Protecting the Sensitive Information on Online Social Networks

N.Anjaiah[1] & CH.Ravi[2]

1. PG Scholar, TKR College of Engineering and Technology, Hyderabad, Telangana, India
Email: anji.n502@gmail.com

2. Associate. Professor, TKR College of Engineering and Technology, Hyderabad, Telangana, India

ABSTRACT—
This paper stimulated by recognition of the need for a finer grain and more personalized privacy in publication of data in social networks. We introduce a privacy protection method for preventing the disclosure of identity of users but also the disclosure of selected features in users profiles .an individual user can select which of selected features of his profile he wishes to be secret. The social networks are modelled as graphs in which users are represented by nodes and features are represented by labels. Labels or features are denoted either as sensitive or as non sensitive. We treat node labels both as background knowledge an adversary may possess, and as sensitive information that has to be protected. We present privacy protection method that allow for graph information to be published in a form such that an opponent who possesses information about a node's neighbourhood cannot safely suppose its identity and its sensitive labelled information. For this, the algorithms convert the original graph into a graph in which nodes are sufficiently identical. The algorithms are designed to do so while losing as little amount of information and while preserving utility as much as possible. We evaluate empirically the extent to which the algorithms preserve the original graph's properties and structure. We show that our algorithm is efficient, effective, and scalable to offer stronger privacy assurances than those in previous study.

KEYWORDS— Protecting the Private data, labeled edges, clustering the nodes, GSINN algorithm, More Efficiency.

Introduction

The publication of data in social networks entails a privacy threat for their users. Sensitive data about users of the online social networks should be protected. The challenge is to develop methods to publishing the social network data in a form that affords effectiveness without compromising privacy protection. Earlier research has proposed various privacy methods with the corresponding protection schemes that prevent both accidental private information leakage and attacks by malicious adversaries. These early privacy methods are mostly concerned with identity of the node and link disclosure. These social networks can be modelled as a graph in which users are represented by nodes and social connection features are edges. The threat definitions and protection algorithms control structural properties of the graph. This paper is motivated by the recognition of the need for a fine grained and personalized privacy protection. Users entrust social networks such as

Facebook and twitter with a possessions of personal information such as their date of birth, address, home location and various opinions.

We refer to these personal information and messages as features in the user's profile. We propose a privacy protection method that can be prevents the disclosure of identity of users and also the selected features in users' profiles. An individual user can select which features of his profile he wishes to secrete. The online social networks are modeled as graphs in that users are nodes and features are labels1. Labels are denoted by either as sensitive label or as non-sensitive label. Figure 1 is a labeled graph representing a small subset of nodes such a online social network. Each node in the graph represents a different user, and the edge between two nodes represents the fact that the two persons are friends hence they are authorized to see sensitive data. Labels annotated to the nodes show the home locations of users. Each letter represents a home town name as a label for each node. Some individuals do not mind their residence being known by the other peoples, but some do, for different reasons. In these cases, the privacy of their labels should be protected at data publication. Therefore the locations are labeled as either sensitive or non-sensitive.

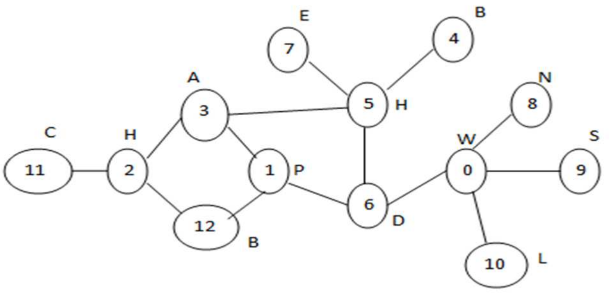

Fig.1 Labeled graph representing social network

The privacy issue arises from the discovery of sensitive labels. One might suggest that such labels should be just deleted. Still, such a solution would present an incomplete view of the network and may conceal interesting statistical information that does not make threats privacy. A more sophisticated approach consists in releasing the data about sensitive labels, while ensuring that the identification of users are protected from privacy threats. We consider such threats as neighborhood attack, in which an opponent finds out sensitive information based on prior knowledge of the number of neighbors of a node and the labels of neighbors. In the example, if an opponent knows that a user has four friends and that these friends are in A (America), B (Brazil) and C (Cape town), D(Durban), respectively, then he can infer that the user is in H (Helsinki). We present privacy protection algorithms that allow for graph to publish the data in a form such that an opponent cannot safely infer the identity and sensitive information labels of users.

