A Comparison of Clustering and Modification based Graph Anonymization Methods with Constraints

David F. Nettleton[a,b]
Contract Researcher
[a]IIIA-CSIC,Bellaterra,Spain;
[b]Dept. Information Technology and Communications,
Pompeu Fabra University,
Barcelona, Spain

Vicenc Torra
Associate Research Professor
IIIA-CSIC, Bellaterra, Spain

Anton Dries
Post-Doc Researcher
Department of Computer Science,
KU Leuven, Leuven, Belgium

ABSTRACT

In this paper a comparison is performed on two of the key methods for graph anonymization and their behavior is evaluated when constraints are incorporated into the anonymization process. The two methods tested are node clustering and node modification and are applied to online social network (OSN) graph datasets. The constraints implement user defined utility requirements for the community structure of the graph and major hub nodes. The methods are benchmarked using three real OSN datasets and different levels of k–anonymity. The results show that the constraints reduce the information loss while incurring an acceptable disclosure risk. Overall, it is found that the modification method with constraints gives the best results for information loss and risk of disclosure.

General Terms

Data Hiding, Search Techniques, Information and Knowledge, Computational Intelligence.

Keywords

Data privacy, information hiding, graphs and networks, online social networks, anonymization, information loss, risk of disclosure

1. INTRODUCTION

Data Privacy in graphs has recently become a topic of renewed interest by researchers, partially due to the emergence of online social networks (OSN), which can be represented and analyzed as graphs. OSN data is of great potential for data analysts from different disciplines, but also represents a threat to data privacy if it is used for the wrong motives. However, the anonymization of graph data represents a challenge, given that anonymization techniques may impair essential structural information in the graph.

Furthermore, it should be the user of the data who defines the utility requirements. These requirements can be expressed as constraints applied to the perturbation process. However, it is possible that the constraints increase the risk of disclosure by information leakage. Hence it is of interest to establish which perturbation method gives the best results for information loss (utility) and risk of disclosure, when the constraints are applied.

The objective of the current work is to test two of the most used perturbation methods with and without constraints in order to evaluate their relative performance.

In the literature, some authors have considered anonymization as a graph partitioning/clustering task based on an overall utility measure[1] or by modifying nodes using a cost function[2]. However, there is a lack of work in the graph anonymization field on benchmarking these methods together and under restricted conditions.

The main contributions of the paper are:

- A comparison of clustering and modification based graph perturbation methods.
- The incorporation of restriction mechanisms in the perturbation methods, which act on the community structure and major hub nodes.
- A comparison of restricted versus unrestricted perturbation methods.

The structure of the paper is as follows: in Section 2 the state of the art is discussed; in Section 3 some preliminary concepts are presented; in Section 4 the anonymization methods are described; in Section 5 the metrics are described for information loss and adversary knowledge, and the privacy model is defined; in Section 6 the empirical results are presented for the different perturbation methods with and without restrictions; finally, in Section 7 the present work is summarized.

2. RELATED WORK

In the following sections, the theme of privacy preserving social network publishing is considered from two general perspectives: (i) adversary information and (ii) anonymization methods.

2.1 Adversary Information

Adversary information is a way of evaluating the disclosure risk and normally involves formulating and submitting informational queries on the data. These queries must take into account the type and amount of knowledge available to the adversary. In [1], Hay et al. consider what an adversary may know or deduce from a graph in terms of three different families of topological queries (as opposed to isomorphic properties). In general, the queries focus on eliciting information about the immediate or close neighborhood of a target node. Wondracek [3] presents a different approach, in which the adversary uses a malicious website to obtain information about users of an on-line social network. Backstrom et al. [4], on the other hand, consider active and passive adversary strategies. In active strategies, the adversary actively tries to affect the data to make it easier to decipher. In passive strategies, the adversary simply observes the data as it is presented. In [5], Cheng et al. consider a K–Isomorphism approach to privacy preserving network publication which protects against structural attacks. The authors refer to a popular type of attack described by Backstrom et al. in [4], which involves the use of embedded sub-graphs. They extend this idea by defining two realistic disclosure targets which are based on node information and link information, respectively.

2.2 Anonymization Methods

In the literature different methods have been used for graph anonymization and in particular, obtaining k−anonymity of the vertices V in a graph G while minimizing information loss. For the purposes of the current work, the methods will be divided into two groups: (a) node modification approaches and (b) node clustering approaches. In the context of data privacy in general, Sweeney's paper [6] was the first to define k−anonymity, and more recently the paper by De Capitani et al. [7], gave key definitions for privacy levels, information loss and risk of disclosure. Also, in [8] Zhou considered l−diversity together with k−anonymity to give a stronger anonymity guarantee.

2.2.1 Node Modification Approaches

Node modification approaches act by choosing similar nodes and making them identical. This can be done by adding nodes to make their degrees the same and by adding edges to make their immediate neighborhood connectivity the same. Using this method, k−anonymity is achieved by obtaining that every node in the graph has at least k-1 other nodes which are indistinguishable from it. Zhou[2] presents a method which selects nodes based on a cost function and then anonymizes them by adding nodes and edges to their neighborhoods. In [9], Nettleton et al. compare two different types of online social network from a data privacy perspective, using 'add link' as the perturbation operator. In [10], Hay et al. presented a simple graph anonymization based on random addition and deletion of edges. The disclosure method attempts re-identification using two types of queries, vertex refinement and sub-graph knowledge.

