Security Informatics
a SpringerOpen Journal

RESEARCH Open Access

Social network integration and analysis using a generalization and probabilistic approach for privacy preservation

Xuning Tang and Christopher C Yang*

Abstract

Social Network Analysis and Mining (SNAM) techniques have drawn significant attention in the recent years due to the popularity of online social media. With the advance of Web 2.0 and SNAM techniques, tools for aggregating, sharing, investigating, and visualizing social network data have been widely explored and developed. SNAM is effective in supporting intelligence and law enforcement force to identify suspects and extract communication patterns of terrorists or criminals. In our previous work, we have shown how social network analysis and visualization techniques are useful in discovering patterns of terrorist social networks. Attribute to the advance of SNAM techniques, relationships among social actors can be visualized through network structures explicitly and implicit patterns can be discovered automatically. Despite the advance of SNAM, the utility of a social network is highly affected by its d completeness. Missing edges or nodes in a social network will reduce the utility of the network. For example, SNAM techniques may not be able to detect groups of social actors if some of the relationships among these social actors are not available. Similarly, SNAM techniques may overestimate the distance between two social actors if some intermediate nodes or edges are missing. Unfortunately, it is common that an organization only have a partial social network due to its limited information sources. In public safety domain, each law enforcement unit has its own criminal social network constructed by the data available from the criminal intelligence and crime database but this network is only a part of the global criminal social network, which can be obtained by integrating criminal social networks from all law enforcement units. However, due to the privacy policy, law enforcement units are not allowed to share the sensitive information of their social network data. A naive and yet practical approach is anonymizing the social network data before publishing or sharing it. However, a modest privacy gains may reduce a substantial SNAM utility. It is a challenge to make a balance between privacy and utility in social network data sharing and integration. In order to share useful information among different organizations without violating the privacy policies and preserving sensitive information, we propose a generalization and probabilistic approach of social network integration in this paper. Particularly, we propose generalizing social networks to preserve privacy and integrating the probabilistic models of the shared information for SNAM. To preserve the identity of sensitive nodes in social network, a simple approach in the literature is removing all node identities. However, it only allows us to investigate of the structural properties of such anonymized social network, but the integration of multiple anonymized social networks will be impossible. To make a balance between privacy and utility, we introduce a social network integration framework which consists of three major steps: (i) constructing generalized sub-graph, (ii) creating generalized information for sharing, and (iii) social networks integration and analysis. We also propose two sub-graph generalization methods namely, edge betweenness based (EBB) and K-nearest neighbor (KNN). We evaluated the effectiveness of these algorithms on the Global Salafi Jihad terrorist social network.

* Correspondence: chris.yang@drexel.edu
College of Information Science and Technology, Drexel University, Philadelphia, USA

Introduction

Social Network Analysis and Mining (SNAM) techniques have drawn significant attention in the recent years due to the popularity of online social media. With the advance of Web 2.0 and SNAM techniques, tools for aggregating, sharing, investigating, and visualizing social network data have been widely explored and developed. SNAM is effective in supporting intelligence and law enforcement force to identify suspects and extract communication patterns of terrorists or criminals. In our previous work [1-3], we have shown how social network analysis and visualization techniques are useful in discovering patterns of terrorist social networks. Attribute to the advance of SNAM techniques, relationships among social actors can be visualized through network structures explicitly and implicit patterns can be discovered automatically.

Despite the advance of SNAM, the utility of a social network is highly affected by its d completeness. Missing edges or nodes in a social network will reduce the utility of the network. For example, SNAM techniques may not be able to detect groups of social actors if some of the relationships among these social actors are not available. Similarly, SNAM techniques may overestimate the distance between two social actors if some intermediate nodes or edges are missing. Unfortunately, it is common that an organization only have a partial social network due to its limited information sources. In public safety domain, each law enforcement unit has its own criminal social network constructed by the data available from the criminal intelligence and crime database but this network is only a part of the global criminal social network, which can be obtained by integrating criminal social networks from all law enforcement units. However, due to the privacy policy, law enforcement units are not allowed to share the sensitive information of their social network data. A naïve and yet practical approach is anonymizing the social network data before publishing or sharing it. However, a modest privacy gains may reduce a substantial SNAM utility. It is a challenge to make a balance between privacy and utility in social network data sharing and integration.

In order to share useful information among different organizations without violating the privacy policies and preserving sensitive information, we propose a generalization and probabilistic approach of social network integration in this paper. Particularly, we propose generalizing social networks to preserve privacy and integrating the probabilistic models of the shared information for SNAM. To preserve the identity of sensitive nodes in social network, a simple approach in the literature is removing all node identities. However, it only allows us to investigate of the structural properties of such anonymized social network, but the integration of multiple anonymized social networks will be impossible. To make a balance between privacy and utility, we introduce a social network integration framework which consists of three major steps: (i) constructing generalized sub-graph, (ii) creating generalized information for sharing, and (iii) social networks integration and analysis. We also propose two sub-graph generalization methods namely, edge betweenness based (EBB) and K-nearest neighbor (KNN). We evaluated the effectiveness of these algorithms on the Global Salafi Jihad terrorist social network.

