went a step further and presented a method, named k-candidate anonymity, which uses generic queries to model the adversary's knowledge. In this method, a vertex v_i is k-candidate anonymous with respect to question Q if there are at least $k-1$ other vertices in the network with the same answer. More recently, Cheng et al. [11] and Wu et al. [44] introduced the k-isomorphism and k-symmetry model, respectively.

When there is little diversity in the sensitive attributes inside an equivalence class, it is possible to obtain information from anonymized data although there are k indistinguishable individuals in each class. If the sensitive information is the same, it is possible to infer it unless the attacker does not know exactly which individual it is. ℓ-Diversity [37] and t-closeness [32] alleviate the problem of sensitive information disclosure. There are other privacy definitions of this flavour but they have all been criticized for being ad hoc [50].

4 The UMGA Algorithm

In this section, we introduce the Univariate Micro-aggregation for Graph Anonymization[2] (UMGA) algorithm, designed to achieve k-degree anonymity in large, undirected and unlabelled networks. The algorithm performs modifications in the original network $G=(V,E)$ only on edge set E. Hence, the vertex set V does not change during anonymization process.

Our algorithm is based on anonymizing the degree sequence. The degree sequence is an interesting tool since the concept of k-degree anonymity for a network can be directly mapped to its degree sequence, as Liu and Terzi showed in [33] and we recall in the following definitions:

Definition 1 A vector of integers V is k-anonymous if every distinct value $v_i \in V$ appears at least k times.

Definition 2 A network $G=(V,E)$ is k-degree anonymous if the degree sequence of G is k-anonymous.

Our algorithm is based on a two-step approach:

1. **Degree Sequence Anonymization.** We construct a k-degree anonymous sequence $\tilde{d} = \{\tilde{d_1},\dots,\tilde{d_n}\}$ from the degree sequence $d=\{d_1,\dots,d_n\}$ of the original network G using Definition 1. In addition, we use the function Δ to reduce the distance from the anonymized sequence to the original one, computed as:
$$\Delta = \sum_{i=1}^{n} |\tilde{d_i} - d_i| \tag{1}$$
The lower the value of Δ, the lower the information loss of the anonymized network.
2. **Graph modification.** We build a new network $\tilde{G}=(V,\tilde{E})$ where its degree sequence is equal to $\tilde{d}$ by using basic edge modification operations. These operations allow us to modify the network structure according to the anonymized degree sequence ($\tilde{d}$). As Definition 2 states, the anonymized network $\tilde{G}$ will be k-degree anonymous.

[2] Java implementation and source code available at: `https://jcasasr.wordpress.com`

4.1 Degree Sequence Anonymization

The objective of this first step is to anonymize the degree sequence of the original network (d), constructing a k-anonymous degree sequence ($\tilde{d}$).

Regarding the degree sequence, notice that:

- The number of elements is n, i.e., $d = \{d_1, \dots, d_n\}$.
- Each $d_i \in d$ must be an integer in the range $[0, n-1]$, since d_i is the degree of vertex v_i.
- $\sum_{i=1}^{n} d_i = 2m$, since each edge is counted twice in the degree sequence. Therefore, the sum of all degree values must be an even number.

Taking Definition 1 into account, we have to modify the values of d in order to create groups of k or more elements. Our method uses the optimal univariate micro-aggregation [21] to achieve the best group distribution and then it computes the values for each group that minimize the distance Δ from the original degree sequence.

We assume d to be an ordered degree sequence of the original network. Otherwise, we apply a permutation f to the sequence to reorder the elements. Let k be an integer such that $1 \leq k < n$ which is the k-degree anonymity value. Typically, k is much smaller than n. According to Hansen and Mukherjee [21], we first construct a new directed network $H_{k,n}$. Then, we compute the optimal partition in $H_{k,n}$, which is exactly the set of groups that corresponds to the arcs of the shortest path from vertex 0 to vertex n in $H_{k,n}$. We denote by g the optimal partition, where $g = \{g_1, \dots, g_p\}$ has $\frac{n}{k} \leq p \leq \frac{n}{2k-1}$ groups, and each $g_j \in g$ has between k and $2k-1$ items. According to what have been stated above, each $d_i \in d$ belongs to a specific group g_j.

Next, we compute the matrix of differences, $M_{p \times 2}$, using each group of the partition. The first column contains the sum of differences between each element of the group and the arithmetic mean of all degrees that belong to the group, using floor function to round the mean value. The second column is computed in the same way, but using de ceiling function instead. Conceptually, M contains the number of degrees in the first column, which we should decrease in this group to fulfil k-degree anonymity. These values are always zero or positive. The second column contains the number of degrees which we should increase in this group to reach k-degree anonymity. These values are always negative or zero. Formally, for $j = \{1, \dots, p\}$, each element m_{ji} is computed as:

$$m_{j1} = \sum_{d_i \in g_j} \left(d_i - \left\lfloor \langle g_j \rangle \right\rfloor \right) \tag{2}$$

$$m_{j2} = \sum_{d_i \in g_j} \left(d_i - \left\lceil \langle g_j \rangle \right\rceil \right) \tag{3}$$

where $\langle g_j \rangle$ is the average value of $d_i \in g_j$. A group with zero values on both columns should not apply any modification on its items because it is already k-anonymous.

Now we have to compute a solution, that is a p-vector where each element m_j is chosen from the two possible values m_{j1} or m_{j2}. The closer $|\sum_{j=1}^{p} m_j|$ to 0, the better the solution. This research is a complex operation with cost $\mathcal{O}(2^q)$ where q is the number of groups with $m_j \neq 0$. Because of the power-law in real networks, q is typically much smaller than p, but still too high for large networks. For that reason, we propose two methods in order to achieve the best combination on reasonable time:

4.1.1 The Exhaustive Method

The first approximation is based on exhaustive search. This method selects the values m_{j1} or m_{j2} for $j = \{1,\dots,p\}$ so that the minimum value of $|\sum_{j=1}^{p} m_j|$ is achieved. To implement this selection, all combinations are considered unless a solution is found with $\sum_{j=1}^{p} m_j = 0$.

4.1.2 The Greedy Method

Alternatively, a greedy approach can be used to reduce search complexity. Values for m_j are selected according to a probability distribution based on the size of values m_{j1} and m_{j2}. The lower the value is, the more probability to be chosen. More specifically,

$$p(m_j = m_{j1}) = 1 - \left(\frac{m_{j1}}{m_{j1} + m_{j2}}\right) \tag{4}$$

$$p(m_j = m_{j2}) = 1 - \left(\frac{m_{j2}}{m_{j1} + m_{j2}}\right) \tag{5}$$

The process is finished when a solution is found with $\sum_{j=1}^{p} m_j = 0$ or when we have a fixed number of iterations without any improvement in the function $\sum_{j=1}^{p} m_j$.

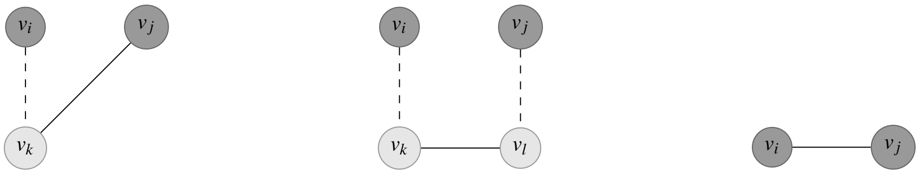
Example 1 Regarding our G network example, depicted in Figure 1, its ordered degree sequence is $d = \{1,2,2,2,2,2,2,3,4\}$. Optimal partition is $g = \{\{1,2,2\}_1, \{2,2\}_2, \{2,2\}_3, \{3,4\}_4\}$ and $p = 4$, i.e., ordered degree sequence is divided into four groups. The matrix of differences is as follows:

$$M_{4,2} = \begin{pmatrix} 2 & -1 \\ 0 & 0 \\ 0 & 0 \\ 1 & -1 \end{pmatrix}$$

Group $g_2 = \{2,2\}$ generates $(0,0)$ as the second row of M. For that reason, the items of the group g_2 do not need to modify their values. However, group $g_1 = \{1,2,2\}$ generates the row $(2,-1)$. Therefore, there are two possibilities to anonymize the group: decrease the values of the second and third items to 1 or increase the value of the first item to 2.

Finally, the space of all possible p-vector values is 2^q where $q = 2$. The exhaustive method explores all the space until it finds an optimal solution. The set of all possible p-vectors is $\{2,0,0,1\}$, $\{2,0,0,-1\}$, $\{-1,0,0,1\}$ and $\{-1,0,0,-1\}$; but the first and the second ones produce invalid solutions due to the fact that their sum of all degree values is not an even number. The third and the fourth solutions can be translated into $\{2,2,2,2,2,2,2,3,3\}$ and $\{2,2,2,2,2,2,2,4,4\}$, both of them are 2-degree anonymous valid sequences. However, the distance from the original sequence computed by Δ function is 0 for the first one and 2 for the second one. The former preserves the same number of edges while the latter increases the number of edges by one. The greedy method selects the value according to the following probability distribution, where the second and the third rows do not contain probability values since they do not need to modify their values according to M:

$$p(M_{4,2}) = \begin{pmatrix} 0.33 & 0.67 \\ - & - \\ - & - \\ 0.5 & 0.5 \end{pmatrix}$$



(a) Edge switch (b) Edge removal (c) Edge addition

Fig. 2: Basic operations for network modification with vertex invariability. Dashed lines represent deleted edges while solid lines are the added ones.