2.2.2 Node Clustering Approaches

Node clustering approaches act by choosing similar nodes and physically grouping them. This can be done by a k−means type algorithm or by a similarity/distance metric to choose similar nodes. Using this method, k−anonymity is achieved by obtaining that every node in the graph is incorporated into a cluster within which there are at least k-1 other nodes. Skarkala et al. [11] present an approach for node clustering/grouping which takes into consideration the privacy protection of the edge weights. Skarkala employs a similarity function to form clusters each containing at least k nodes. Nettleton in [12] applied a perturbation method based on node aggregation and a similarity metric with fixed weights for choosing node pairs. Different types of clustering, fuzzy (fuzzy c-Means) and crisp (k−Means) were applied to graph statistical data in order to evaluate the information loss due to perturbation. In [1], Hay presented an approach in which nodes are grouped into partitions based on a utility function incorporating a distance metric in terms of the number of edges. In order to settle the partitions, the entropy was calculated for the entire graph. Hay's method[1] is distinct to our approach given that Hay’s partitions are guaranteed as having at least k nodes but can have many more (e.g. hundreds, for k=16), whereas our method guarantees between k and 2k-1 nodes in each cluster.

2.2.3 Other Approaches

In [13], Bonchi et al. offer a somewhat different vision of graph anonymization, based on an entropy-based quantification of anonymity. It represents a global method which uses a local quantification based on a-posteriori belief. They also propose a controlled random edge removal (as opposed to adding edges) which they call 'random sparsification'. In [14], Ying and Wu present a spectrum preserving approach to randomizing social networks. The authors based their approach on the observation that many graph structures have a strong association with the spectrum. From this came the idea to define a perturbation strategy which minimizes the change in some given eigenvalues, while maintaining privacy protection.

3. PRELIMINARIES

A graph G is defined as a set of vertices V interconnected by a set of edges E, thus giving G = (V, E). In the current work each node has an arbitrary identifier for data processing purposes however it is assumed this identifier will have no meaning for the adversary and cannot be considered a label. Hence, the graph is considered as unlabeled. A neighborhood sub-graph G^n = (V', E') is a subset of G around a given reference node v^r. Hence $v^r \in$ V' and all other vertices v' $\in$ V' are adjacent vertices of v^r.

In G, a special set of vertices is defined as follows: a *hub* vertex v^h is defined as being a node with a relatively high number of direct connections to other nodes, as quantified by Kleinberg’s metric [15] which is designated as a function *h(v)*. The set of hub vertices is defined as $V^h \subset V$, and $v^h \in V^h$ when $h(v^h)$ is in the top 12% percentile of all values for *h(v)*. The top percentile value for hubs was chosen by empirical study of the respective metric distributions.

Also, a partitioning is mapped onto G which is derived from the community structure identified by the Louvain Method [16]. The mapping of the vertices onto the community structure can be defined as a function $G_c : v_i \rightarrow c$. Hence, a given vertex v_i will belong to one and only one community *c*.

The anonymization method chooses pairs of nodes (v_i, v_j), based on a distance function D(v_i, v_j) and subject to the following restrictions: $v_i \notin V^h$, $v_j \notin V^h$, G_c (v_i) = G_c (v_j). These definitions implement the hub and community restrictions, respectively.

4. DESCRIPTION OF ANONYMIZATION METHODS

In this section the two anonymization methods will be described: clustering and modification. The constraints and the distance metric for sub-graph matching will also be described. Both methods are based on selecting the k most similar nodes and then perturbing them to make them identical, either by clustering or by modification. Each method is applied with and without constraints. The methods are listed in Table 1, and will be explained in detail in the following Sections.

4.1 Graph Alteration Methods – Clustering and Modification

In order to compare the relative performance of the constrained and non constrained approaches, two of the most common state of the art graph alteration methods in the literature have been used: (i) node modification and (ii) node clustering. It is noted that both the methods use the same node matching function, which is described in Section 4.3.

4.1.1 Node Modification

For this method a technique has been implemented which is based on node addition and edge addition/deletion, obtaining k–anonymity using a cost function based on the expected perturbation. This method is similar to the one presented by Zhou in [2]. Due to the unavailability of the original code, a version was programmed and tested by the authors. The implementation uses the distance measure (see Section 4.3) as the cost function, and selects nodes for matching in descending order of degree, as indicated in [2]. Also, when a node is added to increase the degree, the smallest degree is chosen first, again following the guidelines of [2].

Table 1. Summary of anonymization methods

Type	Name	Restrictions
Cluster	cluster_r	Yes
Cluster	cluster	No
Modify	modify	No
Modify	modify_r	Yes

Finally, edges are added to obtain the same internal degree sequences and minimize the difference between the respective sub-graph clustering coefficients. For node modification, two sub-graphs G_1 and G_2 are considered equal when, for the reference node g_1 of G_1 and the reference node g_2 of G_2: degree(g_1) = degree(g_2), num_edges(G_1) = num_edges(G_2) and internal_degree_sequence(G_1) = internal_degree_sequence(G_2). It is noted that the adversary queries used (see Section 5.3) are based on the structural similarity of node neighborhoods [1] rather than on isomorphic properties[4]. Hence, this equality criterion is adequate for both the type of adversary queries considered in the current paper, and in order to compare the relative performance of the different methods under the same conditions. Two versions are implemented: the first has no restrictions so it can choose nodes to match anywhere in the graph. This is called 'modify'. The second is constrained by the community and hub nodes, and is called ‘modify_r’.