This paper is organized as follows. In the next section, we review the existing works about privacy preservation of social network. Previous techniques are classified based on their assumption of attack models the definition of sensitive information, and the privacy preservation techniques. In section 3, we introduce the researchd framework. Social network generalization and integration techniques are introduced in section 4. The experiment design, results and discussions are presented in section 5. We conclude our work and introduce future work in section 6.

Related work

Sensitive information of social network

Given a social network, the definition of sensitive information depends on the specific applications. In the literature, the social network sensitive information can be classified into node properties, neighborhood graphs, edge properties, and network properties in general.

Node properties

In a social network, identity of nodes can be an important type of sensitive property [4-7]. A node with sensitive identity means that its identity is private and should not be released. On the other hands, a node with insensitive identity means that the identity of this node can be released with no harm. Another type of sensitive property of a node can be its degree centrality [8-12]. Given a node, the degree centrality equals to the total number of edges connecting to this node, which is the number of friend in a social network. In a directed graph, edges can be further divided into in-links and out-links. Releasing the degree centrality of a given node, attacker can find out the number of nodes associated to this node which may further release its identity.

Neighborhood graphs

Node neighborhood graph is a concept highly related to degree centrality but with some differences [12]. Given a node and its neighbors, how these neighbors connect with each other can be unique. Publishing the neighborhood graph of a node may release the identify of this node.

Edge properties

Besides the properties of network nodes, Zheleva and Getoor also studied some sensitive properties related to network edges[13]. Two types of information of an edge can be potential sensitive information. One is the existence of an edge between two given nodes. The other is the label of a given edge which represents the type of relationship.

Network properties

Social network data has a set of important properties which can be considered as sensitive information in some cases, such as diameter, radius, betweenness, closeness, clustering coefficient etc.

Social network privacy attack model

To have a better protection against privacy attack, it is important to understand different types of privacy attack models. In this section, we introduce two categories of attack, active and passive attacks [11,14].

Active attacks

Backstorm et al. [14] introduced the active attack model. An adversary can actively select an arbitrary set of target actors, creates a small number of new actors with edges connecting to these targeted users, and then creates a pattern of links among the new actors. By planting new actors and connection patterns in the anonymized social network sophisticatedly, the adversary is able to identify the new actors as well as the targeted actors if the generated connection patterns are uniquely stand out in the anonymized network. Theoretically, the creation of $O(\sqrt{\log n})$ nodes in an n-node network will begin compromising the privacy of the arbitrary targeted nodes. Backstorm et al., [14] further divided the active attacks into walk-based attack and cut-based attack. Both of them employed the strategy of inserting nodes into the target network and then link these nodes with the target nodes. The difference between them is the theoretical number of nodes used in the attack.

Passive attacks

Backstorm et al. [14] also investigated the passive attack model, where adversaries do not create any new nodes or edges. Backstorm et al. pointed out that attacker with certain knowledge can easily differentiate the target nodes or edges from the others due to their unique structural information. Most current studies focus their research on preventing passive attacks, which includes: (1) node passive attack [8-11], where adversaries are supposed to take advantage of node's degree centrality information to uncover node's identity; (2) edge passive attack [13-15], where adversaries are supposed to know the existence of certain edges, leading to the disclosure of sensitive information by tracking the identify of other edges or nodes via known edges; (3) sub-graph passive attack [9,11,12], where adversaries are supposed to make use of sub-graph information known in advanced to identify sensitive information of node, such as node identity; (4) graph metrics passive attack [16], where the adversaries have certain background knowledge of the graph metrics, for example hub fingerprint, closeness centrality or betweenness centrality. With the knowledge of these graph metrics, it's also possible that adversaries can uncover several sensitive information of the social network.

Privacy preservation models and algorithms

In the recent years, a number of approaches for preserving privacy of relational data have been studied extensively, which include k-anonymity [17,18], l-diversity[19], Personalized anonymity[20], and (α,k)-Anonymity[21]. One common objective of these algorithms is to ensure every node is indistinguishable to other (k-1) nodes after anonymization. Although these methods work well in relational table data, most of them cannot deal with social network data due to the complex structure of social network and various background and attack model employed by an adversary. In the recent years, a few research groups have investigated the privacy preservation of social network data. They preserve the data privacy mainly by three approaches: perturbation-based approach, generalization-based approach, and protocol-based approach. Different techniques correspond to different type of sensitive data as well as privacy requirement.

Perturbation-based technique

The perturbation-based technique perturbs a social network by adding, deleting or switching edges in a social network in order to increase the difficulty of identifying a node. Most of them are using greedy algorithm guided by an objective function to modify the social network step by step until the anonymized network satisfied some given conditions. Liu and Terzi proposed the K-degree Anonymous Algorithm to ensure that each network node is indistinguishable to other (K-1) nodes [10]. Starting from the original degree sequence **d** of input graph **G**, the algorithm constructs a new degree sequence $\hat{d}$ which satisfies two conditions including: $\hat{d}$ is k-anonymous and $\sum_i \left| d(i) - \hat{d}(i) \right|$ is minimized. Zhou and Pei proposed the K-neighborhood Algorithm to make sure that node identity cannot be re-identified by an adversary with a confidence larger than 1/k, even though the adversary has background knowledge of the neighborhood graph [12]. The whole process is divided into two phases. First, the algorithm extracts the neighborhoods of all nodes in the