The third and the fourth solutions are the most probable solutions to be chosen. Apart from that, they have the same probability to be explored. □

Once exhaustive or greedy methods have finished, we have a k-degree anonymous sequence ($\tilde{d}$) which minimizes the distance computed by Equation 1. However, notice that in case we start the process with a non-ordered degree sequence, we should apply here the inverse transform f^{-1} to obtain the correct degree sequence.

4.2 Graph modification

In the second step, changes are made in the original network in order to convert it to a k-degree anonymous network. Considering Definition 2, a network is k-degree anonymous if its degree sequence is k-anonymous. Therefore, we have to change its degree sequence from the original one (d) to the anonymized one ($\widetilde{d}$) which we computed in the step above. We have to modify the edge set to meet the k-anonymous degree sequence.

In order to modify the edge set of a given network, we define three basic operations:

- **Edge switch** among three vertices can be defined as follows: if $v_i, v_j, v_k \in V$, $(v_i, v_k) \in E$ and $(v_j, v_k) \notin E$, we can delete (v_i, v_k) and create (v_j, v_k). Figure 2a shows this basic operation. Such modification can be translated in the degree sequence as $\widetilde{d_i} = d_i - 1$ and $\widetilde{d_j} = d_j + 1$, where $\widetilde{d}$ is the degree sequence after this basic operation. Notice that the number of edges in the network remains the same, v_i decreases its degree by one, v_j increases its degree by one and v_k keeps the same degree.
- We define **edge removal** as follows: we select four vertices $v_i, v_j, v_k, v_l \in V$ where $(v_i, v_k) \in E$, $(v_j, v_l) \in E$ and $(v_k, v_l) \notin E$. We delete edges (v_i, v_k) and (v_j, v_l), and create a new edge (v_k, v_l), as shown in Figure 2b. Note that $\widetilde{d_i} = d_i - 1$, $\widetilde{d_j} = d_j - 1$, $\widetilde{d_k} = d_k$ and $\widetilde{d_l} = d_l$.
- Finally, **edge addition** is defined as follows: we select two vertices $v_i, v_j \in V$ where $(v_i, v_j) \notin E$ and creates it. Note that $\widetilde{d_i} = d_i + 1$ and $\widetilde{d_j} = d_j + 1$. It is shown in Figure 2c.

The graph modification step starts by computing the vector $\delta = \widetilde{d} - d$, that indicates which vertices have to modify their degree. In fact, δ indicates precisely which vertices must increase or decrease their degree to fulfil the desired k-degree anonymity. The changes in original network are performed using the basic operations depicted in Figure 2. From δ, we create the list of vertices which must decrease their degree, $\delta^- = \{v_i : \delta_i < 0\}$ and the list of vertices which must increase their degree, $\delta^+ = \{v_i : \delta_i > 0\}$.

Let $\sigma(d)$ be the sum of each element in d, $\sigma(d) = \sum_{i=1}^{n} d_i$, and let $\sigma(\widetilde{d})$ be the sum of each element in $\widetilde{d}$, $\sigma(\widetilde{d}) = \sum_{i=1}^{n} \widetilde{d_i}$. If $\sigma(d) = \sigma(\widetilde{d})$, then there are the same number of edges in the original and in the anonymized networks. Therefore, we do not need to increase or decrease the total number of edges, but apply edge switch modifications.

If $\sigma(\widetilde{d}) < \sigma(d)$, we need to decrease $|\frac{\sigma(d)-\sigma(\widetilde{d})}{2}|$ edges from the network, as we have shown in Figure 2b. In order to decrease by 1 the total number of edges from the network, we choose $v_i, v_j \in \delta^-$ and find two other vertices v_k, v_l where $(v_i, v_k) \in E$ and $(v_j, v_l) \in E$. Then we delete these two edges and create a new one (v_k, v_l).

On the other hand, if $\sigma(\widetilde{d}) > \sigma(d)$, we need to increase $|\frac{\sigma(\widetilde{d})-\sigma(d)}{2}|$ edges to the network, as we have shown in Figure 2c. To increase by 1 the total number of edges, we select $v_i, v_j \in \delta^+$ where $(v_i, v_j) \notin E$ and create it.

Lastly, we have to modify the degree of some vertices, until $\sigma(\widetilde{d}) = \sigma(d) = 0$. This modification is done through edge switch, shown in Figure 2a. For each $v_i \in \delta^-$ and $v_j \in \delta^+$, we find another vertex v_k where $(v_i, v_k) \in E$. We delete this edge and create a new one (v_k, v_j).

We suggest two approaches to select the auxiliary edges needed for the graph modification process.

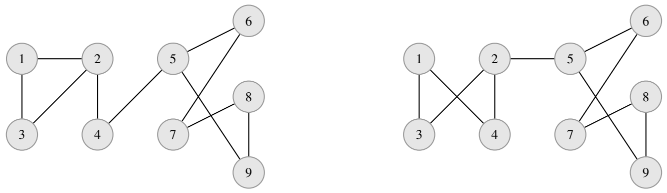
4.2.1 Random edge selection

We can simply select at random the auxiliary edges, that is, (v_i, v_k) and (v_j, v_l) for edge deletion and (v_i, v_k) for edge switch. Certainly, this is the fastest way to select the needed auxiliary edges. Nevertheless, some important edges can be removed or new bridge-like edges can be created by this random approach, affecting considerably the local or global structure of the resulting network.

4.2.2 Neighbourhood centrality edge selection

Alternatively, we can select the auxiliary edges by considering the relevance of each edge. Using this approach, we can remove or create new edges with low relevance, which leads the process to lower information loss. For example, if we remove edges with low value of edge betweenness, we will preserve information flow in the resulting network, since we preserve the most important edges using a measure related to information flow. Thus, identifying a relevant measure is important for reducing the information loss. In addition, we have to choose a measure with a low complexity, since we will compute it many times. For instance, the measure commented above is not a good choice because calculating edge betweenness on all edges in a network involves calculating the shortest paths between all pairs of vertices. This takes $\mathscr{O}(n^3)$ time with the Floyd-Warshall algorithm. In a sparse network, Johnson's algorithm may be more efficient, taking $\mathscr{O}(n^2 log(n) + nm)$ time. Hence, edge betweenness is not useful for large networks, since every time we remove an edge we must re-calculate all values of the edges.

We proposed a measure called edge *neighbourhood centrality* (NC) [10] for quantifying the edge relevance in large networks. In [10] we demonstrated that edge neighbourhood centrality identifies the most important edges in a network with low time complexity, and therefore, this measure is able to work in medium or large networks. The neighbourhood centrality of an edge $\{v_i, v_j\}$ is defined as the fraction of vertices which are neighbours of v_i or v_j, but not of v_i and v_j, simultaneously. The edge neighbourhood centrality is computed as follows:



(a) $\widetilde{G}$ after random edge selection

(b) $\widetilde{G}$ after neighbourhood centrality edge selection

Fig. 3: 2-degree anonymous networks updated using random and neighbourhood edge selection.

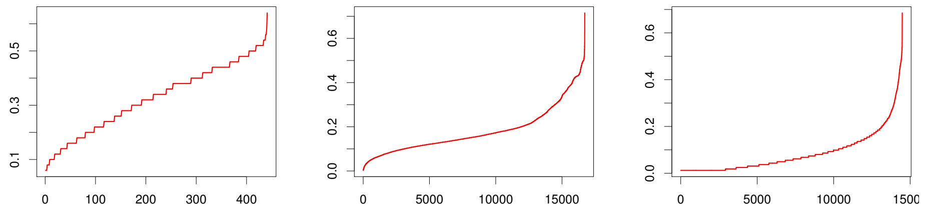
$$NC_{\{v_i,v_j\}} = \frac{|\Gamma(v_i) \cup \Gamma(v_j)| - |\Gamma(v_i) \cap \Gamma(v_j)|}{2max(deg)} \tag{6}$$

Note that this measure can be computed on $\mathcal{O}(m)$ using the adjacency matrix representation of the network. An edge with high score is a bridge-like for neighbourhood vertices. Consequently, choosing the edge with the lowest score on neighbourhood centrality we are preserving the connectivity in the anonymous network.