In this case we consider in which the adversary possesses both structural information and label data. These algorithms that we propose convert the original graph into a graph in which any node with a sensitive label is identical from at least k-1 other nodes. The possibility to infer that any node has a certain sensitive label (we can call such nodes as sensitive nodes) is no larger than 1/k For this purpose we design k-diversity-like model, where we treat node labels as both part of an adversary's background knowledge and as private information that has to be protected.

The algorithms are designed to provide privacy protection while losing as little amount of information and while preserving utility as much as possible. In view of the trade off between data privacy and utility, we evaluate empirically the extent to which the algorithms preserve the original graph's structure and properties such as density, degree distribution and clustering or grouping coefficient. We show that our solution is effective, efficient and scalable while offering stronger confidentiality guarantees than those in earlier research, and that our algorithms scale well as data size increasing.

II. RELATED WORK

In the first necessary anonymization approach in both the contexts of micro and network data consists in removing of identification. This nave method has been recognized quickly as fault to protect privacy. For micro data, Sweeney al. propose k-anonymity to get out of possible identification disclosure in idealistically anonymized micro data. K-diversity is proposed in order to further prevent feature disclosure. Similarly for network data, Backstrom et al, shows that naive anonymization is inadequate as the structure of the released graph may reveal the identity of the individual nodes corresponding to the nodes. Hay et al, highlighted this problem and quantify the risk of re-identification by adversaries with external information that is dignified into structural. Recognizing the problem, several works [5, 11, 18, 20-22, 24, 27, 8, 4, 6] proposed approach that can be applied to the naive anonymized graph, further altering the graph in order to provide certain privacy.

These works are based on graph models other than simple graph [12, 7, 10, 3]. To our knowledge, Zhou and Pei [25, 26] and Yuan et al. [23] were the first to believe modelling networks as labeled graphs, correspondingly to what we consider in this paper. To prevent re-identification attacks by adversaries with immediate neighborhood information like structural knowledge, Zhou and Pei suggest a method that groups nodes and anonymized the neighborhood nodes in the same group by generalizing node labels and adding the edges. They implement a k-anonymity privacy constraint on the graph, each node of which is guaranteed to have the same immediate neighborhood information structure with other k-1 nodes. In [26], they improve the privacy guarantee provided by k-anonymity with the idea of k-diversity, to protect labels on nodes aswell. Yuan et al. [23] try to be more practical by in view of users' different privacy concerns. They divide privacy requirements into three levels, and suggest methods to generalize labels and adjust structure corresponding to every privacy demand. Nevertheless, neither Zhou and Pei, nor Yuan et al. believe labels as a part of the background knowledge. However, in this case adversaries hold label information, the methods of [25, 26, 23] cannot achieved the same privacy. besides, as with the context of microdata, a graph that satisfies a k-anonymity privacy guarantee may still leakage of sensitive information regarding as its labels .

III. PROBLEM DEFINITION

We model a network as G(V;E;Ls;L;T), where V represents a set of nodes or users, E represents a set of edges, L s is a set of sensitive labels, and L is a set of non-sensitive labels. T maps nodes to their labels, T: V→ Ls U L. Then we suggest a privacy model, k-sensitive-label-diversity; in this model, we treat node and labels both as part of an opponent's background knowledge, and as sensitive or private information that has to be protected. These concepts can be clarified by the following two definitions:

Definition 1. The node v comprises the neighbourhood information of node v and the degree of v and the labels of v's neighbors.

Definition 2. (K-sensitive-label-diversity) A sensitive label is associated with For each node v , there must be at least ` k-1 other nodes with the same neighborhood private information, but attached with different sensitive labels.

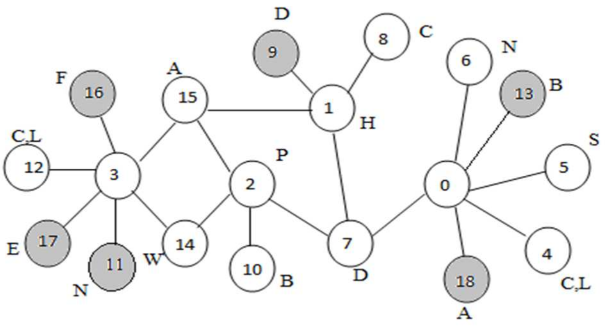

Fig. 2. Privacy-attaining in Social Network

For Example, nodes 0, 1, and 3 have sensitive labels with secrete information. The neighborhood information of node 0, includes its degree, which is four, and the labels on nodes 5, 4,6, and 7, which are C,L, S, N, and D, correspondingly. For example node 2, the neighborhood information includes degree 4 and the labels on nodes 7, 10, 14, and 15, which are D,W, A, and B. In that graph Figure 2 satisfies 2-sensitive-label-diversity because, in that graph, node 0 and node 3 are indistinguishable, having six neighbors with label A, B, {C,L}, D,N,S, separately; likewise, nodes 1 and 2 are identical, as they both have four neighbors with labels B, C, D and A separately.