network. To facilitate the comparisons among neighborhoods of different nodes, the researchers proposed a neighborhood component coding technique to represent the neighborhoods in a concise way. In the second step, the algorithm greedily organizes nodes into groups and anonymizes the neighborhoods of nodes in the same group. The greedy algorithm is guided by an anonymization cost which is measured by the similarity between the neighborhoods of two nodes. Ying and Wu proposed the Spectrum Preserving Algorithm which preserves the privacy by randomly perturbing edges in the network [16]. The whole process can be divided into three steps: at first, the eigenvalues of the input graph is computed; and then based on some proved theorems, the boundaries of eigenvalues are given; finally the algorithm perturbs the graph by adding, deleting or switching edges of the graph. If the eigenvalues of perturbed graph is within the given boundaries, the perturbation is accepted and continued for next perturbation. The algorithm terminates until the precondition is satisfied.

Generalization-based technique

The generalization-based technique preserves a social network by grouping certain number of nodes or edges together and then only release the general information of the groups of nodes or edges. Nodes within a group cannot be differentiated because they all share exactly the same properties of the group. In most cases, a generalization-based technique divides nodes according to some predefined loss functions. Hay et al. proposed a node splitting-based technique to achieve k-anonymity of the social network [9,11]. Starting from a single partition of a social network, the algorithm keeps on splitting the selected partition into two sub-groups until all predefined criteria are satisfied. Similarly, Campan and Truta introduced a node clustering-based approach to satisfy the k-anonymity requirement and minimize the information loss [8]. In their algorithm, clusters are created one at a time. To form a new cluster, a node in V with the maximum degree but not yet be allocated to any cluster is selected as a seed for the new cluster. Then the algorithm puts nodes to this currently processed cluster until it reaches the desired cardinality k. At each step, the current cluster grows with one node. The selected node should not be assigned to any cluster yet but it should be able to minimize the growth of information loss of the current clusters. Zheleva and Getoor proposed an edge clustering-based technique to hide the sensitive information on edges [13]. Their technique is divided into two phases. In the first phase, the technique provides a clustering of the nodes into m equivalence class (C1,C2,. . .,CM) such that each node is indistinguishable in its quasi-identifying attributes from K-1 other nodes. In the second step, this work presents several techniques to protect sensitive information of the social network and then compare their performance, which includes partial-edge removal, cluster-edge anonymization, and cluster-edge anonymization with constraints. Cormode and Srivastava proposed the safe groupings technique for a bipartite Graph G = (V,W,E), where V and W correspond to two types of objects [15]. In their work, a safe grouping of a bipartite graph partitions nodes into groups such that two nodes of the same group of V have no common neighbors in W and vice versa. A greedy algorithm is proposed to find K safe groups of V and L safe groups of W. For each node u, the algorithm attempts to assign u to the first group with fewer than n nodes. If it makes the grouping unsafe, the algorithm will try the second available group and so forth. If there is no group that meets the requirements, a new group will be created to contain this node. After getting K safe groups of V, the algorithm move forward to find L safe groups of W following a similar same process.

Protocol-based technique

The protocol-based technique is using the encryption approach rather than anonymizing the social network data. Social network data is encrypted by following a protocol before sharing with other parties. The protocol ensures that other parties are only able to obtain the insensitive information for their applications but the sensitive information is preserved. Frikken and Golle proposed the pieces assembling approach for private social network analysis [22].

Summary

In this section, we provide a summary of the literature by comparing the privacy preservation techniques and the preserving data in social network as shown in Table 1. In general, some privacy preservation techniques are developed for preserving specific information but are not applied to other information. The choice of the privacy preservation techniques also depends on the application of social network analysis.

Research problem

The existing works focus on preserving privacy of social network data for data publishing so that the global network structure can be analyzed. However, it has not considered how to integrate social network data from different sources so that social network analysis and mining can be conducted on the integrated data and yet the privacy of the shared data can be preserved. Individual published social network data only capture parts of the complete social network. Unless we can integrate multiple social networks and conduct SNAM on the integrated social network, the utility of the anonymized data is still limited. Given multiple law enforcement

Table 1 Classification of privacy preservation techniques based on sensitive information

		Types of sensitive information						
		Node	Node	Link	Link	Subgraph	Aggregated	Other
		Existence	Properties	Existence	Properties	Property	Graph property	Graph information
Privacy preservation techniques	Perturbation		[10]	[16]			[16]	
	Based							
	Technique							
	Generalization	[4,6]	[8,9,11]	[4,6,15]				[8]
	Based							
	Technique							
	Hybrid		[12]	[13]	[13]	[12]		
	(Perturbation & Generalization)							
	Protocol							[22]
	Based							
	Technique							

units and each of them has a criminal social network which captures a partial picture of a complete criminal social network, the objective of this work is preserving the privacy of the shared data from each law enforcement unit and conduct SNAM tasks on the integrated data. In this section, we first define formally the research problem, introduce what sensitive information to preserve, what insensitive information to share and what SNAM task can be conducted. Then, we proposed a research framework to address the research problem.