Example 2 Continuing with our G graph example described in Figure 1, the original degree sequence is $d = \{2,4,2,1,3,2,2,2,2\}$ and the 2-degree anonymous sequence obtained from the previous step is $\widetilde{d} = \{2,3,2,2,3,2,2,2,2\}$. Thus, vector $\delta = \{0,-1,0,1,0,0,0,0,0\}$ clearly indicates the vertices that must change their degree. Specifically, $\delta^- = \{v_2\}$ and $\delta^+ = \{v_4\}$, which means that v_2 has to decrease its degree by 1 while v_4 has to increase it by 1. The sum of the elements in the degree sequences is the same, i.e. $\sigma(d) = \sigma(\widetilde{d})$. Hence, there is no need to add or remove edges; edge switch is the only edge modification requirement to fulfil the anonymous degree sequence.

If we choose the random edge selection method, any edge $v_i : \{v_2, v_i\} \in E$ and $\{v_4, v_i\} \notin E$ is able to perform the structural changes that we need to apply to fulfil the anonymous degree sequence. The set of possible candidates is $v_i = \{v_1, v_3, v_5\}$. If we randomly select v_5 as the auxiliary vertex for edge switch, then edge $\{v_2, v_5\}$ is removed and $\{v_4, v_5\}$ is created. The resulting network is depicted in Figure 3a. Note that the bridge-like function of vertex v_2 was removed in the anonymous version of the network, and therefore the connectivity of the network changed considerably.

An alternative is to choose the neighbourhood edge centrality. In this case, some possible edge switches must be considered and then, the algorithm chooses the edge with the lowest neighbourhood centrality score. In our example, edges $\{v_2, v_5\}$, $\{v_1, v_2\}$ and $\{v_2, v_3\}$ are candidates with a neighbourhood score of 0.875, 0.5 and 0.5, respectively. As stated previously, the first one is a bridge-like connection between two dense components, while the second and third only affect the connectivity between involved vertices. The process selects the edge with the lowest neighbourhood centrality score, which are $\{v_1, v_2\}$ or $\{v_2, v_3\}$. Figure 3b shows the resulting network after removing edge $\{v_1, v_2\}$ and adding $\{v_1, v_4\}$. □



(a) Polbooks (b) Polblogs (c) GrQc

Fig. 4: Neighbourhood centrality values of tested networks.

4.3 Complexity

The first step, degree sequence anonymization, can be divided into two phases: the first one computes the optimal partition in a polynomial time, $\mathcal{O}(max(nlog(n), k^2n))$ as the authors [21] informed. The second one computes the p-vector from the matrix of differences $M_{p\times 2}$. The problem is NP for the exhaustive method ($\mathcal{O}(2^q)$ where q is the number of groups with $m_j \neq 0$). However, we can get a quasi-optimal solution in polynomial time ($\mathcal{O}(wq)$ where w is the number of fixed iterations) using the greedy approach. For instance, if we explore $w = log_2(2^q) = q$, then the complexity is polynomial $\mathcal{O}(q^2)$. Therefore, the problem of degree sequence anonymization is a P-problem and can be resolved in polynomial time for a quasi-optimal solution.

The second step, graph modification, can be implemented by random edge selection or neighbourhood centrality edge selection. As we have seen, each single edge operation can be computed in linear time; therefore, random edge selection can also be computed in linear time. Neighbourhood centrality edge selection involves evaluating all, or at least, a fraction of all possible combinations to preserve the most important edges. For each vertex v_i and v_j, we can generate several possible candidates:

- **Edge switch**: Given v_i and v_j, we select all $v_k \in \Gamma(v_i)$. We will usually find $\langle deg \rangle$ candidates for v_k or $max(deg)$ candidates when v_i is a hub or a vertex with a high degree. Hence, we have to check $\langle deg \rangle$ or $max(deg)$ possible combinations for each pair v_i and v_j. It is important to note that, because of the power-law in real networks, the vertices with the highest degree do not usually meet the k-anonymity and they need to be modified.
- **Edge removal**: Given v_i and v_j, we can select $v_k \in \Gamma(v_i)$ and $v_l \in \Gamma(v_j)$ as candidates. In average, there are $\langle deg \rangle$ possible vertices for v_k and v_l, and $max(deg)$ in the worst case. For every pair, we have to check whether edge $(v_k, v_l) \notin E$. Therefore, we have to check $\langle deg \rangle^2$ possible combinations in the average case, but it should be $max(deg)^2$ in the worst case, when v_i and v_j are hubs-like or vertices with a high degree value.
- **Edge addition**: Given v_i and v_j, we only need to check whether edge $(v_i, v_j) \notin E$. Only one possible combination for every pair of vertices is found here.

The complexity of edge evaluation process is NP. Accordingly, we need to use an heuristic to reduce the complexity of evaluating all possible combinations for each basic edge operation. Figure 4 shows the neighbourhood centrality values of tested networks (see Table 1 for network details). As we can see, neighbourhood centrality values are distributed like a power-law, where most edges have low score values and only few of them have a high score.

Figures 4b and 4c show clearly this phenomenon. Therefore, we focus on preserving the edges with high score of neighbourhood centrality. According to the distribution observed in real networks, we do not have to evaluate all possible combinations; we only need to evaluate a few of them to preserve the most important edges. Evaluating only the log_2 of the total possible combinations, we will find an edge with low NC score value with a very high probability, and we will reduce the complexity of edge selection. The complexity reduction is very important to deal with large networks, as we will demonstrate in Section 5.3. To sum up, the problem of graph modification is a P-problem for random edge selection and NP for neighbourhood centrality edge selection, but it can be resolved in polynomial time by computing the log_2 of all possible combinations.

5 Experimental results

In this section, we will discuss the results of our algorithm from three different perspectives. Firstly, we use the traditional approach focused on information loss using graph theoretic measures to compare our algorithm to other well-known k-degree anonymous algorithms. Secondly, we claim that testing our algorithm with graph-mining processes provides useful information. Thus, several clustering or community detection algorithms are utilized to assess the data utility of our anonymous networks. And thirdly, the scalability of our method is evaluated in large real networks.

Regarding the first perspective, several measures have been designed to evaluate the goodness of the anonymization methods. A natural way is to evaluate to what extent the analysis on anonymized data differs from the original data. Each measure focuses on a particular property of the data. We assume that if these metrics show little variation between original and anonymized data, then the subsequent data mining processes will also show little variation between the original and the anonymized one. Several measures have been used to compare the values obtained by the original and the anonymous datasets in order to quantify the noise introduced by the anonymization process, as illustrated in [23,24,33,46, 47,54]. When we quantify the information loss as described above, we talk about *generic information loss* measures. It is important to emphasize that these generic information loss measures only evaluate structural changes between original and anonymous data. That is, these measures do not evaluate the data mining processes on anonymous data, and as such, they are general or application-independent. We will evaluate our algorithm and two other well-know k-degree anonymous algorithms using generic information loss measures in Section 5.1.

However, the behaviour of anonymized data in the subsequent data mining processes may not coincide with the expected results. Since evaluating the distortion introduced in the graph is not enough, it is necessary to assess the noise introduced in the subsequent data mining processes. In our work we consider the case of clustering-specific processes. Thus, related to our second perspective, we will deal with specific information loss measures based on clustering processes in Section 5.2.

Last but not least, our third perspective in this analysis is focused on evaluating the scalability of our algorithm. In Section 5.3 we will analyse the behaviour of our algorithm when tackling large real networks.

5.1 Generic information loss analysis

In this section, we will compare the result of anonymizing several networks using our algorithm and other two well-known algorithms for *k*-degree anonymity. Specifically, we will use UMGA algorithm with random edge selection (UMGA-R) and neighbourhood centrality edge selection (UMGA-NC). We believe it is important to present a comparison to other well-known algorithms for *k*-degree anonymity. They are the following ones:

- ***k*-Degree-Anonymization** (*k*DA) algorithm by Liu and Terzi [33]. Unfortunately, the code of their algorithm is not publicly available. However, we use an experiment ran by Ying et al. [47] which compared the *k*-degree algorithm by Liu and Terzi to a random-based approach. We compare our algorithm to Liu and Terzi's by running a series of experiments identical to those of Ying et al. and comparing the performance of our algorithm to these published values.
- **VertexAddition** by Chester et al. [14]. VertexAddition algorithm is focused on *k*-degree anonymity by introducing fake vertices into the network and linking them to each other and to real vertices in order to achieve the desired *k*-degree anonymity value.

We apply all algorithms on the same data with the same parameters and compare the results in terms of information loss and data utility. It is important to note that the privacy level is the same for all algorithms, as we compare results with the same *k* value. For this reason, our comparison is about information loss between networks with the same privacy level achieved through different anonymization methods. Clearly, the lower the information loss, the better the algorithm.

5.1.1 Measures

In order to compare the algorithms, we use several structural and spectral measures. Information loss was defined by the discrepancy between the results obtained between the original and the anonymous data. The first structural measure is **average distance** ($\langle dist \rangle$), also known as average path length. It is defined as the average of the distances between each pair of vertices in the network. Information diffusion and spread is closely related to this measure. Formally, it is defined as:

$$\langle dist \rangle (G) = \frac{\sum_{i,j=1}^{n} d(v_i, v_j)}{\binom{n}{2}} \tag{7}$$

where $d(v_i, v_j)$ is the length of the shortest path from v_i to v_j, meaning the number of edges along the path.

