Research Paper

Volume : 3 | Issue : 5 | May 2014 • ISSN No 2277 - 8179

Privacy in Social Network Data by Anonymization and Protection of Sensitive Labels

Engineering

KEYWORDS :

Arathy G — PG Scholar, Department of ComputerScience and Engineering, Vedavyasa Institute of Technology, University of Calicut, Kerala, India.

S.Kavitha Murugesan — HOD Department of ComputerScience and Engineering, Vedavyasa Institute of Technology, University of Calicut, Kerala, India.

ABSTRACT

Privacy is one important factor to achieve privacy for social science research and business analysis. The problem occurs for publishing and sharing social network data, where privacy is a major concern. Many models has been developed to prevent node reidentification through structure information ,but still an attacker may be able to infer ones private information if a group of nodes largely share the same sensitive labels. The problem with existing approach is that there are chances which alter the graph properties-degree L-diversity model is defined here. The k-anonymity privacy requirement for publishing microdata requires that each equivalence class (i.e., a set of records that are indistinguishable from each other with respect to certain "identifying" attributes) contains at least k records. Recently, several authors have recognized that k-anonymity cannot prevent attribute disclosure. The notion of diversity has been Proposed to address this diversity requires that each equivalence class has at least well-represented values for each sensitive attribute. This method considers protection of structural information and sensitive labels of individuals. The methodology is based on adding noise nodes. Here a new algorithm is defined, in which noise nodes are added to the original graph and this is done with least distortion to graph properties. In order to evaluate the effectiveness of proposed technique extensive experiments are done.

I. INTRODUCTION

Networks are graphical structures with a set of nodes connected by a set of links. A social network represents social entities as a node and the links between them may describe friendship, kinship, same taste of music or some kind of interaction. An information network is another example of a network structure where the nodes represent scientific publications and their links represent citation. Other such examples are internet networks where nodes represent computers and edges represent communication, signed networks where the nodes represent users of social networking site and edges represent positive and negative values indicating friendship or enmity and so forth.

But in reality, social networks may not be simple. It may be classified as directed/undirected, weighted/unweighted, labelled/unlabeled edges etc. For example, a financial transaction network is directed, where a directed edge is provided if a node (user having bank account) transfers money to another node where the reverse is meaningless. Road networks have intersections of roads as nodes and roads themselves as edges. An edge can be given a weight denoting the distance. This is an example of weighted network.

Such social networks are of great importance to researchers from various fields such as sociology, psychology, market research, or epidemiology. But, the social network data cannot be provided to explorers as such, since it might reveal the identity of respondents. Therefore, it is necessary to anonymize the data before making it available to explorers to ensure the privacy of the individuals whose sensitive information is included in the data. Data anonymization should be such that it preserves the privacy of individuals and maintains the utility of the data for researchers [3]. Preserving privacy means that the identity of individuals should not be revealed using the knowledge of the attacker and maintaining utility means that the parameters like average shortest path length and clustering coefficient should be maintained with less difference between the original data and published data

Online social networks connect people from different parts of the world via internet. They also help in publishing their details online. These details are precious sources of data for researchers. But these data cannot be provided to the researchers as such, since it is against the Data Protection Act 1998 [4]. Thus, there is a need to anonymize the data before providing them to researchers. Anonymizing the data should be such that it does not reveal the identity of the users and it is useful for the researchers. But these two factors go hand in hand. This has led to the need of a work that satisfies both users and researchers

II. PROBLEM DESCRIPTION

Here we define a social nework data in the form of a graph as: Definition 1. Its is four tuple G(v,e,z,l) where v is a set of vertices and each vertex in it represents a node in the network. Z is set of labels that vertices have .

when a social network graph is being published the attacker can easily reidentify the nodes with degree information and reveal all the sensitive information .So here k-degree l-diversity principle is udes for those published graphs which have the same spirit of k-l diversity in rational data .

Definition 2:For each vertex in graph there exist atleast k-1 other vertices having same degree in graph and also the vertex having te same degree must contain l distinct sensitive labels.