Problem definition

Given a set of network $\mathcal{G} = \{G_1, G_2, \ldots, G_n\}$ in a distributed setting where each organization i owns its piece of G_i, assuming the complete network $G(G = \cup_{i=1}^{n} G_i)$ is unknown to each individual organization, the goal of this paper is to study how to anonymize each G_i into G_i' so that: 1) the sensitive identities of G_i can be protected; 2) G_i' can be shared with other organizations and the integrated anonymization graph $G'(G' = \cup_{i=1}^{n} G_i'$ can be used for SNAM task. Concretely, each network G_i' consists of both insensitive nodes and sensitive nodes. Node identities of those insensitive nodes are known to the public or the sharing parties while the node identities of those sensitive nodes are unknown to the public and needed to be protected. Our focus in this paper is to protect the node identities of those sensitive nodes. On the other hand, for SNAM purpose, some network properties, including topology, diameter and some other abstract features of the anonymized network, will be released and shared across organizations. Last but not least, it's important to note that some network features cannot be preserved in our method, such as neighborhood information. Therefore, not all SNAM tasks can be achieved in our integrated anonymized network. In this paper, we only study how to preserve the usefulness of the integrated anonymized network regarding to distance-related analysis, such as computing the closeness of each node. To summarize, although some existing works have studied how to anonymize network for data publishing, the research problem that we study here is different. We not only anonymize a given network to protect its' node identity but also focus on integrating anonymized networks to achieve better SNAM results

Framework of social network integration with privacy preservation

To further motivate our research framework, we assume organization P (O_P) has a social network G_P and organization Q (O_Q) has another social network G_Q, both G_P and G_Q are partial networks of a complete social network which is unknown to any organization. O_P needs to conduct a Social Network Analysis and Mining (SNAM) but G_P is incomplete due to its limited sources of information. As a result, it will be difficult or even impossible for O_P to get accurate SNAM results. If there is no privacy concern between different organizations, one can integrate G_P and G_Q to generate an integrated $\boldsymbol{G}$ and obtain a better SNAM result. However, due to privacy concern, O_Q cannot share G_Q completely with O_P, but only shares the insensitive information of G_Q with O_P according to the privacy policies. At the same time, O_P does not need all data from O_Q but only those that are critical for the SNAM tasks. For these reasons, to integrate social networks of different organizations without violating privacy policies, we only need to share information that is critical to the performance of SNAM and yet preserve the sensitive information.

Figure 1 demonstrates the general framework of social network integration for SNAM. In this framework, O_Q employs sub-graph generalization techniques to create a generalized social network, G_Q', from G_Q without violating the privacy policy. The generalized social network only contains generalized information of G_Q without releasing any sensitive information. For example, a generalized social network cannot release the exact identity of each nodes or exact shortest distance between any two nodes. On the other hand, generalized information can include diameter of a sub-graph, average number of adjacent nodes between two subgroups, degree of an insensitive node and other insensitive information. The generalized social network G_Q' will then be integrated with G_P to support a social network analysis and mining task. Given the generalized information from G_Q, it is expected to achieve better performance on SNAM task than conducting the analysis and mining on G_P alone. There are two important sub-tasks in our proposed framework which we will address in the following sections:

Task 1 Given a social network G with sensitive information, produce generalized social network G' and determine the generalized information which can be released.

Task 2 Integrate a generalized social network with the local social network, and then utilize shared generalized information to achieve better SNAM results.

Notations

In Table 2, we define a set of notations for the proposed social network integration techniques.

Methodology

Social network generalization

In task one, given a social network G with sensitive information, we employ clustering-based technique to produce a generalized social network G'. We suppose $G = (V, E)$, where V is a set of nodes, E is a set of edges and $|V| = n$, K of these nodes are insensitive nodes, and n-K of these nodes are sensitive nodes. We generate a generalized social network in two steps. In the first step, we decompose G into K sub-graphs $G_i = (V_i, E_i)$, where $V = U_{i=1 \text{to} K} V_i$ and each sub-graph contains one insensitive node. In the second step, each sub-graph will be transformed to a generalized node of the generalized graph G'. Furthermore, two generalized nodes will be connected in G' if and only if there is one or more edges connecting nodes from these two sub-graphs respectively.

In this section, we propose two graph partition algorithms, *K*-nearest neighbor (*KNN*) method and Edge betweenness based (EBB) method, to generate a generalized social network G' for sharing purpose. Both KNN and EBB methods are developed by following one common principle that the identity of insensitive nodes can be published safely while the identity of sensitive nodes cannot, so that, to produce a generalized social network, we need to divide the original network into several sub-graphs each of which represented by an insensitive nodes, and the final generalized network should be also represented by these insensitive nodes.