4.1.2 Node Clustering

For this method, a node aggregation method has been implemented which groups the nodes into super-nodes each of which contains at least k and at most *2k-1* of the original nodes. An optimum clustering is obtained by using a similarity function (see Section 4.3) to pair the most similar nodes for aggregation for each *k* value. Hence, for each node in the graph, the *k-1* most similar nodes will be identified and these nodes will unified into one super-node. If there are *2k* or more identical nodes (that is, taking into account that some nodes will already be identical in the graph), they will be grouped in super-nodes each containing at least *k* nodes and at most *2k-1*. Two versions are implemented: the first has no constraints so it can choose nodes to match anywhere in the graph. This is called 'cluster'. The second is constrained by the community and hub nodes and is called 'cluster_r'.

4.2 Search Constraints

Two search strategies are considered: (i) no constraints, in which nodes can be searched for and matched anywhere in the graph; (ii) with constraints, in which node search and matching is restricted to node pairs in the same community and excludes top hub nodes. A search is performed for the best *k-1* matches of a given reference node. For the constrained approach, a “Community Structure” algorithm is initially executed to partition the complete graph into “communities”. Blondel’s algorithm, also known as the Louvain Method[16], is used for this purpose. The top 12% percentile hub nodes are also identified by calculating their corresponding metrics using the HITS algorithm[15]. The percentile values were chosen by empirical study of the metric distributions. In practice, these top percentile proportions tend to represent a small number of high degree nodes in the graph.

4.3 Similarity Metric for Sub-graph Matching

In order to calculate the similarity between two node neighborhoods, computation cost is a key consideration. Hence, a similarity metric has been chosen which calculates a distance based on sub-graph characteristics which can be pre-calculated. The sub-graph characteristics are the degree of the reference node, number of edges in the sub-graph, clustering coefficient and statistics of the degrees of the neighbor nodes. The former characteristics are designed to reflect the internal structure of the sub-graph, whereas the latter characteristics reflect a key descriptive feature of the neighbors (their degree), which effectively considers the neighborhood at a distance of 2 from the reference node. A weight vector is trained using a simulated annealing process with an exact isomorphism matcher as the target (optimum) value. The trained similarity metric approximates an isomorphism matcher and also takes into account the degrees of the neighbors of the reference node. The neighborhood sub-graph matching method used in this work was recently presented as a European Patent application[17]. The algorithm operates in two phases: (i) a 'training' phase in which the weights are learned for the distance metric from samples and (ii) a 'runtime' phase which processes the complete dataset, matching nodes using the trained distance metric, and anonymizing their sub-graphs to obtain k–anonymity.

4.4 Pseudo-code of Data Processing

In this section, the main procedures used for data processing are defined: “Pre-calculate”, "Train" and "Run" (the latter calls each of the four methods).

```
Main Procedure
Input: original graph G = (V, E), anonymization level k
Output: anonymized graph G’
Pre-calculate
    Calculate statistics for each neighborhood sub-graph
    $G_1 \dots G_n$
    Calculate hub metrics
    Calculate communities $c_1 \dots c_i$ using Louvain method
Train
    Apply simulated annealing process to find
    optimum weights for distance function
Run
    Let H be the set of hub nodes h above the hub percentile
threshold
    Let k be the privacy level
    For each (g) $\in$ (G) , g $\notin$H do
        Let $c_i$ be the community to which node g belongs
        Let $G_{g1}$ be the neighborhood sub-graph for g
        Call methods
            Clustering methods:
                Find k-1 nodes most similar to $g^{\ddagger}$
                    cluster_r(graph $G_{g1}$, $c_i$, k)  //  restricted
                    cluster(graph $G_{g1}$, k)          // unrestricted
                 Aggregate the k neighborhood sub-graphs $G_g$ and
                   [$G_{g2} \dots G_{gk}$] by calling
                 Aggregate(vector of sub-graphs[$G_{g1} \dots G_{gk}$])
            Modification methods:
                Find k-1 nodes most similar to $g^{\ddagger}$
                    modify_r(graph $G_{g1}$, $c_i$, k)  // restricted
                    modify(graph $G_{g1}$, k)          // unrestricted
                Modify the k neighborhood sub-graphs
                [$G_{g2} \dots G_{gk}$] to make them the same as $G_{g1}$ by
                calling
        **Modify**(vector of sub-graphs[$G_{g1}$ ... $G_{gk}$])
    **End for each**
```

‡Each method returns the best *k-1* matches [G_{g2} ... G_{gk}] which comply with restrictions

5. METRICS FOR INFORMATION LOSS, PRIVACY LEVEL AND RISK OF DISCLOSURE

In this Section, the definitions are given for information loss and risk of disclosure. Information loss is defined in the habitual manner, as the change in correlation between the variable in the original file and the corresponding variable in the perturbed file. For risk of disclosure, a set of candidate anonymity queries are defined, similar to those of Hay[1].