IV. ALGORITHM

The main aim of this algorithms that we propose is to make appropriate clustering of nodes, and appropriate changing the neighbors' labels of nodes of each group to satisfy the k-sensitive-label-diversity condition. We want to cluster nodes with as similar neighborhood information as possible so that we can modify as some labels as possible and add some noisy nodes as possible as. We propose an algorithm, G0lobal-similarity-based Indirect Noise Node (GSINN) that does not attempt to heuristically reduce the similarity calculation as the other two algorithms, Direct Noisy Node Algorithm (DNNA) and Indirect Noisy Node Algorithm (INNA) to do. Algorithm DNNA and INNA, which we formulate first, sort nodes by degree and contrast neighborhood information of nodes with similar degree.

4.1 Algorithm GSINN

This algorithm starts out with formation of group formation, during which all nodes that have not yet been grouped are taken into consideration, in grouping-like approach. In the first running of algorithm, two nodes with the maximum similarity of their neighborhood labels are cluster together. Their neighbor labels are personalized to be the same immediately so that nodes in one group always have the equivalent neighbor labels. For two nodes, v1 with neighborhood label set (LSv1), and v2 with neighborhood label set (LSv2), Now we can calculate neighborhood label similarity (NLS) as follows:

$$\text{NLS}(v1;\, v2) = \frac{|\,\text{LSv1} \cap \text{LSv2}\,|}{|\,\text{LSv2} \cup \text{LSv1}\,|}$$

Larger value indicates larger similarity of the two neighborhoods. Then nodes having the maximum similarity with any node in the group are clustered into the group till the cluster has k nodes with dissimilar sensitive labels. There after, the algorithm proceed to create the next cluster. If fewer than k nodes are left after the last cluster formation, these remainder nodes are grouped into existing groups according to the similarity between different nodes and groups. After having formed these groups, we need to ensure that each group's members are indistinguishable in terms of neighborhood private information. Thus, neighborhood labels are changed after every clustering operation, so that labels of nodes can be accordingly updated immediately for the next clustering operation. This modification process ensures that all nodes in a cluster that have same neighborhood information. The purpose is achieved by a series of adjustment operations. To change graph with as low information loss as possible, we devise three alteration operations these are: label union, inserting edge and addition of noise node. Label union and inserting edge among nearby nodes are preferred to node addition, as they incur less modification to the overall graph structure. Inserting edge is to complement for both an absent label and inadequate degree value. A node is linked to an existing nearby node with that sensitive label. These Label union adds the missing label values by create super-values common among labels of nodes.

Labels of the two or more nodes combine their values to a single super-label value, being the combination of their values. This approach maintains data integrity, in the sense that the true label of node is included among the values of its label super-value. After such edge insertion and label union operations, if there are nodes in a group still having dissimilar neighborhood information, noise nodes with non-sensitive labels are added into the graph so as to provide the nodes in group identical in terms of their neighbors' labels. We consider the association of two nodes' neighborhood labels as an example.

Global-Similarity based indirect Noise Node Algorithm

Input: graph G(V,E,L,Ls), parameter k;
Result: Modified Graph G'

```
while $V_{left} > 0$ do
    if $|V_{left}| \geq k$ then
        calculate pair wise node similarities;
        group G ← ($v_1$, $v_2$) with $Max\_{similarity}$;
        Adjust neighbors of graph G;
        while $|G| < k$ do
            dissimilarity($V_{left}$; G);
            group G ← v with $Max\_{similarity}$;
            Alter neighbors of G without adding noisy nodes ;
    else if $|V_{left}| < k$ then
        for each v € $V_{left}$ do
            similarity(v; Gs);
            $G_{Max\_similarity}$ ←v;
        Change neighbors of $G_{Max\_ similarity}$ without adding
        noisy nodes;
Add predictable noisy nodes;
Return $G'(V' ;E' ;L' )$;
```

One node may need a noisy node to be added as its instant neighbor since it does not have a neighbor with assured label that the other node has; such a label on the other node may not be changeable, as its is already connected to another sensitive label node, which prevents the re-changing on existing modified groups. In this algorithm, adding noise node operation that is expected to make the nodes inside each group satisfy k-sensitive-label-diversity are recorded, but not performed right away. Only after all the preliminary clustering operation are perform, the algorithm proceeds to procedure the predictable node addition operation at the last step. Then, if two nodes are expected to have the same labels of neighbors and are within two hops (having common neighbor information), only one node is added. In other words, we combine few noisy nodes with the same label, thus results in fewer noisy nodes.