K-nearest neighbor (KNN) method

Given a social network G with K insensitive nodes $v_1^c, v_2^c, \ldots, v_k^c$ KNN method divides G into K sub-graphs by assigning each node v to its nearest insensitive node. Let $SP^D(v, v_i^C)$ be the distance of the shortest path between v and v_i^C. Starting from the sensitive nodes

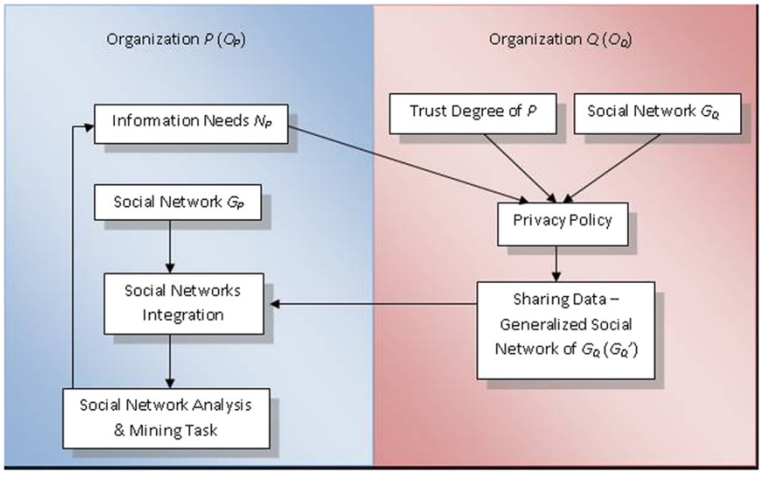


Figure 1 General framework of social network integration for SNAM.

Table 2 Notations and definitions

$G = (V, E)$	a social network G with |V| nodes and |E| edges
V	node set of G
E	edge set of G
G'	a generalized version of G
$G_i = (V_i, E_i)$	a sub-graph of G where $V = \cup_{i=1\ to\ k} V_i$ and $E_i \subset E$
v_i^c	the center of a sub-graph G_i which is an insensitive node too
v_p	node p
$Num(G_i)$	The number of nodes in G_i
$Num(G_i, G_j)$	the number of nodes in G_i that are adjacent to another subgraph G_j
$SP^D(v_p, v_q, G_i)$	the distance of the shortest path between nodes v_p and v_q in G_i
$SP^D(v, v_i^C)$	the distance of the shortest path between v and v_i^C in G_i
$Prob(SP^D(\cdot)=\beta)$	The probability of the distance that equals to β
$S\_SP^D(G_i)$	shortest length of the shortest paths between any two nodes in G_i ($S\_SP^D(G_i) = \{SP^D(v_m, v_n, G_i) \mid \forall\ v_p, v_q \in V_i, SP^D(v_m, v_n, G_i) \le SP^D(v_p, v_q, G_i)\}$)
$L\_SP^D(G_i)$	longest length of the shortest paths between any two nodes in G_i ($L\_SP^D(G_i) = \{SP^D(v_m, v_n, G_i) \mid \forall\ v_p, v_q \in V_i, SP^D(v_m, v_n, G_i) \ge SP^D(v_p, v_q, G_i)\}$)
$S\_SP^D(v_i^C, G_i)$	shortest length of the shortest paths between v_i^C and other nodes in G_i ($S\_SP^D(v_i^C, G_i) = \{SP^D(v_m, v_i^C, G_i) \mid v_p \in V_i, SP^D(v_m, v_i^C, G_i) \le SP^D(v_p, v_i^C, G_i)\}$)
$L\_SP^D(v_i^C, G_i)$	longest length of the shortest paths between v_i^C and other nodes in G_i ($L\_SP^D(v_i^C, G_i) = \{SP^D(v_m, v_i^C, G_i) \mid v_p \in V_i, SP^D(v_m, v_i^C, G_i) \ge SP^D(v_p, v_i^C, G_i)\}$)

adjacent to insensitive node, KNN method assign sensitive node, one node per time, to the closest sub-graph G_i where $SP^D(v, v_i^C)$ is shorter than or equal to $SP^D(v, v_j^C)$ where $j = 1, 2, .., K$ and $j \neq i$. After dividing a social network into K sub-graphs, we collapse all nodes of a sub-graph into one generalized node, and represent this node with the identity of the insensitive node of this sub-graph. Finally, for each possible pair of generalized nodes, say G_i and G_j in the generalized graph G', an edge will be created if and only if there is one or more edges between any two nodes in G from sub-graph G_i and G_j respectively.

Figure 2 presents a simple example to illustrate the idea of KNN and show how it works to produce generalized social network. Figure 2 (a) is the given social network which has seven nodes. Among them, v_1 and v_2 are insensitive nodes while the others are all sensitive nodes. By using KNN method, the given social network will be divided into two isolated social networks as shown in Figure 2 (b). Finally, one sub-graph is represented by v_1 and another sub-graph is represented by v_2, Figure 2 (c) demonstrates the final generalized social network where two generalized nodes are connected together because v_4 and v_5 are connected in G. The *KNN* subgraph generation algorithm is presented below:

length = 1;
V = V - {v_1^C, v_2^C, ... v_K^C};
While V ≠ Ø
  For each $v_j \in V$
  For each $i = 1$ to K
    IF($SP^D(v_j, v_i^C)$ == length);
      $V_i = V_i + v_j$;
      $V = V - v_j$;
    End For;
  End For;
  length++;
End While
For each $(v_i, v_j) \in E$
  IF($Subgraph(v_i)$ == $Subgraph(v_j)$)
  //$Subgraph(v_i)$ is the subgraph such that $v_i \in Subgraph(v_i)$
    $G_k = Subgraph(v_i)$
    $E_k = E_k + (v_i, v_j)$