5.1 Information Loss

Four metrics are used in order to evaluate information loss. The first two are basic graph statistics (degree, clustering coefficient), and the last two are related to the community structure of the graph (hub value and number of communities). The distribution of each variable in the original data file is correlated with that of the same variable in the perturbed file, and the deviation from 1 is the information loss.

inf loss$_1$	degree
inf loss$_2$	clustering coefficient
inf loss$_3$	hub value†
inf loss$_4$	number of communities‡

†As calculated by HITS algorithm; ‡as calculated by Louvain method

The clustering coefficient calculation has been implemented in Java. The hub value (HITS) and the community partitioning have been calculated using the Gephi software[18]. **Hub metric (HITS hub):** A hub node is characterized by having a large number of direct connections to other nodes. In order to quantify the hub value of a node, the popular HITS algorithm has been used, as defined by Kleinberg in [15]. **Communities**: The community partitioning is a key characteristic of the graph that is to be preserved. Information loss is measured by the number of communities into which the graph is partitioned, as calculated by the Louvain method [16]. In the following, the four information loss metrics are designated as m_1 to m_4. If G is the original graph, G' the perturbed graph, m_1 the degree values for the original graph, and m_1' the degree values for the perturbed graph, then the information loss will be:

$$\mathbf{IL}(G, G', m_1) = 1 - \mathbf{corr}(m_1, m_1') \quad (1)$$

where IL is the information loss function and *corr* is a correlation function. The information loss for metrics m_2 and m_3 would follow in a similar manner. In the case of m_4, the absolute difference is taken between the number of communities N_c in G and the number of communities N_c' in G', thus:

$$\mathbf{IL}(G, G', m_4) = |\mathbf{diff}(m_4, m_4')| \quad (2)$$

The value obtained from equation (2) can be normalized in order to compare between different benchmark datasets.

5.2 Definition of Privacy for Clustering and Modification Approaches

The objective of anonymization is to obtain a given anonymity level of *k*. The clustering algorithm is given the parameter *k* and produces a graph consisting of super-nodes which contain a minimum of *k* and a maximum of *2k-1* basic nodes. If a super-node reaches a size of *2k* nodes, it will be divided into two super-nodes, each containing *k* nodes. Nodes are grouped based on similarity using the distance metric described in Section 4.3. Hence, nodes are grouped into partitions so that an adversary will be unable to distinguish between the nodes in a partition. The probability that an adversary successfully re-identifies a node will be between *1/k* and 1/(*2k-1*), divided by the number of super-nodes created for the given reference node. For the modification algorithm, an approximation of Zhou's method[2] has been implemented which, for a given node, modifies *k-1* other nodes to make them the same (using our distance based similarity metric). That is, for each node there will be *k-1* other nodes with the same degree, number of edges in the neighborhood sub-graph, and same clustering coefficient (that is, the connectivity between neighbors). Hence, the probability of an adversary re-identifying a node will be at least *1/k*. It is noted that nodes which are already identical will not be modified and there will probably be nodes in the graph which already have more than *k* identical nodes (especially those with a low degree). An anonymity model is employed in which a graph satisfies *k–candidate* anonymity if for every structural query over the graph, there exist at least *k* nodes that match each adversary query.

5.3 Adversary Knowledge – Structural Queries

In order to evaluate what the adversary knows or can deduce from the graph, similar lines to Hay[1] have been followed. Firstly, *vertex refinement* will be considered, followed by the *hub fingerprint*.

Vertex refinement: H_1 (x) returns the degree of x, H_2 (x) returns the multi-set of each neighbors' degree, and so on. In general, $H_i(x)$ returns the multi-set of values which are the result of evaluating H_{i-1} on the set of nodes adjacent to x. In the present work, up to two levels of query, H_1 and H_2 are considered, as defined in [1].

Hub fingerprint: a hub fingerprint query $F_i(x, HB)$ gives a list of the shortest paths from node x to each of the hub nodes defined in the vector HB. Hay defines HB as the five highest degree nodes for the Enron dataset and the ten highest degree nodes for the HepTh and Net-trace datasets. Following Hay[1] it is assumed that the value *i* designates the maximum distance of visible hub connections. If the shortest path to a hub exceeds the 'visibility horizon' then the distance is assigned a value of zero (open world assumption). Hence, query $F_1(x$, HB) returns the list for x with a visibility horizon of 1, and $F_2(x$, HB) returns the list for x with a visibility horizon of 2. As an example, consider $F_2(x$, HB), HB = {a, b, c} which gives a resulting distance vector of {2, 2, 0}. This means that node x is at distance 2 from hubs 'a' and 'b', and at a distance greater than 2 (beyond the visibility horizon) from hub 'c'.