Harmonic mean of the shortest distance (h) is an evaluation of connectivity, similar to the average distance. The inverse of the harmonic mean of the shortest distance is also known as the global efficiency. Formally:

$$h(G) = \left(\frac{1}{n(n-1)} \sum_{\substack{i,j=1 \\ i \neq j}}^{n} \frac{1}{d(v_i, v_j)} \right)^{-1} \tag{8}$$

Modularity (Q) indicates the goodness of the community structure. It is defined as the fraction of all edges that lie within communities minus the expected value of the same quantity in a network in which the vertices have the same degree, but edges are placed at random

Network	$\lvert V\rvert$	$\lvert E\rvert$	$\langle deg\rangle$	k
Polbooks	105	441	8.40	1
Polblogs	1,222	16,714	27.31	1
GrQc	5,242	14,484	5.53	1

Table 1: General properties of tested networks.

without regard to the communities. **Transitivity** (T) is one type of clustering coefficient, which measures and characterizes the presence of local loops near a vertex. It measures the percentage of paths of length 2 which are also triangles. Lastly, **sub-graph centrality** (SC) is used to quantify the centrality of vertex v_i based on the sub-graphs. Formally:

$$SC(G) = \frac{1}{n}\sum_{i=1}^{n} SC_i = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=0}^{\infty}\frac{P_i^k}{k!} \tag{9}$$

where P_i^k is the number of paths from v_i to v_i with length k.

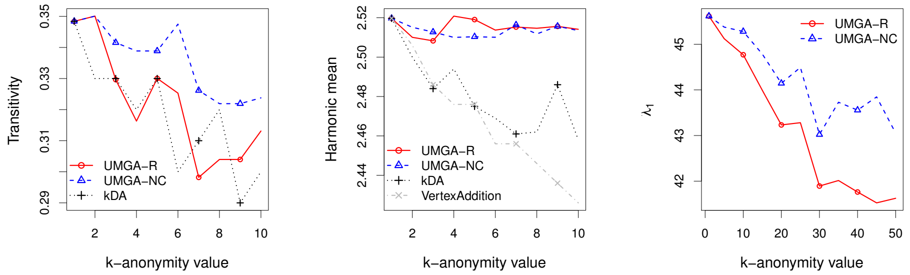
Moreover, two spectral measures which are closely related to many network characteristics [46] are used. **The largest eigenvalue of the adjacency matrix A** (λ_1) where λ_i are the eigenvalues of A and $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. The eigenvalues of A encode information about the cycles of a network as well as its diameter. **The second smallest eigenvalue of the Laplacian matrix L** (μ_2) where μ_i are the eigenvalues of L and $0 = \mu_1 \leq \mu_2 \leq \ldots \leq \mu_m \leq m$. The eigenvalues of L encode information about the tree structure of G. μ_2 is an important eigenvalue of the Laplacian matrix and can be used to show how good the communities separate, with smaller values corresponding to better community structures.

5.1.2 Tested networks

Table 1 shows a summary of the networks' main features including number of vertices, edges, average degree and default k-anonymity value. US politics book data (polbooks) [27] is a network of books about US politics published around the 2004 presidential election and sold by the online bookseller Amazon. Political blogosphere data (polblogs) [1] compiles the data on the links among US political blogs. Finally, GrQc collaboration network [29] is from the e-print arXiv and covers scientific collaborations between authors of papers submitted to General Relativity and Quantum Cosmology category.

5.1.3 Empirical results

Results are disclosed in Table 2. Each row indicates the scored value for the corresponding measure and algorithm; and each column corresponds to an experiment with a different k-anonymity value. For each dataset and algorithm, we vary k from 1 to 10 ($k = 1$ correspond to the original dataset) and compare the results obtained on λ_1, μ_2, $\langle dist\rangle$, h, Q, T and SC. The last column corresponds to the average error $\langle \mathscr{E}\rangle$. Each characteristic is reported from two to four times, corresponding to UMGA-R (indicated by R), UMGA-NC (indicated by NC), kDA and VertexAddition (indicated by VA). A bold row points out the best algorithm for each measure and network. Values of Liu and Terzi's algorithm (kDA) are taken from Ying et al. [47], and values of Chester et al. algorithm (VertexAddition) are taken from [14]. Unfortunately, values for all measures and algorithms are not available. Perfect performance in a row would be indicated by achieving exactly the same score as in the original network



(a) Polbooks network (b) Polblogs network (c) GRQC network

Fig. 5: Examples of information loss evolution during anonymization process using T, h and λ_1 measures in Polbooks, Polblogs and GRQC datasets.

(the $k = 1$ column). Although deviation is undesirable, it is inevitable due to the edge or vertex modification process. Complementary information is introduced in Figure 5, where we can see graphical details about algorithms' behaviour during anonymization process. These are the same values we reported in Table 2, but we believe that visual analysis can help us to understand where data come from.

The first tested network, Polbooks, is a small collaboration network. We present values for UMGA-R, UMGA-NC and *k*DA algorithms (except for average distance, which is not available from Ying et al. [47]). As shown in Table 2, UMGA-R and UMGA-NC algorithms introduce less noise in all datasets and measures, except for UMGA-R algorithm on T, where *k*DA achieves a slightly better result. For instance, we can deepen on the behaviour of transitivity in Figure 5a. Transitivity value for the original network ($k = 1$) is 0.348. As can be seen, values of UMGA-NC remain close to the original one over all anonymization range, while values are similar for UMGA-R and *k*DA. The results of the comparability are very encouraging. UMGA-NC outperforms on all measures, except on μ_2 where the values are close to UMGA-R and the average error ($\langle \mathscr{E} \rangle$) is almost equal. *k*DA produces much more information loss than UMGA-R and UMGA-NC, specifically on λ_1, μ_2, Q and SC where the average error is between two and four times greater than UMGA-NC.

Polblog is the second tested network. UMGA-R and UMGA-NC obtain similar values, which are the best values on all measures. The values of *k*DA are presented for all measures (except average distance), but the results are far from our algorithm. For instance, the average error is close to 0.26 for UMGA-R and UMGA-NC, while it outbursts to 1.75 for *k*DA on λ_1. As stated previously, eigenvalues encode information about cycles and the diameter of a network. Thus, keeping λ_1 values close to the original one implies better preserving path lengths and cycles in anonymous data. Similar results appear on h, Q, T and SC. Values for VertexAddition are presented for h, T and SC (other values are not available from Chester et al. [14]). Clearly, the noise introduced by our algorithm is much less than the noise introduced by theirs. For example, the average error on h is 0.039 for VertexAddition, while the same value is close to 0.005 for ours. As shown in Figure 5b, perturbation on h is lower when anonymization process is done by UMGA-R or UMGA-NC. That is, values of UMGA-R and UMGA-NC remain much closer to the original ones than values of *k*DA and VertexAddition. Lower information loss on h indicates that connectivity and path lengths are less affected in anonymous data, i.e. more similar to the original ones. The same behaviour is observed in other measures. UMGA-R and UMGA-NC have obtained an average error