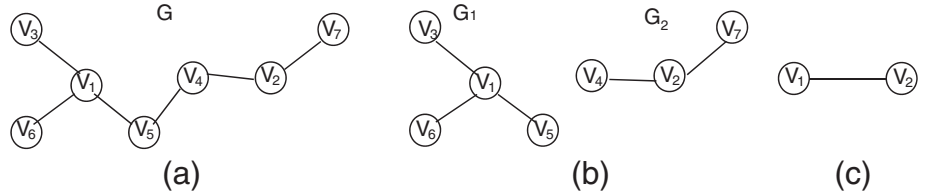


Figure 2 Illustrations of generating subgraphs.

```
    ELSE
      Create an edge between Subgraph(v_i) and
      Subgraph(v_j) and add it to E'
  End For
```

Edge betweenness based (EBB) method

Instead of assigning sensitive nodes to the closest sub-graphs represented by insensitive nodes, the EBB method progressively remove edges with the highest betweenness and it also ensure that each separated sub-graph contains exactly one insensitive node. The betweenness of an edge is defined as the number of shortest paths between pairs of nodes that pass through it. If a network consists of a few of dense communities which are only loosely connected by some inter-community edges, these inter-community edges will have high betweenness, so that removing them will naturally break the social network into multiple communities. The EBB algorithm is presented as follows:

```
//EBB(G), Edge Betweenness Based method
Initialize e = {};
While(there are more than one insensitive node in
graph G)
  Identify edge (v_i,v_j) in G which is not an element of e
  and has the highest betweenness;
  Remove (v_i,v_j) from G;
  IF(G is still connected after removing edges (v_i,v_j))
    EBB(G);
  ELSE IF (G is disconnected and split to two graph
  G_p and G_q)
    IF(No insensitive node in G_p) or (No insensitive
    node in G_q)
      Add (v_i,v_j) back to G;
      e = e + (v_i,v_j);
Go Back to Step 2;
    ELSE
      EBB(G_P);
      EBB(G_q);
End While;
//Add edge between generalized node to form
generalized graph
For each (v_j,v_j) ∈ E
  IF(Subgraph(v_i) == Subgraph(v_j))
    G_k = Subgraph(v_i)
    E_k = E_k + (v_i,v_j)
  ELSE
    Create an edge between Subgraph(v_i) and
    Subgraph(v_j) and add it to E'
End For
```

Figure 3 shows an example of how EBB method works to produce generalized social network. Given a social network with nine nodes, v_1 and v_2 are insensitive nodes while all other nodes are sensitive nodes. Since edge (v_1, v_2) has the highest Betweenness and it is safe to be removed, EBB method delete this edge to form two separated sub-graphs each of them contains exactly one insensitive node, as shown in Figure 3 (b). Finally, the EBB method generalizes these two sub-graphs into two generalized nodes, and then connects them to form the generalized graph as shown in Figure 3 (c).

Generalized sub-graph information

Given a generalized social network G_i and its center v_i^C, we select shareable network properties based on the information need and the privacy policy. In this paper, we treat node identity as sensitive information that we should protect, and consider distance between nodes to be useful information for SNAM task. Let v_a and v_b be any two nodes in G_i and the length of the shortest path between v_a and v_b be $SP^D(v_p,v_q,G_i)$. We define the longest length of the shortest paths between any two nodes in G_i, denoted by $L\_SP^D(G_i)$, as

$$L\_SP^D(G_i) = \{SP^D(v_m, v_n, G_i) | \exists v_m, v_n, \forall v_a, v_b \in V_i, SP^D \times (v_m, v_n, G_i) \geq SP^D(v_a, v_b, G_i)\}$$

We also define the shortest length of the shortest paths between any two nodes in G_i, denoted by $S\_SP^D(G_i)$, as

$$S\_\mathrm{SP^D}(\mathrm{G_i}) = \{\mathrm{SP^D}(\mathrm{v_m}, \mathrm{v_n}, \mathrm{G_i}) | \exists \mathrm{v_m}, \mathrm{v_n}, \forall \mathrm{v_a}, \mathrm{v_b} \in \mathrm{V_i}, \mathrm{SP^D} \times (\mathrm{v_m}, \mathrm{v_n}, \mathrm{G_i}) \leq \mathrm{SP^D}(\mathrm{v_a}, \mathrm{v_b}, \mathrm{G_i})\}$$

To reduce the risk of releasing sensitive information, instead of sharing exact information of shortest path, we propose to share the expected length between two nodes within a generalized social network. Formally speaking, the length of any shortest paths in G_i, α, must be smaller or equal to $L\_SP^D(G_i)$ and larger or equal to $S\_SP^D(G_i)$, where $S\_SP^D(G_i) \leq \alpha \leq L\_SP^D(G_i)$. We compute and share the probability of the length of the shortest path between any two nodes in G_i, denoted as $Prob(SP^D(G_i) = \alpha)$, and $0 \leq Prob(SP^D(G_i) = \alpha) \leq 1$

