Privacy Preserving Social Graphs for High Precision Community Detection

Himel Dev *
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology, Dhaka, Bangladesh
himeldev@gmail.com

ABSTRACT

Discovering communities from a social network requires publishing the social network's data. However, community detection from raw data of a social network may reveal many sensitive information of the involved parties, e.g., how much a user is involved in which communities. An individual may not want to reveal such sensitive information. To resolve this issue, we address the problem of privacy preserving community detection in social networks. More specifically, we want to ensure that community detection is possible from the published social graph/data but the identity of users involved in a community should not be disclosed.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*

Keywords

Social Networks, Community Detection, Privacy

1. PROBLEM FORMULATION

Community Detection: The community detection in a social graph $G(V, E)$ involves grouping vertices into clusters $C = \{C_1, C_2, C_3, ...\}$, where C_i contains vertices from V that are closely related, and hence forms a community.

Privacy Preserving Community Detection: Our objective is to detect original communities from the published social graph, but the identity of the users involved in a community should not be disclosed.

2. RELATED WORK

Several anonymization methods have been proposed to battle the privacy attacks on social network data [5, 3, 4]. A popular class of these methods involves graph modification, i.e., anonymizing a graph by modifying (i.e, inserting and/or deleting) edges and vertices. However, none of these works strongly consider preserving *community structure* while developing privacy protection techniques. As a result, the developed graph modification techniques lead to indiscriminate modification of edges without focusing on the underlying community structure. Such indiscriminate modification of edges may disfigure the community structure and lead to misleading communities which highly deviate from the communities in the original social graph. Thus, state-of-the art graph modification approaches fail to serve our purpose of high precision community detection from privacy preserving social graphs.

*Under supervision of Dr. Mohammed Eunus Ali and Dr. Tanzima Hashem from CSE, BUET

SIGMOD/PODS'14, June 22–27, 2014, Snowbird, UT, USA.
ACM 978-1-4503-2376-5/14/06.
http://dx.doi.org/10.1145/2588555.2612668.

3. METHODOLOGY

Our solution is based on a probability graph, where each edge is assigned a probability denoting the likelihood of two users to belong to the same community. We use the likelihood information from the probability graph to construct a privacy-preserving version of the original social graph which is highly accurate in terms of community detection queries. In particular, we greedily modify the original social graph focusing on the community structure preservation, using the likelihood information from the probability graph.

Probability Graph: Given a weighted/un-weighted social graph G, we construct a *Probability Graph* (weighted) G_p, where each weight p_{uv} represents a probability between two vertices u and v to belong in the same community [1]. The vertices of G_p are similar to those in G, but the number of edges in G_p is higher. Note that, an edge with weight p_{uv} exists in G_p, when u and v are either directly connected or connected via one or more common neighbors in G.

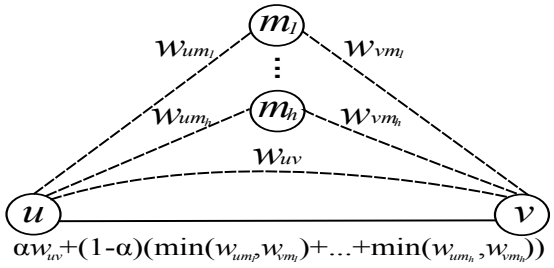

Figure 1: Probability (p_{uv}) Calculation

Let, the degree of interaction between two users u and v in the original social graph G be w_{uv}. Also assume that in

the original social graph G, u and v have h common neighbors (mutual friends) $\{m_1, m_2, ..., m_h\}$, and the degree of interaction of u and v with any common neighbor m_i be w_{um_i} and w_{vm_i} respectively. Then, in the probability graph G_p, we define the *probability* p_{uv} of two users u and v to belong to the same community as $p_{uv} = \alpha * w_{uv} + (1 - \alpha) * \sum_{i=1}^{h} min(w_{um_i}, w_{vm_i})$.

Privacy Preserving Social Graph: We construct privacy preserving version of social graph which can be used to identify communities analogous to the communities in the original social graph. We construct the privacy graph by greedily modifying the original graph using likelihood information from the probability graph. The key idea of this greedy modification is to replace the random edge deletion-addition scheme of state-of-the-art graph modification approaches with a biased random scheme that favors the existing inter-community edges during edge deletion and the non-existing inner-community edges during edge addition.