Polbooks		k=1	2	3	4	5	6	7	8	9	10	$\langle\mathscr{E}\rangle$
λ_1	R	11.93	12.11	12.00	12.17	11.82	11.84	12.18	12.25	12.25	11.89	0.163
	NC		**12.09**	**11.97**	**11.85**	**11.85**	**11.95**	**12.09**	**12.08**	**12.08**	**11.86**	**0.090**
	kDA		12.00	12.05	12.11	12.22	12.30	12.31	12.64	12.72	12.85	0.383
μ_2	**R**	0.324	**0.359**	**0.428**	**0.497**	**0.330**	**0.398**	**0.653**	**0.593**	**0.593**	**0.495**	**0.143**
	NC		0.360	0.451	0.453	0.453	0.383	0.599	0.524	0.524	0.640	0.147
	kDA		0.430	0.450	0.600	0.600	0.790	0.630	0.650	0.970	0.880	0.312
$\langle dist\rangle$	R	3.079	2.928	2.826	2.770	3.029	2.861	2.647	2.694	2.694	2.795	0.247
	NC		**2.987**	**2.883**	**2.896**	**2.896**	**2.988**	**2.765**	**2.856**	**2.856**	**2.762**	**0.182**
h	R	2.450	2.392	2.343	2.314	2.428	2.356	2.252	2.276	2.276	2.326	0.109
	NC		**2.416**	**2.371**	**2.379**	**2.379**	**2.418**	**2.312**	**2.350**	**2.350**	**2.312**	**0.077**
	kDA		2.350	2.320	2.280	2.280	2.230	2.270	2.260	2.200	2.190	0.167
Q	R	0.402	0.400	0.396	0.388	0.398	0.389	0.377	0.379	0.379	0.394	0.012
	NC		**0.400**	**0.393**	**0.396**	**0.396**	**0.401**	**0.386**	**0.386**	**0.386**	**0.385**	**0.009**
	kDA		0.390	0.390	0.380	0.380	0.360	0.370	0.370	0.340	0.350	0.027
T	R	0.348	0.350	0.330	0.316	0.330	0.325	0.298	0.304	0.304	0.313	0.027
	NC		**0.350**	**0.342**	**0.339**	**0.339**	**0.347**	**0.326**	**0.322**	**0.322**	**0.324**	**0.013**
	kDA		0.330	0.330	0.320	0.330	0.300	0.310	0.320	0.290	0.300	0.023
SC	R	2.524	2.779	2.189	2.372	2.145	1.933	2.132	2.336	2.336	1.976	0.303
	NC		**2.774**	**2.358**	**2.224**	**2.224**	**2.338**	**2.363**	**2.389**	**2.389**	**2.110**	**0.204**
($\times10^3$)	kDA		2.480	2.560	2.530	2.760	2.440	2.680	3.600	3.580	4.120	0.431
Polblogs		k=1	2	3	4	5	6	7	8	9	10	$\langle\mathscr{E}\rangle$
λ_1	R	74.08	73.88	73.78	73.95	73.93	73.67	73.70	73.82	73.74	73.63	0.260
	NC		**73.93**	**73.81**	**73.92**	**73.95**	**73.74**	**73.80**	**73.75**	**73.63**	**73.61**	**0.256**
	kDA		74.89	74.50	75.16	75.10	76.32	75.82	76.67	77.42	78.42	1.758
μ_2	R	0.168	0.168	0.168	0.168	0.168	0.168	0.168	0.168	0.168	0.168	0.000
	NC		0.168	0.168	0.168	0.168	0.168	0.168	0.168	0.168	0.168	0.000
	kDA		0.168	0.168	0.168	0.168	0.168	0.168	0.168	0.168	0.168	0.000
$\langle dist\rangle$	**R**	2.738	**2.724**	**2.721**	**2.739**	**2.737**	**2.730**	**2.731**	**2.730**	**2.732**	**2.728**	**0.007**
	NC		2.733	2.729	2.725	2.724	2.724	2.732	2.726	2.731	2.727	0.009
h	**R**	2.506	**2.496**	**2.494**	**2.507**	**2.505**	**2.500**	**2.501**	**2.501**	**2.502**	**2.500**	**0.005**
	NC		2.501	2.499	2.496	2.496	2.496	2.502	2.498	2.502	2.499	0.006
	kDA		2.500	2.484	2.494	2.475	2.469	2.461	2.462	2.486	2.458	0.026
	VA		2.506	2.486	2.476	2.476	2.456	2.456	2.446	2.436	2.426	0.039
Q	**R**	0.405	**0.403**	**0.403**	**0.405**	**0.405**	**0.403**	**0.403**	**0.403**	**0.403**	**0.402**	**0.001**
	NC		0.404	0.403	0.403	0.403	0.403	0.403	0.402	0.403	0.402	0.002
	kDA		0.402	0.401	0.401	0.396	0.394	0.395	0.389	0.387	0.385	0.010
T	R	0.226	0.224	0.223	0.225	0.224	0.223	0.223	0.224	0.223	0.223	0.002
	NC		**0.224**	**0.224**	**0.224**	**0.224**	**0.223**	**0.225**	**0.224**	**0.223**	**0.224**	**0.001**
	kDA		0.225	0.223	0.224	0.221	0.222	0.220	0.219	0.221	0.221	0.004
	VA		0.219	0.215	0.207	0.205	0.200	0.226	0.190	0.185	0.183	0.020
SC	R	1.218	1.003	0.905	1.069	1.054	0.806	0.832	0.939	0.872	0.782	0.270
	NC		**1.052**	**0.932**	**1.040**	**1.068**	**0.871**	**0.921**	**0.875**	**0.776**	**0.765**	**0.266**
	kDA		2.730	1.870	3.610	3.400	1.450	6.940	6.250	4.460	4.040	2.386
($\times10^{29}$)	VA		1.300	1.410	2.160	2.880	2.660	5.550	5.370	11.000	8.250	2.969
GrQc		k=1	5	10	15	20	25	30	35	40	50	$\langle\mathscr{E}\rangle$
λ_1	R	45.61	45.12	44.77	43.99	43.23	43.28	41.89	42.01	41.76	41.62	2.450
	NC		**45.37**	**45.28**	**44.78**	**44.14**	**44.49**	**43.02**	**43.72**	**43.55**	**43.05**	**1.353**
μ_2	R	-1.26	-3.98	-2.37	-4.09	-3.33	-2.67	-3.55	-2.00	-1.91	-4.40	1.966
($\times10^{-14}$)	**NC**		**-3.06**	**-2.71**	**-1.95**	**-2.01**	**-4.34**	**-1.57**	**-2.35**	**-2.76**	**-2.48**	**1.300**
$\langle dist\rangle$	R	6.049	6.002	5.942	5.918	5.862	5.861	5.864	5.892	5.816	5.915	0.150
	NC		**6.026**	**6.009**	**5.955**	**5.948**	**5.922**	**5.904**	**5.935**	**5.876**	**5.897**	**0.102**
h	R	8.863	8.784	8.696	8.650	8.568	8.571	8.571	8.607	8.597	8.639	0.229
	NC		**8.821**	**8.794**	**8.718**	**8.702**	**8.664**	**8.641**	**8.682**	**8.589**	**8.626**	**0.162**
T	R	0.630	0.622	0.612	0.603	0.588	0.585	0.579	0.571	0.560	0.548	0.044
	NC		**0.625**	**0.617**	**0.611**	**0.588**	**0.595**	**0.589**	**0.589**	**0.578**	**0.584**	**0.033**
SC	R	1.235	0.754	0.530	0.244	0.114	0.120	0.030	0.034	0.027	0.023	0.951
($\times10^{16}$)	**NC**		**0.971**	**0.882**	**0.538**	**0.283**	**0.402**	**0.093**	**0.187**	**0.158**	**0.096**	**0.776**

Table 2: Results for UMGA-R (R), UMGA-NC (NC), *k*DA [33] and VertexAddition (VA) [14] algorithms. Bold rows indicate the algorithm that achieves the best results (i.e., the lowest information loss) for each measure. The values for *k*DA are taken from Ying et al. [47] and the ones for VertexAddition are taken from [14].

smaller than *k*DA and VertexAddition on all measures, showing that the information loss is reduced and the data utility is clearly improved.

Finally, GrQc is larger than other networks (in terms of the number of vertices), so results for this network are evaluated on $k \in [1,50]$. In addition, Q is not evaluated since this network does not have community labels. Unfortunately, there is no data for *k*DA or VertexAddition algorithms for this network. Thus, we cannot compare our algorithm versus theirs, but we include this network in order to evaluate our two approaches with a larger network. On GrQc the UMGA-NC achieves the best results on all metrics, reducing considerably the information loss and raising data utility, as can be seen in Figure 5c.

5.2 Clustering-specific information loss analysis

In this section we want to compare our random-based (UMGA-R) and neighbourhood centrality-based (UMGA-NC) edge selection approaches on graph-mining processes. We define the specific information loss measures as a task-specific measure for quantifying the data utility and the information loss associated to a data publishing process. We focus on clustering-specific processes, since it is an important application for social and healthcare networks. We want to analyse the utility of the perturbed data by evaluating it on different clustering processes. Like generic measures, we compare the results obtained both by the original and the perturbed data in order to quantify the level of noise introduced in the perturbed data. This measure is specific and application-dependent, but it is necessary to test the perturbed data in graph-mining processes.

5.2.1 Clustering evaluation framework

We ran 4 graph clustering algorithms to evaluate our anonymization algorithm using the implementations from the `igraph` library. All of them are unsupervised algorithms based on different concepts and developed for different applications and scopes. An extended revision and comparison of them can be found in [28,49]. The selected clustering algorithms are: (1) *Fastgreedy* [15], a hierarchical agglomeration algorithm for detecting community structure based on modularity optimization; (2) *Walktrap* [39] intends to find densely connected sub-graphs, i.e. communities, in a graph via random walks; (3) *Infomap* [40] optimizes the map equation, which exploits the information-theoretic duality between the problem of compressing data and the problem of detecting significant structures in the graph; and (4) *Multilevel* [3], a multi-step technique based on a local optimization of Newman-Girvan modularity in the neighbourhood of each node. Although some algorithms permit overlapping among different clusters, we did not allow it in our experiments by setting the corresponding parameter to zero, mainly for ease of evaluation.

We consider the following approach to measure the clustering assessment for a particular anonymization and clustering method: (1) apply the anonymization process p to the original data G and obtain $\widetilde{G}$; (2) apply a particular clustering method c to G and obtain clusters $c(G)$ and apply the same method to $\widetilde{G}$ to obtain $c(\widetilde{G})$; and (3) compare the clusters $c(G)$ to $c(\widetilde{G})$, as shown in Figure 6. In relation to information loss, it is clear that the more similar $c(\widetilde{G})$ is to $c(G)$, the less information loss. Thus, clustering-specific information loss measures should evaluate the divergence between both sets of clusters $c(G)$ and $c(\widetilde{G})$. Ideally, results should be the same. That is, the same number of sets with the same elements in each set. In this case, we can say that the anonymization process has not affected the clustering process. When the sets do not match, we should be able to calculate a measure of divergence.