V. EXPERIMENTAL EVALUATION

We estimate our approaches using both artificial and real data sets. All of the approach have been implemented in Python. The experiments are conducted on an Intel core, 2Quad CPU, 2:83GHz machine with 4GB of main memory running Windows 7 OS. We are using three data sets. The first data set is a network of hyperlinks between weblogs on US politics. And the second data set that we are using is generated from the Face book dataset. And the third data set that we use is a family of artificial graphs with unstable number of nodes. The first and second datasets are used for the estimation of effectiveness (data utility and information loss). The third data set is used to measure running time and scalability.

5.1. Data Utility

We can compare the data utilities we maintain from the original graphs, in view of size on degree distribution, label distribution, degree centrality, grouping, coefficient, average node path length and graph density. We show the number of the noisy nodes and edges needed for each approach.

0.1

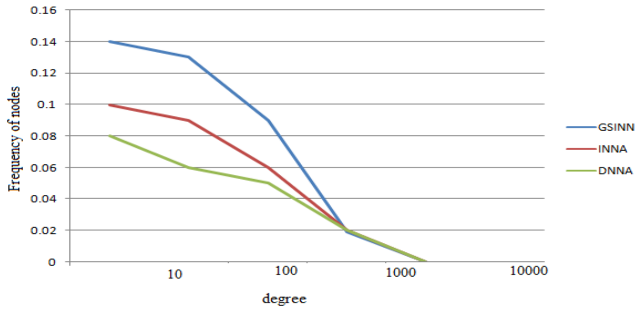

Fig. 3. Facebook Graph Degree Distribution

Above Figure shows the degree distribution of the node in Facebook graph both before and after modification of labels. Each subfigure in Figure 3 shows degree distributions of graphs personalized by one algorithm. We can see that the degree distributions of the changed graphs look like the original ones well, particularly when l is small. To sum up, these measurements shows the graph structure properties are preserved to a large degree. The strong similarity of the label distributions in most cases indicates that the sensitive label information, another aspect of graph label information, is well maintained. They suggest as well that algorithm GSINN does maintain graph properties better than the other two while these three algorithms complete the same privacy constraint.

5.2.Information Loss

In the view of data utility and releasing of data, we aim to keep information loss low. Information loss in this case contains both structure information loss as well as label information loss. We can measure the loss of information in the following way: for any node v € V ,v the set of labels in the modified graph. Thus, for the customized graph including n noisy nodes, and m noisy edges, information loss is defined as

$$IL = w1n + w2m + (1-w1-w2)\sum D(lv1; lv2)$$

where w1 , w2 and 1-w1-w2 are weights nodes for each part of the information loss. Figure 4 shows the measurements of information loss on the artificial data set using each algorithm. Algorithm GSINN introduces the least information loss in the network.

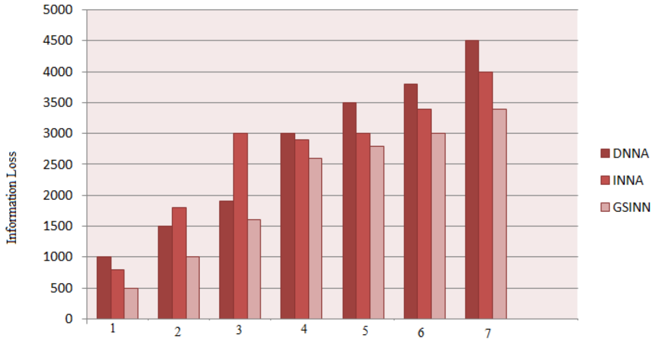

Fig. 4 Information Loss

5.3 Algorithm Scalability

We measure the runtime of the methods for a series of artificial graphs with varying different number of nodes in our third dataset. Figure 5 presents the runtime of each algorithm as the number of nodes increasing order. Algorithm DNNA is faster than the other two algorithms, showing good scalability at the cost of large number of noisy nodes added. Algorithm GSINN can also be adopted for reasonably large graphs as follows: We divide the nodes to two different categories, with or without sensitive labels. Such smaller granularity reduces the number of nodes the anonymization method needs to process, and thus it improves the overall efficiency.