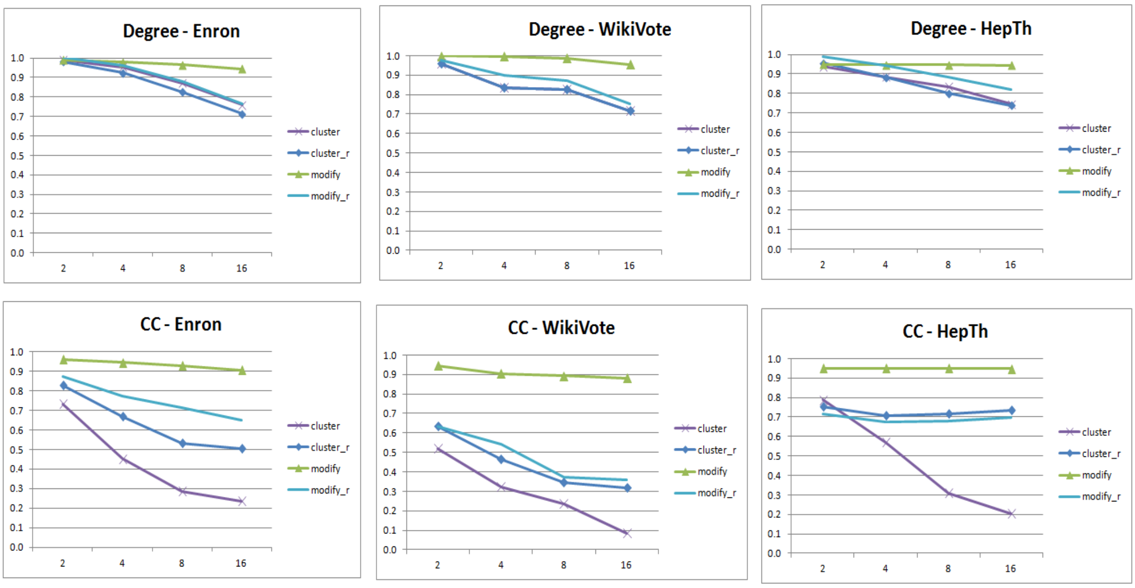

Fig 1: Information Loss: effect of anonymization on metrics for different datasets and perturbation methods. The figures show the degree of correlation (y-axis) between the original data and perturbed data for different values of k (x-axis).

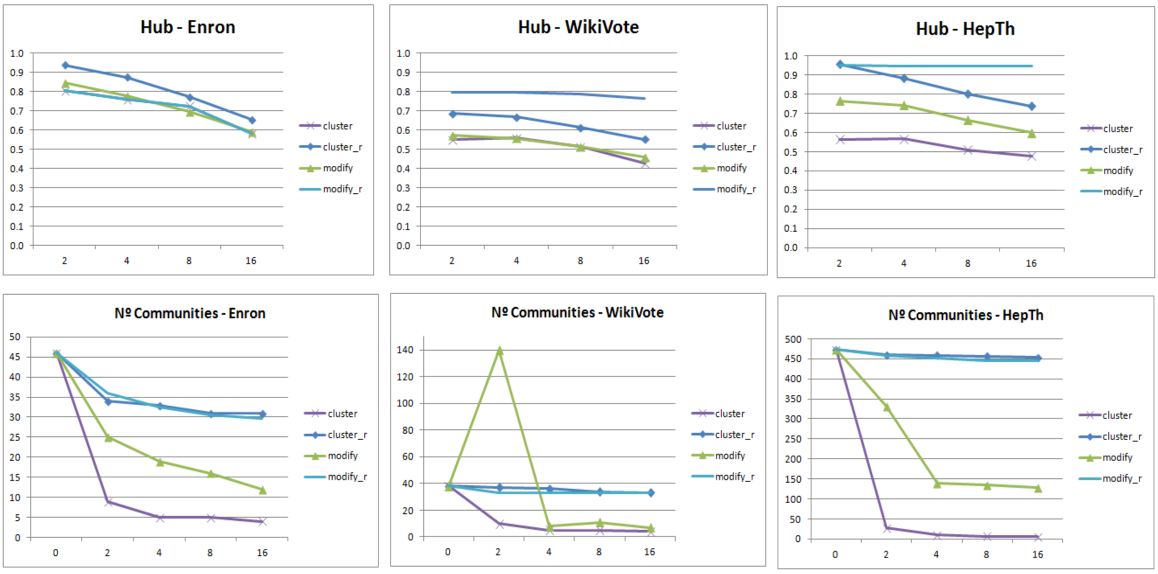

Fig 2: Information Loss: effect of anonymization on metrics for different datasets and perturbation methods. For communities, the figures show the raw data values (y-axis) for different values of k (x-axis).

6. EMPIRICAL TESTING AND RESULTS

In this Section, the results for information loss and risk of disclosure are presented for the different methods, metrics and datasets.

6.1 Datasets

The Ca-HepTh[19], Enron [20] and WikiVote [21] datasets have been used for empirical testing. These datasets offer distinct statistical characteristics and are widely used in the graph privacy literature, which allows other researchers to compare results. In the remainder of the paper, these datasets will be referred to as 'HepTh', 'Enron' and 'WikiVote', respectively. The 'HepTh' and 'WikiVote' datasets were taken directly from the Stanford Large Network Dataset Collection (SNAP) website (available at http://snap.stanford.edu/data/). In the case of the 'Enron' dataset, the data was processed by the authors from the mysql dump file available at http://www.isi.edu/~adibi/Enron/Enron.htm.

6.2 Information Loss vs. Anonymization Level (k)

For the metrics of Figure 1 (degree and cc) and the first metric of Figure 2 (hub), the information loss is quantified by first calculating the graph metrics for the different graph datasets corresponding to k=0, k=2, k=4, k=8 and k=16. Then, the value was correlated for each metric for the k=0 dataset with each of the other datasets (k=2 to k=16). The difference between the correlations is then interpreted as the information loss. For the second metric of Figure 2, 'number of communities', the absolute values are plotted and compared.

In Figures 1 and 2, the information loss is depicted for progressively increasing anonymization levels. That is, increasing values of *k*.

Tables 2 and 3 show a quantified summary of the relative performance of the methods, in terms of the number of times they were first, second, third or fourth best in each of the cases shown in Figures 1 and 2. A lower overall score means a better relative performance. For example, in Table 2 'modify' came first for the 'degree' metrics for all three datasets (first row of Figure 1). Hence its score is (1+1+1)/3 = 1.0. 'cluster', on the other hand, came equal second for 'degree-Enron', equal third for 'degree-WikiVote' and third for 'degree-HepTh'. Hence its score is (2+3+3)/3=2.7. If two methods gave a tie, for example, for first position in a given case, both methods were awarded one point for first position. It is concluded that 'modify_r' is the overall winner (rank=1), followed by 'modify' However, for the 'NC' and 'hub' metrics, 'cluster_r' came first and second, respectively. Hence, it can be concluded that the restrictions (community, hub) mitigated the information loss as expected. It can also be observed that some of the relative performances are dataset and metric dependent.