Similarly, let the length of the shortest path between v_a and v_i^C, be $SP^D(v_a,v_i^C,G_i)$. We define the longest length of the shortest paths between v_i^C and other nodes within G_i, denoted by $L\_SP^D(v_i^C,G_i)$, as

$$\mathrm{L\_SP^D}\left(\mathrm{v_i^C}, \mathrm{G_i}\right) = \mathrm{SP^D}\left(\mathrm{v_m}, \mathrm{v_i^C}, \mathrm{G_i}\right) \left| \forall v_a \in V_i, \mathrm{SP^D} \left(\mathrm{v_m}, \mathrm{v_i^C}, \mathrm{G_i}\right) \geq \mathrm{SP^D}\left(\mathrm{v_a}, \mathrm{v_i^C}, \mathrm{G_i}\right)\right.$$

We also define the shortest length of the shortest paths between v_i^C and other nodes within G_i, denoted by and $S\_SP^D(v_i^C,G_i)$, as

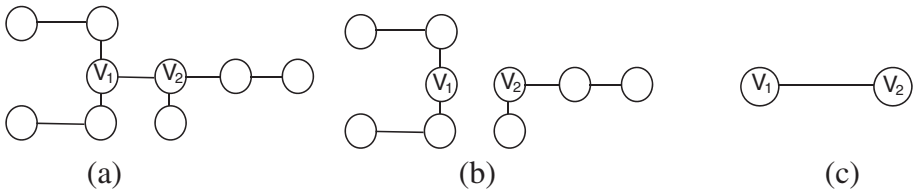


Figure 3 Illustrations of generalizing subgraph byEBB.

$$\mathrm{S\_SP^D}\left(\mathrm{v_i^C}, \mathrm{G_i}\right) = \left\{\mathrm{SP^D}\left(\mathrm{v_m}, \mathrm{v_i^C}, \mathrm{G_i}\right) \middle| \forall v_a \in V_i, \mathrm{SP^D}\left(\mathrm{v_m}, \mathrm{v_i^C}, \mathrm{G_i}\right) \leq \mathrm{SP^D}\left(\mathrm{v_a}, \mathrm{v_i^C}, \mathrm{G_i}\right)\right\}$$

Since the length of shortest paths between v_i^C and any other nodes in G_i must be smaller or equal to $L\_SP^D(v_i^C, G_i)$ and larger or equal to $S\_SP^D(v_i^C, G_i)$, denoted as $S\_SP^D(v_i^C, G_i) \leq \beta \leq L\_SP^D(v_i^C, G_i)$. We compute the probability of the length of the shortest path between any node and v_i^C, $Prob(SP^D(v_i^C, G_i) = \beta)$, where $0 \leq Prob(SP^D(G_i) = \alpha) \leq 1$.

We also denote $Num(G_i)$ as the number of nodes in G_i and $Num(G_i, G_j)$ as the number of nodes in G_i that are adjacent to another subgraph G_j.

The generalized subgraph information for sharing includes: (i) $L\_SP^D(G_i)$, (ii) $S\_SP^D(G_i)$, (iii) $Prob(SP^D(G_i) = \alpha)$, (iv) $L\_SP^D(v_i^C, G_i)$, (v) $S\_SP^D(v_i^C, G_i)$, (vi) $Prob(SP^D(v_i^C, G_i) = \beta)$, (vii) $Num(G_i)$, and (viii) $Num(G_i, G_j)$.

Generalized graph integration and social network analysis

In section 4.2 and 4.3 we introduced how to divide a social network into sub-graphs, and then generalize these sub-graphs to nodes, then finally produce a generalized social network. We also discussed what kind of information will be shared along with the generalized social network. In this section, given a generalized social network G' and the shareable information of the sub-graphs of G', we propose our own techniques to integrate social network and shared information to improve the performance of SNAM task.

Suppose organization O_p has a social network G_p and organization O_Q has another social network G_Q, O_p wants to integrate G_Q with its own G_p to compute more accurate closeness centrality. We propose to achieve this goal without violating the privacy policies in three steps: (1) produce generalized social network G'_p and G'_Q; (2) integrate G'_p and G'_Q into $G_{Integrated}$; (3) estimate the distance between any two nodes of the integrated social network. Among these three steps, step one can be achieved by our proposed techniques in section 4.2 and 4.3. In step two, although the sub-graphs represented by a common insensitive node in G'_p and G'_Q are different and the connectivity between these insensitive nodes are also different, according to our proposed techniques, G'_p and G'_Q are represented by the same group of insensitive nodes since G_p and G_Q share same insensitive nodes. As a result, we can combine G'_p and G'_Q into $G_{Integrated}$ by taking union of their edges. In this section, we focus on the step 3 which estimate distances between any two nodes based on G_p, $G_{Integrated}$ and shared information of sub-graphs of G'_Q.

To re-estimate the distance between two nodes v_i and v_p of G_p by making use of $G_{Integrated}$ and the shared information of sub-graphs of G'_Q, we first identify the two closest insensitive nodes for v_i and v_j in G_p, and then use $G_{Integrated}$ and the generalized information of G'_Q to re-estimate their distances. Formally speaking, let the closest insensitive node to v_i in G_p be V_A^C, and the second closest insensitive node to v_i in G_p be V_A^C. We set the weights λ_A and $\lambda_{A'}$ as

$$\lambda_A = \frac{SP^D\left(v_i, v_{A\prime}^c, G_P\right)}{SP^D\left(v_i, v_A^c, G_P\right) + SP^D\left(v_i, v_{A\prime}^c, G_P\right)},$$

$$\lambda_{A'} = \frac{SP^D\left(v_i, v_A^c, G_P\right)}{SP^D\left(v_i, v_A^c, G_P\right) + SP^D\left(v_i, v_{A\prime}^c, G_P\right)},$$

with $\lambda_A + \lambda_{A'} = 1$ and the weight of the closest insensitive node is higher.