Here l diversity [2] is specified to make sure that there is atleast l distinct sensitive labels in each group. The method of anonymization is demonstrated by using l diversity .A graph that satisfy the KDLD constraints is called a KDLD graph .The goal of this paper is to protect individuals privacy by adding edges/nodes to KDLD graph.The additional edges or nodes is called noise nodes/edges.

To design a KDLD graph we uses a two step algorithm .In the first step we compute target degree for each node, so that it satisfies the the KDLD constraints with a minimum degree change,and in the second step we change each nodes degree to te obtained target degree.this is done by adding noise nodes or edges.

Definition 3:the generated sensitive degree sequence is a sequence of n triples [T[1]...,T[n].whre T[i] is a triple(id,d,l) at position i in p,d is degree and l is the sensitive label with node id.

Inorder to solve any subproblem 2 steps are used respectively:

Kdld sequence generation and finally graph construction.

III. ARCHITECTURE

This section explains the architecture diagram and the module description of the system to be developed. The architecture diagram is as shown in Fig 3.1 below.

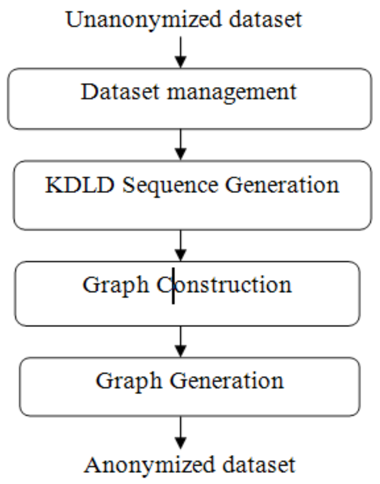

Fig 3.1: Architecture of the system

As shown in the architecture diagram of the system in Fig 3.1, the system takes as input the unanonymized dataset. The unanonymized dataset is converted into two tables: node table and edge table. The node table contains the details of the nodes in the input social network data and the edge table contains the details of the edges in the input social network data. These tables are used in the generation of the KDLD sequence which forms the basis for the conversion of the unanonymized dataset to anonymized dataset. The KDLD sequence is a representation of the target graph in the form of degree sequence. Using this sequence, the input graph is modified based on the algorithms specified to an anonymized graph. This is done in the graph construction step of the architecture. The outputs of the graph construction step are two tables modified from the node and edge table. These tables are to be converted to a graphical representation for publishing. This is done in the graph generation step.

IV. KDLD SEQUENCE GENERATION

This module collects the necessary information needed for the generation of the KDLD sequence and converts the input sequence to a KDLD sequence. Given a graph G, its sensitive degree sequence is a sequence of n triples: [P[1],...,P[n]] where P[1].d ≥ P[2].d ≥ ... ≥ P[n].d, P[i] is a triple (id, d, s) at position 'i' in P, 'd' is the degree, and 's' is the sensitive label associated with node 'id' [1]. A KDLD sequence is defined as follows. KDLD sequence: A sensitive degree sequence P is a KDLD sequence if P satisfies the following constraint: P can be divided into a group of subsequences $[[P[1],\ldots P[i_1]]$, $[P[i_1 + 1] ,\ldots P[i_2]]$, $[P[i_2+1]$, $\ldots,P[i_3]]$,...,$[[P[i_m+1]$,...,P[j]] such that for any subsequence $P_x = [P[i_x] ,\ldots,P[i_x+1]]$, satisfies three constraints:

1) All the elements in P_x share the same degree ($P[i_x].d = P[i_x + 1].d = \ldots = P[_{ix+1}].d$).
2) P_x has size at least k ($i_{x+1} - i_x + 1 \geq k$).
3) P_x's label set $\{P[t].s \mid i_x \leq t \leq i_{x+1}\}$ have at least l distinct values.