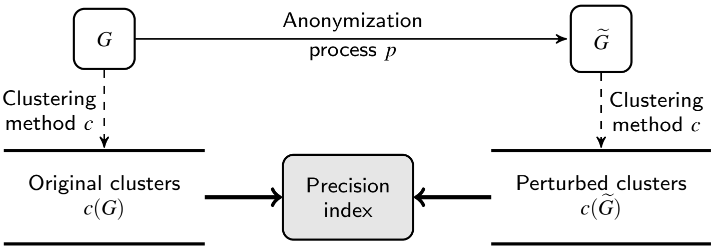


Fig. 6: Framework for evaluating the clustering-specific information loss measure.

Network	$\lvert V \rvert$	$\lvert E \rvert$	$\langle deg \rangle$	k
Zachary's karate club	34	78	4.588	1
American college football	115	613	10.661	1
Erdős	433	1,314	6.069	1
Enron	36,692	183,831	10.020	1

Table 3: General properties of tested networks.

For this purpose, we use the *precision index* [6]. Assuming we know the true communities of a graph, the precision index can be directly used to evaluate the similarity between two cluster assignments. Given a graph of n vertices and q true communities, we assign the same labels $l_{tc}(\cdot)$ to vertices which belong to the same community. In our case, the true communities are the ones assigned in the original dataset (i.e. $c(G)$), since we want to obtain communities as close as possible to the ones we would get on non-anonymized data – we are not interested in the ground truth communities. Assuming the anonymous graph has been divided into clusters (i.e. $c(\widetilde{G})$), we examine all the vertices for each cluster and assign them the most frequent true label in that cluster as the predicted one $l_{pc}(\cdot)$. Then, the precision index can be defined as follows:

$$precision\_index(G, \widetilde{G}) = \frac{1}{n} \sum_{v \in G} \nVdash_{l_{tc}(v) = l_{pc}(v)} \tag{10}$$

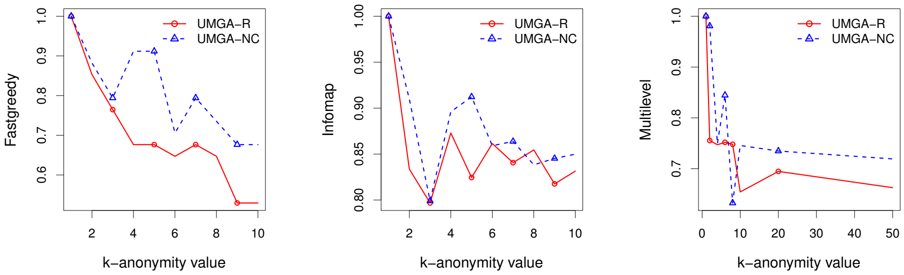
where $\nVdash$ is the indicator function such that $\nVdash_{x=y}$ equals 1 if $x = y$ and 0 otherwise. Note that the precision index is a value in range [0,1], which takes value 0 when there is no overlap between the sets, and value 1 when the overlap between the sets is complete.

5.2.2 Tested networks

Four different network are evaluated in this section. They present different sizes and structures, as shown in Table 3. Zachary's karate club [48] is a small social graph widely used in clustering and community detection. American college football [20] is a graph of American football games among Division IA colleges during regular season Fall 2000. Erdős network presents a list of mathematician Paul Erdős' co-authors and their respective co-authors. Lastly, Enron email communication network [31] covers all the email communication within a dataset of around half million emails.

Network	Method	IM	ML	FG	WT
Karate	UMGA-R	0.205	0.238	0.300	**0.232**
	UMGA-NC	**0.141**	**0.226**	**0.191**	0.282
Football	UMGA-R	0.086	0.052	0.157	**0.035**
	UMGA-NC	0.086	**0.003**	**0.053**	0.039
Erdős	UMGA-R	0.147	0.291	0.191	0.069
	UMGA-NC	**0.122**	**0.259**	**0.187**	0.114
Enron	UMGA-R	**0.081**	0.248	0.128	0.182
	UMGA-NC	0.103	**0.199**	**0.121**	**0.126**

Table 4: Average precision index error over 10 levels of *k*-anonymity for UMGA-R and UMGA-NC on 4 clustering algorithms. Bold values indicate the method that achieves the best result (i.e. the lowest information loss) in each metric and dataset.



(a) Karate network (b) Erdős network (c) Enron network

Fig. 7: Precision index values for UMGA-R and UMGA-NC using Fastgreedy, Infomap and Multilevel clustering algorithms on Karate, Erdős and Enron datasets.

5.2.3 Empirical results

Table 4 presents the error on precision index computed using the four clustering algorithms previously described. It is crucial to consider the results on graph-mining processes, since the main goal of the anonymous data is to provide valuable information to researchers and third-parties to understand the structure and the behaviour of real networks. Hence, we have considered clustering as an important graph mining task that can be useful to quantify data utility in our experimental framework.

As we can see, our algorithm using edge neighbourhood centrality (UMGA-NC) gets better results on most of our clustering algorithms and datasets. For instance, it accomplishes the best results on 3 of 4 tested algorithms in Karate, Erdős and Enron networks. We can deepen in some specific analysis presented in Figure 7. The result of precision index using Fastgreedy algorithm in Karate network is introduced in Figure 7a. As the figure shows, the number of corrected classified vertices is always greater using the UMGA-NC algorithm, instead of using the random-based edge selection process. The results are clear: the algorithm leads to a better data utility using edge neighbourhood centrality. A similar behaviour can be seen in Figure 7b, where Infomap algorithm in Erdős dataset is used. Lastly, the largest network tested in these experiments presents alike results, as shown in Figure 7c.

Network	$\lvert V\rvert$	$\lvert E\rvert$	$\langle deg\rangle$	k
Caida	26,475	53,381	2.016	1
Amazon	403,394	2,443,408	6.057	1
Yahoo!	1,878,736	4,079,161	2.171	1

Table 5: General properties of our tested large networks.

5.3 Scalability analysis

After generic and clustering-specific information loss analysis we have performed in previous sections, we want to test our algorithms with large networks. Our main goal is to prove that they are able to deal with large networks of thousands and millions of vertices and edges. We cannot perform the previous analysis with large networks due to fact that the complexity of some measures is very expensive and they cannot be computed in reasonable time. All tests in this section are made on a 4 CPU Intel Xeon X3430 at 2.40GHz with 32GB RAM running Debian GNU/Linux.

5.3.1 Tested networks

We have tested our algorithm with three real and large networks. All these networks are undirected and unlabelled. Table 5 shows a summary of the networks' main features including number of vertices, edges, average degree and k-anonymity value. Caida [29] is an undirected network of autonomous systems of the Internet connected with each other from the CAIDA project, collected in 2007. Amazon [30] is based on "customers who bought X also bought Y" feature of the Amazon website. Yahoo! Instant Messenger friends connectivity graph (version 1.0) [45] contains a non-random sample of the Yahoo! Messenger friends network from 2003.

5.3.2 Performance of our algorithm

Table 6 shows the results of our scalability experiments in large networks. We apply the UMGA algorithm to our tested networks, using both exhaustive and greedy search methods with random edge selection (UMGA-E-R and UMGA-G-R) and neighbourhood centrality edge selection (UMGA-E-NC and UMGA-G-NC). We test our algorithm for values of $k = \{10, 20, 50, 100\}$ in each network and compute the number of possible combinations (2^q) in order to provide an approximation of the complexity. For each method, we show the computation time of the algorithm (time), the difference between the original edge set E and the anonymized one $\tilde{E}$ ($ed = |E| - |\tilde{E}|$), and the percentage of modified edges ($\%mod = 1 - \frac{|E\cap\tilde{E}|}{|E\cup\tilde{E}|}$).