Table 2. Information loss: relative performance of methods by metric

	cluster_r	modify	cluster	modify_r
degree	3.3	1.0	2.7	2.0
cc	2.7	1.0	4.0	2.3
hub	1.7	3.0	3.7	1.3
NC	1.0	2.7	3.0	1.3
Rank	3	2	4	1

Table 3. Information loss: relative performance of methods by dataset

	cluster_r	modify	cluster	modify_r
Enron	2.0	1.7	3.0	1.7
WikiVote	2.2	2.2	3.5	1.7
HepTh	2.2	1.7	3.5	1.7
Rank	3	2	4	1

6.3 Risk (Adversary Information) vs. Anonymization Level (k)

The risk is quantified by applying three different adversary queries, which have been previously described in Section 5.3. The risk is measured in terms of candidate set sizes, following the guidelines of Hay[1]. That is, the highest risk exists for nodes for the lowest candidate set size (=1), whereas the lowest risk exists for nodes for the highest candidate set size. In Figures 3 and 4 the risk is plotted for each of the adversary queries, for each dataset and for increasing values of *k*. For space restrictions, only the lowest risk candidate set is shown for each adversary query. The proportion of nodes in the lowest risk candidate set is a key indicator of risk and was the candidate set which best characterized the adversary queries and methods. The candidate sets for adversary queries 1, 2, 4 and 5 were defined with the following frequencies: '=1', '2-4', '5-10', '11-20' and '>20'. It was observed that candidate sets with frequencies less than *k* (the privacy level) had zero members in all cases, thus confirming that k−anonymity was achieved.

6.3.1 Adversary Query 1: vertex refinement $H_1(x)$

Figure 3 (row 1) shows the trends for the different candidate sets, datasets and values of *k*, for the first adversary query, vertex refinement $H_1(x)$. This query simply returns the degree of a given node. In Figure 3 (row 1), which shows the proportion of nodes in the candidate set, it can be observed for all original datasets (*k*=0) that the great majority (90%) of the degree values are in the highly frequent candidate set ('>20', low risk). The remaining 10% are distributed through the other higher risk candidate sets: '=1', '2-4', '5-10' and '11-20'. Looking at Figure 3 (row 1), it can be seen that all methods follow a similar trend, with the exception of 'cluster' which shows a lower proportion of nodes in the '>20' bucket, with respect to the other methods.

6.3.2 Adversary Query 2: vertex refinement $H_2(x)$

Figure 3 (row 2) shows the trends for the different candidate sets, datasets and values of *k*, for the second adversary query, vertex refinement $H_2(x)$. This query returns the degrees (in a vector) of each of the immediate neighbors of a given node. In terms of the datasets, all methods follow a similar trend except for the 'HepTh' dataset. In this last case the 'cluster' method displays a significantly smaller proportion in contrast to the other three methods. Also, it can be seen that the 'modify_r' method displays a slightly higher relative proportion of nodes for the 'WikiVote' and 'HepTh' datasets.

6.3.3 Adversary Query 3: hub fingerprint $F_2(x, H)$

Figure 4 shows the trends for the different candidate sets, datasets and values of *k*, for the third adversary query, hub fingerprint (See Section 5.3). It is recalled that this query returns a vector of the shortest path length to a set of 10 top hubs in the graph. The 'hub' value for each node is quantified by calculating the HITS 'Hub Update Rule' metric, as commented in Section 5.1. With respect to the original datasets (*k*=0), it can be observed that the majority of the vector frequencies are in the '>20' candidate set. This set initially contained approx. 69% of the nodes for 'Enron', 63% for 'WikiVote' and 92% for 'HepTh'. As a result of anonymization up to k=16, in general an increase can be seen in the '>20' low risk set. In Figure 4, it can be seen that 'modify' shows the highest relative proportion for all datasets, followed by 'modif_r', whereas 'cluster' has the lowest or equal lowest.

6.3.4 Summary of the adversary query results

In this Section an overall picture will be presented of the results of the different adversary queries, taking into account the detailed analysis which has already been seen in Sections 6.3.1 to 6.3.3. In order to synthesize the results in a quantitative manner, the methods and datasets will be ranked in terms of their performance for increasing values of *k* and adversary query type. The candidate set with the highest number of candidates (lowest risk) will be used as the benchmark. If more candidates fall into this category then the overall identification risk will be lower. The scoring scheme for Tables 4 and 5 is calculated in the same way as for Tables 2 and 3 in Section 6.2.

Tables 4 and 5 contain a quantified summary of the relative performance of the methods, based on the number of times they were first, second, third or fourth best in each of the cases shown in Figures 3 and 4. It can be seen that 'modify_r' and 'modify' are the winning methods. It can also be noted that 'cluster_r' always has a better score than 'cluster'. For the three adversary queries (first three rows of Table 4), the modification and restricted methods gave the lowest risk. Hence it can be concluded that even though the perturbation is restricted, the constrained methods have the lowest risk.