Biased Random Scheme: Under the biased random scheme, (i) an existing potential inter-community edge has higher probability of being deleted and (ii) a non-existing potential inner-community edge has higher probability of being added, during the modification of the social graph to construct a privacy preserving version of it. The scheme identifies the potential inter and inner community edges based on the likelihood information from the probability graph G_p. Note that, a high weighted edge in the probability graph is a potential inner-community edge, and a low weighted edge in the probability graph is a potential inter-community edge. The scheme works as follows:

1. Split the edge set of probability graph G_p into two disjoint sets E (analogous to the edge set in G) and E' (potential inner-community edges non-existent in G). Note that, each edge e_{uv} that belongs to one of these sets, has an associated probability value p_{uv}.

2. Create an edge set E^p for the privacy graph and initialize it with the edges from the set E.

3. (i) For each edge $e_{uv} \in E$, calculate its probability of being deleted from the privacy graph as: $\frac{f(1-p_{uv})}{\sum_{e_{ij} \in E} f(1-p_{ij})}$.
   (ii) For each edge $e_{uv} \in E'$, calculate its probability of being added to the privacy graph as: $\frac{f(p_{uv})}{\sum_{e_{ij} \in E'} f(p_{ij})}$.
   Here, $f()$ is a monotonic function which defines the degree of bias during privacy graph construction. For example, an exponential $f()$ function implies a highly biased graph modification technique, which is most likely to delete only the existing inter-community edges and add only the non-existing inner-community edges during edge modification. However, such a function makes the process relatively deterministic, which is not preferred in case of privacy preserving techniques.

4. Calculate the cumulative deletion/addition probabilities (cp) for edge set E/E'.

5. To delete/add an edge, (i) generate a random number $U(0, 1)$, (ii) if the number is within the range $cp_{(i-1)}$ and cp_i, delete/add the ith edge of the set E/E' from/to the edge set E^p. Note that, cp_0 is 1.

6. For consecutive m edge deletion or addition, repeat step 3 to step 5 for m times.

During edge addition, one can also allow edges that are not in the set E' by preserving their slot. For example, to probabilistically allow x% edges barring the set E', we need to calculate the addition probability of an edge $e_{uv} \in E'$ (associated with step 3) as: $(1 - \frac{x}{100}) * \frac{f(p_{uv})}{\sum_{e_{ij} \in E'} f(p_{ij})}$.

Further, for the edges outside E', we need to distribute the remaining probability $(\frac{x}{100})$ uniformly among the edges.

The biased random scheme can also be used with weighted graphs. In case of weighted graphs, we assign weights to the newly added edges by generating random numbers using the probability distribution corresponding to the current edge weights.

4. EXPERIMENTAL EVALUATION

We evaluate the performance of our proposed privacy graph construction method by comparing it with the state-of-the-art random m edge deletion-insertion method backed up by many approaches [2]. We compare these two methods in terms of the degree to which these privacy preserving methods preserve the underlying community structure of the original social graph. More specifically, for each of these privacy preserving methods, we determine the relevance/similarity of identified communities from the original social graph and the privacy graph constructed via the corresponding method. We use normalized mutual information (NMI) and pairwise F-measure (PWF) to compare the similarity of identified communities from the original social graph and corresponding privacy graph. Then, we compare the NMI and PWF values (degree of relevance) attained by each method to identify the superior method in terms of community structure preservation. We can see that, for two networks (Karate and Jazz) our algorithm achieves significantly higher NMI and PWF values compared to the competing random m edge deletion-insertion method.

Algorithm ↓	Karate		Jazz	
	NMI	PWF	NMI	PWF
Proposed	0.8283	0.8551	0.8643	0.8539
Random m del-add	0.4759	0.4454	0.6753	0.5257

5. REFERENCES

[1] H. Dev, M. E. Ali, and T. Hashem. User interaction based community detection in online social networks. In *Database Systems for Advanced Applications*, volume 8422, pages 296–310. 2014.

[2] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava. Anonymizing social networks. Technical Report 07-19, University of Massachusetts Amherst, March 2007.

[3] X. Wu, X. Ying, K. Liu, and L. Chen. A survey of privacy-preservation of graphs and social networks. In *Managing and Mining Graph Data*, volume 40, pages 421–453. 2010.

[4] E. Zheleva and L. Getoor. Privacy in social networks: A survey. In *Social Network Data Analytics*, pages 277–306. 2011.

[5] B. Zhou, J. Pei, and W. Luk. A brief survey on anonymization techniques for privacy preserving publishing of social network data. *SIGKDD Explor. Newsl.*, 10(2):12–22, December 2008.