VI. CONCLUSION AND FUTUREWORK

We have investigated in this paper the protection of private label sensitive information in publication of social network. We can consider graphs with rich sensitive label information, which are organized to be either sensitive label or non-sensitive label. We assume that adversaries have prior knowledge about a every node's degree and the labels of its neighboring node, and can use that to infer the sensitive labels of targets. We recommend a model for attaining privacy while publishing the data in social networks, in which node labels are both part of adversaries' background knowledge and sensitive label information that has to be protected. We accompany our model with algorithms that convert a network graph before publishing data, so as to limit adversaries' assurance about sensitive label data. Our experiments on both actual and artificial data sets confirm the effectiveness, efficiency and scalability of our technique in maintaining significant graph properties while providing a comprehensible confidentiality guarantee.

REFERENCES

[1]. L. A.Adamic and N. Glance. The political blogosphere and the 2004 U.S. election. divided they blog. In LinkKDD, 2005.

[2]. L. Backstrom, C. Dwork, and J. M. Kleinberg.Wherefore art thou R3579X?

Anonymized social networks, hidden patterns, and structural steganography. Com-mun. ACM, 54(12), 2011.

[3]. S. Bhagat, G. Cormode, B.Krishnamurthy, and D. S. and. Class-based graph anonymization for social network data. PVLDB, 2(1), 2009.

[4]. A. Campan and T. M.Truta. A clustering approach for data and structural anonymity in social networks. In PinKDD, 2008.

[5]. J. Cheng, A.W.-C. Fu, and J. Liu. K-isomorphism: privacy-preserving network publication against structural attacks. In SIGMOD, 2010.

[6]. G. Cormode, D.Srivastava, T.Yu, and Q. Zhang. Anonymizing bipartite graph data using safe groupings. PVLDB, 19(1), 2010.

[7]. S. Das O. Egecioglu, and A. E. Abbadi. Anonymizing weighted social network graphs. In ICDE, 2010.

[8]. A. G. Francesco Bonchi and T.Tassa. Identity obfuscation in graphs through the information theoretic lens. In ICDE, 2011.

[9]. M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis. Resisting structural re-identification in anonymized social networks. PVLDB, 1(1), 2008.

[10]. Y.Li and H.Shen. Anonymizing graphs against weight-based attacks. In ICDM Workshops, 2010.

[11]. K. Liu and E. Terzi. Towards identity anonymization on graphs. In SIGMOD, 2008.

[12]. L. Liu, J.Wang, J. Liu, and J. Zhang. Privacy preserving in social networks against sensitive edge disclosure. In SIAM International Conference on Data Mining, 2009.

[13].A.Machanavajjhala, J.Gehrke, D. Kifer, and M. Venkita subramaniam. Diversity privacy beyond k-anonymity. In ICDE, 2006.

[14]. MPI. http://socialnetworks.mpi-sws.org/.

[15]. Y. Song, P. Karras, Q.Xiao, and S.Bressan. Sensitive label privacy protection onsocial network data. Technical report TRD3/12, 2012.

[16]. Y. Song, S. Nobari, X. Lu, P.Karras, and S. Bressan. On the privacy and utilityof anonymized social networks. In iiWAS, pages 246{253, 2011.

[17]. L.Sweeney. K-anonymity: a model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems.

[18]. C.-H. Tai, P. S.Yu, D.-N. Yang, and M.-S. Chen. Privacy-preserving social network publication against friendship attacks. In SIGKDD, 2011.

[19]. O. Tore, A. Filip, and S.John. Node centrality in weighted networks: generalizing degree and shortest paths. Social Networks, 32(3), 2010.

20. W. Wu, Y. Xiao, W. Wang, Z. He, and Z.Wang. K-symmetry model for identity anonymization in social networks. In EDBT, 2010.

[21]. X. Ying and X.Wu. Randomizing social networks: a spectrum preserving approach. In SDM, 2008.

[22]. X.Ying and X. Wu. On link privacy in randomizing social networks. In PAKDD,2009.

[23]. M. Yuan, L. Chen, and P. S. Yu. Personalized privacy protection in social networks. PVLDB, 4(2), 2010.

[24]. L. Zhang and W.Zhang. Edge anonymity in social network graphs. In CSE, 2009.

[25]. B. Zhou and J. Pei. Preserving privacy in social networks against neighborhood attacks. In ICDE, 2008.