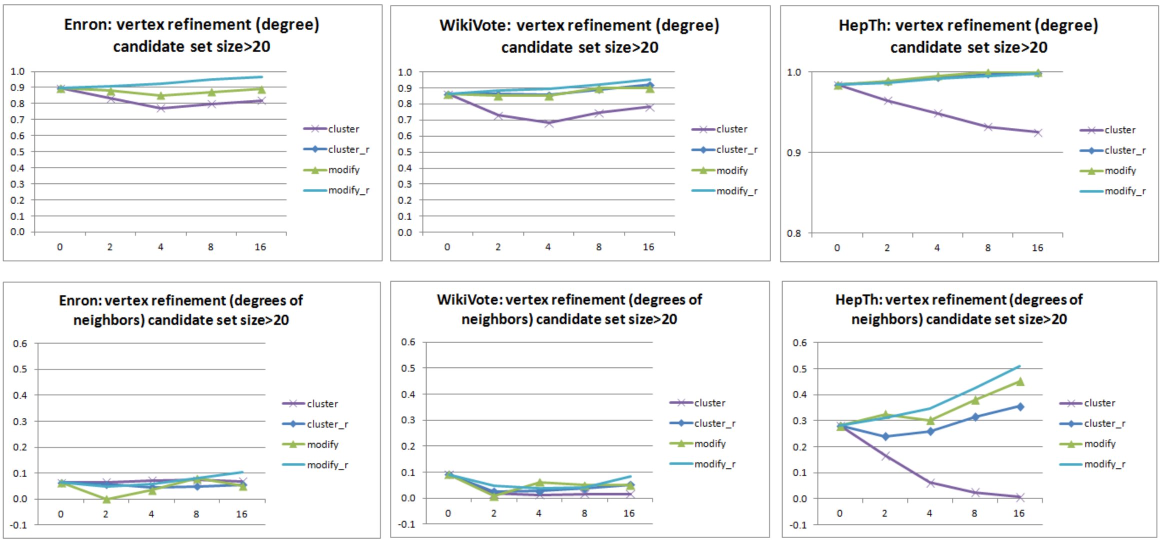

Fig 3: Risk, Adversary Queries on degree (vertex refinement H1(x)) and degrees of neighbors (vertex refinement H2(x)). Effect of anonymization on risk for candidate sets '>20' (degrees and degrees of neighbors) for three different datasets. The figures show the percentage of nodes (y-axis) in the candidate set for different values of k (x-axis).

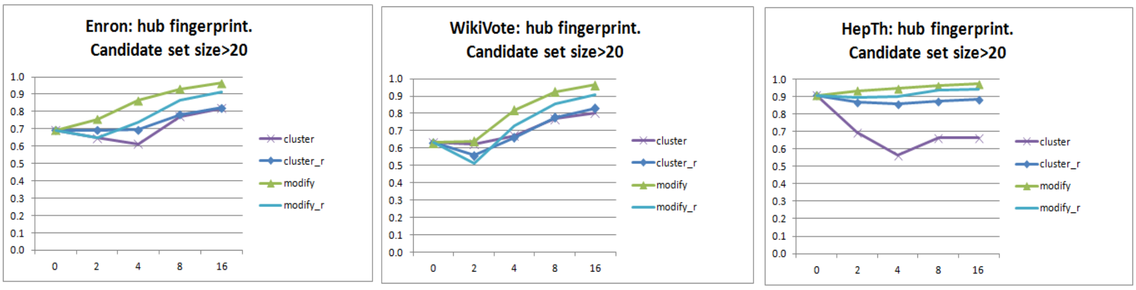

Fig 4: Risk, Hub Fingerprint with visibility horizon of 2. Effect of anonymization on risk for candidate set '>20', for three different datasets. The figures show the percentage of nodes (y-axis) in the candidate set for different values of k (x-axis).

Table 4. Risk: relative performance of methods by adversary query

	cluster_r	modify	cluster	modify_r
$H_1(x)$	1.7	1.7	2.7	1.0
$H_2(x)$	3.0	2.7	3.3	1.0
$F_2(x, H)$	3.3	1.0	3.67	2.0
Rank	3	2	4	1

Table 5. Risk: relative performance of methods by dataset

	cluster_r	modify	cluster	modify_r
Enron	2.7	2.3	3.0	1.3
WikiVote	3.0	1.7	3.3	1.3
HepTh	2.3	1.3	3.3	1.7
Rank	3	2	4	1

Table 6. Graph characteristics vs. best performing methods

	Enron	HepTh	WikiVote
Dataset character-istics	High D, high CC, low NC, v. high ACS	High D, low CC, low NC, high ACS	Low D, high CC, high NC, low ACS
Risk minimization	'modify_r' (H_1, H_2)	'modify' (F_2)	'modify_r' (H_1, H_2)
Information loss minimization	'modify_r' (Hub); 'modify' (D, CC); 'cluster_r' (NC)	'modify_r' (Hub); 'modify' (D, CC); 'cluster_r' (NC)	'modify_r' (Hub); 'modify' (D, CC); 'cluster_r' (NC)

D=degree; CC=Clustering Coefficient; NC=number of communities; ACS=average community size; H_1, H_2 and F_2 are the adversary queries

7. SUMMARY AND CONCLUSIONS

The node modification and node clustering methods for graph perturbation have been implemented with specific constraints (on community structure and hubs) which mitigate the effect of the perturbation and hence the information loss. It has been seen that the constrained methods have not incurred an increase in risk with respect to the non-constrained methods.