Caida is a quite sparse network. More than 49% of their vertices have a degree between 1 and 10, 21.71% between 11 and 100, 18.66% between 101 and 1,000 and 10.26% have a degree greater than 1,000. The maximum degree is 2,628. Because of this, it is necessary to modify more than 6% of the edges in order to obtain a $k = 10$. This percentage augments when the value of k grows. For $k = 50$ and $k = 100$ the algorithm needs to modify the total number of edges, decreasing 9 and increasing 41, respectively. These 41 edges only represent 0.018% of the total edges, so we believe that the noise introduced in the network is minimum. We can see similar times and results for exhaustive and greedy methods, but

General			**Exhaustive Method**				**Greedy Method**			
Network	k	2^q	**ed**	**%mod**	**R time**	**NC time**	**ed**	**%mod**	**R time**	**NC time**
Caida	10	2^{24}	0	6.06%	0:00:06	0:00:36	0	6.06%	0:00:06	0:00:34
	20	2^{21}	0	11.65%	0:00:07	0:01:05	0	11.65%	0:00:07	0:01:05
	50	2^{13}	-9	18.43%	0:00:08	0:01:45	-9	18.43%	0:00:08	0:01:45
	100	2^{9}	-9	25.81%	0:00:10	0:01:48	41	25.85%	0:00:10	0:01:46
Amazon	10	2^{54}	0	0.16%	0:14:16	0:22:47	0	0.16%	0:11:54	0:19:47
	20	2^{50}	0	0.26%	2:57:37	3:07:58	0	0.26%	0:11:55	0:22:25
	50	2^{32}	0	0.39%	0:12:07	0:33:07	0	0.39%	0:12:07	0:31:17
	100	2^{30}	-1	0.53%	2:11:47	2:42:48	-1	0.53%	0:13:57	0:53:37
Yahoo!	10	2^{38}	0	0.05%	1:34:34	1:52:25	0	0.05%	1:31:40	1:50:24
	20	2^{28}	0	0.07%	2:19:21	2:46:47	0	0.07%	1:31:57	1:59:34
	50	2^{19}	0	0.13%	1:33:09	2:22:59	0	0.13%	1:32:12	2:20:54
	100	2^{16}	0	0.18%	1:34:27	2:46:33	0	0.18%	1:34:32	2:47:58

Table 6: UMGA's results for tested networks. For each network, we test our algorithms for values of $k = \{10, 20, 50, 100\}$ and computes the number of possible combinations (2^q). For each method, we show the difference between the original edge set E and the anonymized one $\tilde{E}$ (*ed*), the percentage of modified edges (*%mod*) and the computation time for UMGA-R (R time) and UMGA-NC (NC time).

UMGA-R consumes much less time than UMGA-NC. The number of possible combinations is small and the exhaustive method can deal with it.

Amazon is a network larger than Caida, so we can see greater differences between exhaustive and greedy methods. The complexity rises to 2^{54} when $k = 10$. Notice that smaller k-values imply bigger complexity since more group of vertices (g_j) are possible; and therefore, more possible combinations. When $k = 100$ there is no solution with $\sum_{j=1}^{p} m_j = 0$. For that reason, the exhaustive method explores all the possible combinations. In this case, it is 2^{30} and it can be done with reasonable time. For other values of k, there is a solution equal to 0, so the exhaustive method does not explore all combinations. Indeed, it explores less than 0.1% in all cases. However, the greedy method finds the same results in all experiments and it spends much less time. The time used by UMGA-NC is only two or three times larger than the one used by UMGA-R, much less than the difference in Caida network.

Yahoo! network is the largest tested network, but it is less sparse than others: 99.21% of the vertices have a degree between 1 and 100, 0.75% between 101 and 1,000, and only 0.03% have a degree greater than 1,000. Vertices with a degree value less than 100 are well protected and they are more than 99%. These characteristics imply that k-degree anonymous networks from $k = 10$ to $k = 100$ can be achieved with less than 0.20% of modifications in the edge set. Hence, the utility of the anonymized network will be almost intact. As the previous examples illustrate, UMGA-NC consumes more time than UMGA-R, but time is reduced to less than double.

Notice that UMGA-NC spends more time in all experiments, but the time does not grow exponentially with the network because of the heuristic approach applied. Consequently, we can see that neighbourhood centrality edge selection can deal with large networks, improving the data utility with reasonable time consuming.

6 Conclusion

In this paper we have presented a new algorithm for anonymization in large networks. It is based on edge set modification in order to fulfil the desired *k*-degree anonymity value. The new algorithm, called Univariate Micro-aggregation for Graph Anonymization (UMGA), is based on the modification of the degree sequence using univariate micro-aggregation technique. This process obtains an anonymized degree sequence which is *k*-degree anonymous and minimizes the distance from the original one. Then, we use the three basic operations to translate the modifications made on anonymized degree sequence to the network edge set.

In addition, we have proposed a method to preserve the most important edges in the network. Instead of modifying one of the possible edges randomly, this method considers the edge relevance and preserves the most important edges in the network. We demonstrate that using this method we can clearly improve the data utility, reducing the information loss on anonymized data. We also demonstrated that our algorithm is better, in terms of information loss and data utility, than the two other well-known *k*-degree anonymous algorithms.

Moreover, we have introduced a framework for testing our algorithm on clustering processes. We performed an analysis using clustering algorithms and proved that using edge neighbourhood centrality our algorithm leads the anonymization process to better clustering results, i.e. results closer to the ones performed on original data.

We have also shown that the algorithm is able to anonymize large networks. We have used three different real networks to test our algorithm based on the exhaustive and greedy methods. Both methods show good results in all networks, but the greedy method spends less time to get similar (in much cases, the same) results. In addition, the greedy method remains much more stable over time than exhaustive method. The tests proved that our algorithm can anonymize large networks based on *k*-degree anonymity concept.

Many interesting directions for future research have been uncovered by this work. It would be very interesting to investigate how this method can deal with other type of networks (directed or edge-labelled networks, for example). Additionally, other graph-mining processes can be considered to evaluate the real data utility, such as information flow, contagion processes.

Acknowledgements This work was partly funded by the Spanish Government through grants TIN2011-27076-C03-02 “CO-PRIVACY” and TIN2014-57364-C2-2-R “SMARTGLACIS”.

References

1. Adamic, L. A., and Glance, N. (2005). The political blogosphere and the 2004 U.S. election. In Int. Workshop on Link Discovery. New York, USA: ACM, pp. 36–43.
2. Backstrom, L., Dwork, C., and Kleinberg, J. (2007). Wherefore art thou r3579x? anonymized social networks, hidden patterns, and structural steganography. In Int. Conf. on World Wide Web. New York, USA: ACM, pp. 181–190.
3. Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008(10):P10008.
4. Boldi, P., Bonchi, F., Gionis, A., and Tassa, T. (2012). Injecting Uncertainty in Graphs for Identity Obfuscation. Proceedings of the VLDB Endowment, 5(11), pp. 1376–1387.
5. Bredereck, R., Froese, V., Hartung, S., Nichterlein, A., Niedermeier, R., and Talmon, N. (2014). The Complexity of Degree Anonymization by Vertex Addition. In Proc. of the 10th International Conference on Algorithmic Aspects in Information and Management. Vancouver, Canada: Springer, pp. 44–55.
6. Cai, B.-J., Wang, H.-Y., Zheng, H.-R., and Wang, H. (2010). Evaluation repeated random walks in community detection of social networks. In International Conference on Machine Learning and Cybernetics. Qingdao, China: IEEE, pp. 1849–1854.