These constraints are together called the KDLD constraint. The graph obtained from the KDLD sequence is called the KDLD graph which is the anonymized output. The KDLD sequence is generated using the following algorithm:

Step 1: Represent each node as a triple containing node id, degree and sensitive label chosen by the user.

Step 2: Arrange the triples as a sequence.

Step 3: Sort the sequence based on degree of the nodes.

Step 4: From the KDLD sequence using KL-BASED algorithm.

The KL-BASED algorithm forms a group by satisfying k-degree constraint first followed by l-diversity constraint .

The algorithm is:
Step 1: Select the first 'k' elements of the sorted sequence as a group.

Step 2: Check whether the group contains atleast 'k' elements and atleast 'l' different sensitive labels.

a) If it satisfies both the conditions, goto step 3.
b) Else add the next element to the group and repeat Step 2.

Step 3: Check whether the next element can be added to the existing group or if it needs to form a new one. This is done by calculating two costs: C_{merge} and C_{new}. C_{merge} considers the current group including the next element and C_{new} considers a new group comprising the next 'k' elements. To calculate these costs, find the difference between the average of the degrees in the group and the degree of each element in the group and then adding them together.

a) If C_{merge} is small, add the next element to the current group and goto Step 3.
b) Else, if C_{new} is small, form the new group and goto Step 2. c)

V KDLD GRAPH CONSTRUCTION

After getting the new sensitive degree sequence P_{new}, a new graph G' should be constructed. The KDLD Graph Construction module does this in the following four steps:

1) Proceed without noise nodes.
2) Decrease degree with noise nodes.
3) Increase degree with noise nodes.
4) Matching the degree of the noise nodes.

These steps are discussed in detail in the following portions.

PROCEED WITHOUT NOISE NODES: If a part of the social network data satisfies any of the three cases mentioned below, the corresponding modifications are to be made in the node and edge table. Case 1: Node u needs to increase its degree, v needs to decrease its degree and u,v are direct neighbors. In this case, w is a randomly selected one direct neighbor of v which does not connect to u. Remove the edge (v,w) and add a new edge (u,w) (See Fig 4.2). By doing this, u's degree is increased by 1 and v's degree is decreased by 1 while all the other nodes' degrees remain unchanged. This operation changes the distance between u,w from 2 to 1 and the distance between v,w from 1 to 2. Therefore, only the paths passing through u,w or v,w change their lengths at most by 1

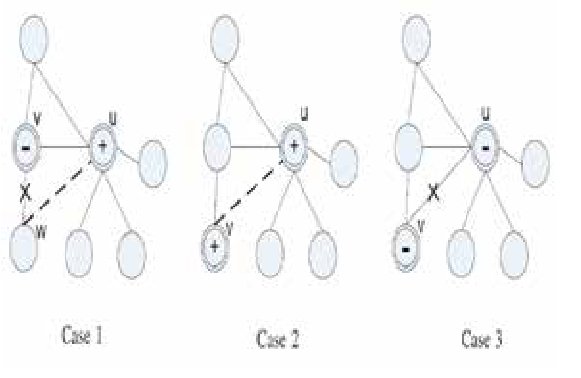

Fig 4.2: Case 1 to proceed without noise nodes [1]

Case 2: Two nodes u, v are two hop neighbors and they both need to increase their degree. If there does not exist link (u , v), add link (u , v) (See Fig 4.3). The distance between u , v is changed from 2 to 1. Only the lengths of shortest paths passing though u, and v change by 1.

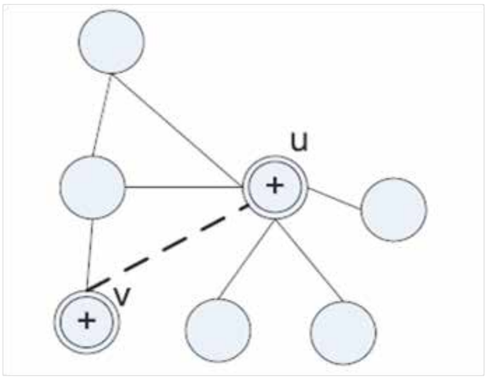

Fig 4.3: Case 2 to proceed without noise nodes [1]

Case 3: Two nodes u and v that both need to decrease their degrees are direct neighbors. If after removing link (u , v), u and v are still two hop neighbors, remove the link (u , v) as in

Fig 4.4. The distance between u , v is changed from 1 to 2. So it only changes the lengths of shortest paths passing through u , v at most by 1.