In terms of information loss, the best method varies depending on the metric used, and in some cases on the dataset characteristics. The 'modify' (unrestricted) method was best for the degree and clustering coefficient metrics, for all datasets. 'modify_r' was best for the Hub metric and ‘cluster_r’ was best for the ‘NC’ (number of communities) measure. Overall, ‘modify_r’ and ‘modify’ gave the lowest overall information loss. A general summary can be seen in Table 6.

8. ACKNOWLEDGMENTS

This research is partially supported by the Spanish MEC (projects ARES CONSOLIDER INGENIO 2010 CSD2007-00004 -- eAEGIS TSI2007-65406-C03-02 -- and HIPERGRAPH TIN2009-14560-C03-01).

9. REFERENCES

[1] 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 (SESSION: Privacy and authentication) Vol. 1 , Issue 1, pages 102-114.

[2] Zhou, B., Pei, J. 2008. Preserving Privacy in Social Networks against Neighborhood Attacks, IEEE 24th International Conference on Data Engineering (ICDE), 2008, pp. 506 - 515.

[3] Wondracek, G., Holz, T., Kirda, E., Kruegel, C. 2010. A Practical Attack to De-Anonymize Social Network Users, Proc. of the 2010 IEEE Symp. on Security and Privacy, pp. 223-238.

[4] Backstrom, L., Dwork, C. and Kleinberg, J. 2007. Wherefore Art Thou R3579X? Anonymized Social Networks, Hidden Patterns, and Structural Steganography, WWW '07, Proc. 16th Int. Conf. on WWW, pp. 181 - 190, ACM, NY, USA, 2007.

[5] Cheng, J., Fu, A. W. C. and Liu, J. 2010. K–Isomorphism: Privacy Preserving Network Publication against Structural Attacks, Proc. ACM SIGMOD Int. Conf. on Mgt. of Data, pp. 459-470.

[6] Sweeney, L. 2002. k–anonymity: a model for protecting privacy, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (IJUFKS). Vol. 10, Issue: 5(2002) pp. 557-570.

[7] De Capitani di Vimercati, S., Foresti, S., Livraga, G. and Samarati, P. 2012. Data Privacy: Definitions and Techniques, Int. J. of Uncert., Fuzziness and Know. Based Sys., Vol. 20, Iss. 6, Dec. 2012, p. 793.

[8] Zhou, B., Pei, J. 2011. The k–anonymity and l–diversity approaches for privacy preservation in social networks against neighborhood Attacks, Know. and Inf. Sys., July 2011, Vol. 28, Iss. 1, pp 47-77.

[9] Nettleton, D.F., Sáez-Trumper, D., Torra, V. 2011. A Comparison of Two Different Types of Online Social Network from a Data Privacy Perspective, Proc. MDAI 2011. LNAI, Vol. 6820, pp. 223-234.

[10] Hay, M., Miklau, G., Jensen, D., Weis P. and Srivastava, S. 2007. Anonymizing Social Networks, SCIENCE Technical Report 07-19 (2007) pp. 107--3, Vol. 245.

[11] Skarkala, M.E., Maragoudakis, M., Gritzalis, S., Mitrou, L., Toivonen, H., and Moen, P. 2012. Privacy Preservation by k–Anonymization of Weighted Social Networks, ASONAM, page 423-428. IEEE Computer Society.

[12] Nettleton, D.F. 2012. Information Loss Evaluation based on Fuzzy and Crisp Clustering of Graph Statistics. Proc. WCCI 2012, World Congress on Comp. Intelligence 2012, FUZZ-IEEE, pp. 1-8.

[13] Bonchi, F., Gionis A. and Tassa, T. 2011. Identity Obfuscation in Graphs Through the Information Theoretic Lens, ICDE '11, Proceedings of the 2011 IEEE 27th Int. Conf. on Data Engineering, pp. 924-935, IEEE Computer Society Washington, DC, USA.

[14] Ying, X. and Wu, X. 2008. Randomizing Social Networks: a Spectrum Preserving Approach, In Proc. SIAM Int. Conf. on Data Mining (2008), pp. 739-750.

[15] Kleinberg, J. 1999. Authoratitive sources in a hyperlinked environment, Journal of the ACM, 46(5):604-632.

[16] Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebure, E. 2008. Fast unfolding of communities in large networks, in Journal of Statistical Mechanics: Theory and Experiment (10), 2008, pp. 1000.

[17] Nettleton, D.F. and Dries, A. 2013. Local Neighbourhood Sub-Graph Matching Method, European Patent application number: 13382308.8.

[18] Bastian, M., Heymann, S. and Jacomy, M. 2009. Gephi: An Open Source Software for Exploring and Manipulating Networks, in: Proc. 3rd. Int. Conf. on Weblogs and Social Media, 2009, pp. 361–362.

[19] Leskovec, J., Kleinberg, J., Faloutsos, C. 2007. Graph Evolution: Densification and Shrinking Diameters, ACM Transactions on Knowledge Discovery from Data (ACM TKDD), 1(1).

[20] Shetty, J. and Adibi, J. 2005. Discovering Important Nodes through Graph Entropy - The Case of Enron Email Database, KDD ’2005, Chicago, Illinois.

[21] Leskovec, J., Huttenlocher, D. and Kleinberg, J. 2010. Signed Networks in Social Media, CHI '10 Proc. 28th Int. Conf. on Human Factors in Computing Systems, pp. 1361-1370, ACM, NY, USA.