7. Campan, A., Alufaisan, Y., and Truta, T. M. (2015). Preserving Communities in Anonymized Social Networks. Transactions on Data Privacy, 8(1), 55–87.
8. Campan, A., and Truta, T. M. (2009). Data and Structural *k*-Anonymity in Social Networks. In Privacy, Security, and Trust in KDD, USA: Springer-Verlag, pp. 33–54.
9. Casas-Roma, J., Herrera-Joancomartí, J., and Torra, V. (2013). An Algorithm For *k*-Degree Anonymity On Large Networks. In IEEE Int. Conf. on Advances on Social Networks Analysis and Mining. Niagara Falls, CA: IEEE, pp. 671–675.
10. Casas-Roma, J., Herrera-joancomartí, J., and Torra, V. (2013). Analyzing the Impact of Edge Modifications on Networks. In Int. Conf. on Modeling Decisions for Artificial Intelligence. Barcelona: Springer-Verlag, pp. 296–307.
11. Cheng, J., Fu, A. W., and Liu, J. (2010). *k*-isomorphism: privacy preserving network publication against structural attacks. In Int. Conf. on Management of Data. New York, USA: ACM, pp. 459–470.
12. Chester, S., Kapron, B. M., Ramesh, G., Srivastava, G., Thomo, A., and Venkatesh, S. (2011). *k*-Anonymization of Social Networks By Vertex Addition. In ADBIS 2011 Research Communications, pp. 107–116.
13. Chester, S., Gaertner, J., Stege, U., and Venkatesh, S. (2012). Anonymizing Subsets of Social Networks with Degree Constrained Subgraphs. In IEEE Int. Conf. on Advances on Social Networks Analysis and Mining. Washington, USA: IEEE, pp. 418–422.
14. Chester, S., Kapron, B. M., Ramesh, G., Srivastava, G., Thomo, A., and Venkatesh, S. (2013). Why Waldo befriended the dummy? *k*-Anonymization of social networks with pseudo-nodes. Social Network Analysis and Mining, 3(3), pp. 381–399.
15. Clauset, A., Newman, M. E. J., and Moore, C. (2004). Finding community structure in very large networks. Physical Review E, 70(6):1–6.
16. De Capitani di Vimercati, S., Foresti, S., Livraga, G., and Samarati, P. (2012). Data Privacy: Definitions and Techniques. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 20(6), pp. 793–818.
17. C. Dwork. (2006). Differential Privacy. Int. Conf. on Automata, Languages and Programming, 4052, pp. 1–12.
18. Dwork, C. (2011). A firm foundation for private data analysis. Communications of the ACM, 54(1), pp. 86–95.
19. Ferri, F., Grifoni, P., and Guzzo, T. (2011). New forms of social and professional digital relationships: the case of Facebook. Social Network Analysis and Mining, 2(2), pp. 121–137.
20. Girvan, M., and Newman, M. E. J. (2002). Community structure in social and biological networks. Proceedings of the National Academy of Sciences of the United States of America, 99(12):7821–7826.
21. S. L. Hansen and S. Mukherjee. (2003). A Polynomial Algorithm for Optimal Univariate Microaggregation. IEEE Trans. on Knowledge and Data Engineering, 15(4), pp. 1043–1044.
22. Hartung, S., Hoffmann, C., and Nichterlein, A. (2014). Improved upper and lower bound heuristics for degree anonymization in social networks. In Proc. of the 13th International Symposium on Experimental Algorithms. Copenhagen: Springer Verlag, pp. 376–387.
23. Hay, M., Miklau, G., Jensen, D., Weis, P., and Srivastava, S. (2007). Anonymizing Social Networks. Computer Science Department, University of Massachusetts Amherst, Technical Report No. 07-19, 2007.
24. Hay, M., Miklau, G., Jensen, D., Towsley, D., and Weis, P. (2008). Resisting structural re-identification in anonymized social networks. Proc. of the VLDB Endowment, 1(1), pp. 102–114.
25. Hay, M., Li, C., Miklau, G., and Jensen, D. (2009). Accurate Estimation of the Degree Distribution of Private Networks. In IEEE Int. Conf. on Data Mining (ICDM). Miami, FL: IEEE, pp. 169–178.
26. Hay, M., Liu, K., Miklau, G., Pei, J., and Terzi, E. (2011). Privacy-aware data management in information networks. In Int. Conf. on Management of Data. New York, USA: ACM Press, pp. 1201–1204.
27. Krebs, V. (2006). http://www.orgnet.com.
28. Lancichinetti, A., and Fortunato, S. (2009). Community detection algorithms: a comparative analysis. Physical Review E, 80(5):056117.
29. Leskovec, J., Kleinberg, J., and Faloutsos, C. (2007). Graph evolution: Densification and Shrinking Diameters. ACM Trans. on Knowledge Discovery from Data, 1(1), pp. 1–40.
30. Leskovec, J., Adamic, L. A., and Huberman, B. A. (2007). The dynamics of viral marketing. ACM Trans. on the Web, 1(1), 5:1–5:46.
31. Leskovec, J., Lang, K., Dasgupta, A., and Mahoney, M. (2009). Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters. Internet Mathematics. 6(1):29–123.
32. Li, N., Li, T., and Venkatasubramanian, S. (2007). *t*-Closeness: Privacy Beyond *k*-Anonymity and ℓ-Diversity. In IEEE Int. Conf. on Data Engineering. IEEE, pp. 106–115.
33. Liu, K., and Terzi, E. (2008). Towards identity anonymization on graphs. In ACM SIGMOD Int. Conf. on Management of Data. New York, USA: ACM, pp. 93–106.

34. Lu, X., Song, Y., and Bressan, S. (2012). Fast Identity Anonymization on Graphs. In Proc. of the 23rd International Conference on Database and Expert Systems Applications. Vienna, Austria: Springer Berlin Heidelberg, pp. 281–295.
35. Nagle, F. (2013). Privacy Breach Analysis in Social Networks. In T. Özyer, Z. Erdem, J. Rokne, and S. Khoury (Eds.), Mining Social Networks and Security Informatics. Dordrecht: Springer Netherlands, pp. 63–77.
36. Nguyen, H. H., Imine, A., and Rusinowitch, M. (2015). Anonymizing Social Graphs via Uncertainty Semantics. In Proc. of the 10th ACM Symposium on Information, Computer and Communications Security. Singapore: ACM, pp. 495–506.
37. Machanavajjhala, A., Kifer, D., Gehrke, J., and Venkitasubramaniam, M. (2007). ℓ-diversity: Privacy beyond k*k*-anonymity. ACM Trans. on Knowledge Discovery from Data, 1(1), 3:1–3:12.
38. McSherry, F., and Mironov, I. (2009). Differentially private recommender systems. In Proc. of the 15th ACM SIGKDD iInt. Conf. on Knowledge Discovery and Data Mining. New York, USA: ACM, pp. 627–636.
39. Pons, P., and Latapy, M. (2005). Computing communities in large networks using random walks. In 20th International Symposium Computer and Information Sciences. Istanbul, Turkey: Springer, pp. 284–293.
40. Rosvall, M., and Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. Proceedings of the National Academy of Sciences of the United States of America, 105(4):1118–1123.
41. Samarati, P. (2001). Protecting Respondents' Identities in Microdata Release. IEEE Trans. on Knowledge and Data Engineering, 13(6), pp. 1010–1027.
42. Sweeney, L. (2002). *k*-anonymity: a model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10(5), pp. 557–570.
43. Tripathy, B. K., and Panda, G. K. (2010). A New Approach to Manage Security against Neighborhood Attacks in Social Networks. In IEEE Int. Conf. on Advances on Social Networks Analysis and Mining. IEEE, pp. 264–269.
44. Wu, W., Xiao, Y., Wang, W., He, Z., and Wang, Z. (2010). *k*-symmetry model for identity anonymization in social networks. In Int. Conf. on Extending Database Technology. New York, USA: ACM, pp. 111–122.
45. Yahoo! Webscope, Yahoo! Instant Messenger friends connectivity graph, version 1.0. `http://research.yahoo.com/Academic\_Relations`
46. Ying, X., and Wu, X. (2008). Randomizing Social Networks: a Spectrum Preserving Approach. In SIAM Conf. on Data Mining. Atlanta, Georgia, USA: SIAM, pp. 739–750.
47. Ying, X., Pan, K., Wu, X., and Guo, L. (2009). Comparisons of randomization and *k*-degree anonymization schemes for privacy preserving social network publishing. In Workshop on Social Network Mining and Analysis. New York, USA: ACM, pp. 10:1–10:10.
48. Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. Journal of Anthropological Research, 33(4):452–473.
49. Zhang, K., Lo, D., Lim, E.-P., and Prasetyo, P. K. (2013). Mining indirect antagonistic communities from social interactions. Knowledge and Information Systems (KAIS), 35(3):553–583.
50. Zheleva, E., and Getoor, L. (2011). Privacy in Social Networks: A Survey. Social Network Data Analytics. Springer, pp. 277–306.
51. Zhou, B., and Pei, J. (2008). Preserving Privacy in Social Networks Against Neighborhood Attacks. In IEEE Int. Conf. on Data Engineering (ICDE). Washington, USA: IEEE, pp. 506–515.
52. Zhou, B., and Pei, J. (2011). The *k*-anonymity and ℓ-diversity approaches for privacy preservation in social networks against neighborhood attacks. Knowledge and Information Systems, 28(1), pp. 47–77.
53. Zhou, B., Pei, J., and Luk, W. (2008). A brief survey on anonymization techniques for privacy preserving publishing of social network data. ACM SIGKDD Explorations Newsletter, 10(2), pp. 12–22.
54. Zou, L., Chen, L., and Özsu, M. T. (2009). *k*-Automorphism: A General Framework For Privacy Preserving Network Publication. Proc. of the VLDB Endowment, 2(1), pp. 946–957.

Author Biographies

Jordi Casas-Roma is lecturer at Faculty of Computer Science, Multimedia and Telecommunications at Universitat Oberta de Catalunya (UOC) and also part-time associate professor at Universitat Autònoma de Barcelona (UAB) . He holds a Ph.D on Computer Science (UAB, 2014), Master degree on Advanced Artificial Intelligence (Universidad Nacional de Educación a Distancia, UNED, 2011) and Bachelor degree on Computer Science (UAB, 2002). His teaching activities mainly concentrate in computer security, big data and databases. He belongs to the research group KISON (K-ryptography and Information Security for Open Networks) and his main research interests include graph theory, social network analysis, privacy-preservation and data mining.

Jordi Herrera-Joancomartí is associate professor at Department of Information and Communications Engineering in the Universitat Autònoma de Barcelona (UAB). He finished a graduate in Mathematics at Universitat Autònoma de Barcelona in 1994 and he received his Ph.D. degree in 2000 from Universitat Politècnica de Catalunya. In 2000, he joined the Universitat Oberta de Catalunya and founded the KISON research group in which he is now an external researcher. Currently, he is a member of the SENDA research group at UAB. His research interests include topics in the field of privacy and computer security and more precisely the bitcoin ecosystem, mobile crowdsensing scenarios and privacy in Online Social Networks.

Vicenç Torra is a Professor at the School of Informatics, University of Skövde in Sweden. He was Associate Professor at the Artificial Intelligence Research Institute (IIIA-CSIC), and visiting researcher at the University of Tsukuba (Japan). He has written four books. His fields of interests are decision making, information fusion and soft computing tools and their applications to privacy issues. He is editor of the Transactions on Data Privacy, and founder of the conference series Modeling Decisions for Artificial Intelligence (MDAI).