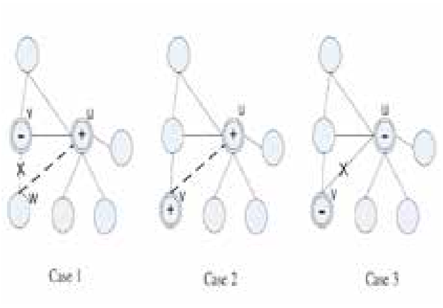

Fig 4.4: Case 3 to proceed without noise nodes [1]

DECREASE DEGREE WITH NOISE NODES: For any node u whose degree is still larger than its target degree after Step 1, its degree is decreased by following the following algorithm:

- Create a new node n and connect u with n.
- Choose a friend of u at random and provide its sensitive label to n.
- The target degree of n is the degree of a node which is less than and closest to current degree of u + 2 – target degree of u.
- Randomly select an edge (u , v) from the original graph, delete this edge, then add a new edge (n , v). By doing this, v only changes from u's one hop neighbor to its two hop neighbor. This random edge modification process is repeated until the degree of noise node reaches its target degree or u reaches its target degree as in Fig 4.5.

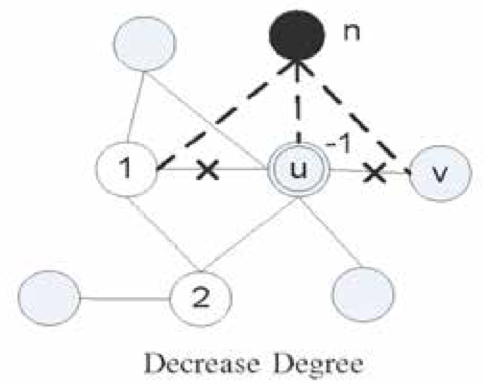

Fig 4.5: Decrease degree with noise nodes [1]

INCREASE DEGREE WITH NOISE NODES: For each vertex u in G which needs to increase its degree, the following algorithm is used to make its degree reach the target degree.

- Create a noise node n and connect u with n.
- Check whether there exists a node v within two hops of u, and v also needs to increase its degree (In Step 2, some noise nodes are added, such v may exist.), then connect n with v (See Fig 4.6). Since v is within two hops of u, connecting v with n will not change the distance between u and v.
- After this step, if n's degree is bigger than the minimum degree in KDLD sequence but does not appear in the sequence, recursively delete the last created link until the degree of n equals to a degree in the sequence.

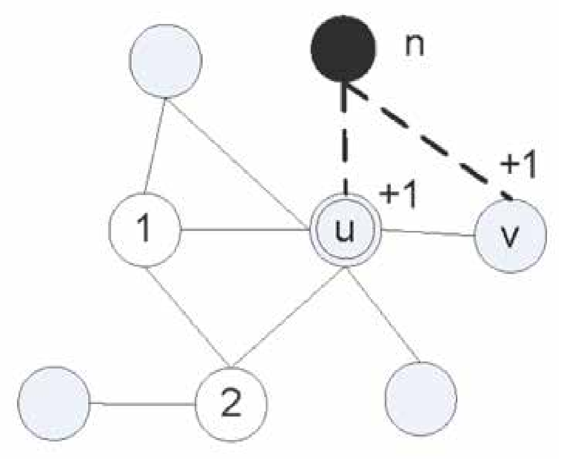

Fig 4.6: Increase degree with noise nodes [1].

MATCHING THE DEGREE OF THE NOISE NODES: In this step, the degree of each noise node is updated to a degree in KDLD sequence. This hides the noise node from the attacker as shown in example Fig 4.7. Furthermore, since it is assumed that an attacker use the degree information of some nodes to do the attack, if the noise nodes have degrees as the same as some original nodes, an attacker cannot distinguish them from the original nodes with the degree information. This is done by the following algorithm:

- Select a node whose degree is not in the KDLD sequence.
- If it is odd, choose an odd degree from the KDLD sequence just greater than it.
- If it is even, choose an even degree from the KDLD sequence just greater than it.
- Set this as the target degree of the noise node.
- Choose an edge (1,2) closest to the noise node as shown in Fig 4.7 and update the graph as shown in the figure below:

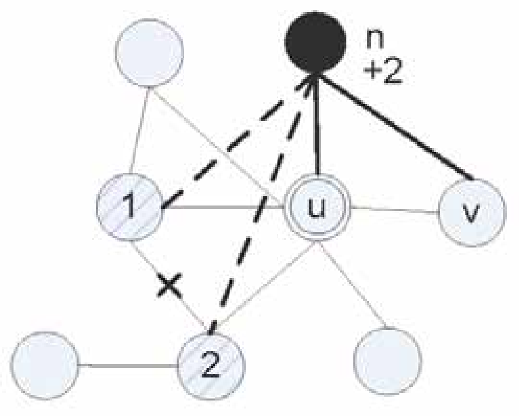

Fig 4.7: Matching the degree of the noise nodes [1].

All these steps basically include adding nodes and adding or deleting edges. These changes are updated in the node table and edge table respectively

VI GRAPH CONSTRUCTION

The node and edge table obtained from the previous step is converted to a dataset in xml format. This helps in displaying the social network data. This dataset is saved in the location of the input dataset. Graph# is a graph drawing tool implemented in C#. It is used to display the graph. The software tool takes as input the updated dataset as input and provides the graph. This is the final anonymized social network data.

VII CONCLUSION

Social networking sites such as MySpace, Facebook, Twitter, and Orkut have millions of registered users, and the resulting social graph structures have millions of vertices (users or social actors) and edges (social associations).Recent research has explored these social networks for understanding their structure, criminal intelligence, information discovery, advertising and marketing, and others. As a result, companies (such as Facebook) hosting the data are interested in publishing portions of the graphs so that independent entities can mine the wealth of information contained in these social graphs. Privacy-preserving publication of social networks is a very promising, challenging, and rapidly evolving research area. Ensuring privacy in this context still requires a number of issues to be carefully addressed. First, although there has been much progress in terms of designing algorithms to protect privacy in social network data, these algorithms generally does not preserve the graph properties like the distance between two nodes and graph utility. This calls for effective and efficient algorithms to protect social network data from identity and link disclosure. The KDLD algorithm is one such algorithm for preserving the privacy of the published social network data. But, it focuses only on passive attacks. As future work, the algorithm can be enhanced to overcome active attacks also.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support and facilities provided by Department of CSE, Vedavyasa Institute of Technology. Authors also extend their thanks to the Head of the Department Prof. Kavitha Murugesan for her immense help during the course of the project.

REFERENCE

[1] Mingxuan Yuan, Lei Chen, Member, IEEE, Philip S. Yu, Fellow, IEEE, and Ting Yu “Protecting Sensitive Labels in Social Network Data Anonymization” IEEE transactions on knowledge and data engineering, vol. 25, no. 3, march 2013. | [2] N. Li and T. Li, “T-Closeness: Privacy Beyond K-Anonymity and L-Diversity”, Proc. IEEE 23rd Int',l Conf. Data Eng. (ICDE ',07), pp. ,106-115, 2007. | [3] Campan, A., Truta, T.: Anonymization of centralized and distributed social networks by sequential clusterings. In: PinKDD (2008).. | [4] Data Protection in Research [online]. Available: http://www.canterbury.ac.uk/Research/ Documents/DataProtection.pdf. |