Similarly, let the closest insensitive node to v_j in G_P be v_A^C, and the second closest insensitive node to v_j in G_P be v_B^C, we set the weights λ_B and $\lambda_{B'}$ as

$$\lambda_B = \frac{SP^D\left(v_j, v_{B\prime}^c, G_P\right)}{SP^D\left(v_j, v_B^c, G_P\right) + SP^D\left(v_j, v_{B\prime}^c, G_P\right)},$$

$$\lambda_{B\prime} = \frac{SP^D\left(v_j, v_B^c, G_B\right)}{SP^D\left(v_j, v_B^c, G_P\right) + SP^D\left(v_j, v_{B\prime}^c, G_P\right)},$$

with $\lambda_B + \lambda_{B'} = 1$

In $G_{Integrated}$, v_A^C, v_A^c, v_B^C, and $v_{B'}^C$ are the centers of generalized sub-graphs G_A, G_A, G_B, and G_B, respectively.

We estimate the distance between v_i and v_j, $d(v_i,v_j)$, by integrating the estimated distances of the four possible paths going through these insensitive nodes by a linear combination with weights equal to $\lambda_\alpha \times \lambda_b$.

$$d(v_i, v_j) = \sum_{\substack{a\in\{A,A\prime\} \\ b\in\{B,B\prime\}}} \lambda_a \times \lambda_b \times D(v_i, v_j)$$

$D(v_i, v_j)$ is the estimated distance between v_i and v_j on the path going through v_a^c and v_b^c, where a can be A or A' and b can be B or B'.

$$D(v_i, v_j) = \begin{cases} D'(G_a, v_i) + 1 + \sum_{\forall G_k} (E(G_k) + 1) + D'(G_b, v_j) & a \neq b \\ D''(v_i, v_j) & a = b \end{cases}$$

where G_k is a generalized node on the shortest path between G_α and G_b in $G_{Integrated}$ If $a \neq b$ which means v_i *and* v_j are not in the same subgraph, then $D(v_i, v_j)$ is estimated by $D'(G_a, v_i), D'(G_b, v_j)$, and $E(G_k)$. Otherwise, if v_i *and* v_j are in the same subgraph then $a = b$. In this case, $D(v_i, v_j)$ is estimated by $D''(v_i, v_j).D'(G_a, v_i)$ corresponds to the expected length of the distance between v_i and the sub-graph gatekeeper within G_α. Similarly, $D'(G_b, v_j)$ corresponds to the expected length of the distance between v_j and the sub-graph gatekeeper within G_b. In addition, $E(G_k)$ is the expected length of the distance between any two nodes of sub-graph G_k that the shortest path between v_i and v_j is going through. If v_i is not the same as $v_a^c, D'(G_a, v_i)$ is computed by $E(G_\alpha)$ and the percentage of nodes in G_a that is adjacent to the sub-graph that is immediately following G_a in the shortest path between v_i and v_j in $G_{Integrated}$. If v_i is the same as $v_a^c, D'(G_a, v_i)$ is equal to the expected length of the distance between the insensitive node, v_a^c, to the other nodes in G_a. Computation of $D'(G_b, v_i)$ is done similarly.

$$D'(G_a, v_i) = \begin{cases} \left(1 - \frac{Num(G_a, G_k)}{Num(G_a)}\right) \times E(G_a) & v_i \neq v_a^c \\ \sum_{\beta=S\_SP(v_a^c, G_a)}^{L\_SP(v_a^c, G_a)} Prob(SP^D(v_a^c, G_a) = \beta) \times \beta & v_i = v_a^c \end{cases}$$

where $\frac{Num(G_a,G_k)}{Num(Ga)}$ is the percentage of nodes in G_α as a gatekeeper which is adjacent to G_k and G_k is the sub-graph that immediately follows G_α in the shortest path between v_i and v_j in $G_{Integrated}$.

$E(G_k)$ represents the expected length of the distance between any two nodes of the sub-graph G_k, which is computed as:

$$E(G_k) = \sum_{\alpha=S_SP(G_k)}^{L\_SP(G_k)} (Prob(SP^D(G_k) = \alpha) \times \alpha$$

$D''(v_i, v_j)$ corresponds to the estimated distance between v_i and v_j when both v_i and v_j are nodes of the same sub-graph. In this case, if any of v_i or v_j is the same as v_a^c, $D''(v_i, v_j)$ should equal to the expected length of the distance from the insensitive node to the other nodes in, G_α. Otherwise, $D''(v_i, v_j)$ should equal to the expected length of the distance between two nodes of the sub-graph.

$$D''(v_i, v_j) = \begin{cases} \sum_{\beta=S\_SP(v_a^c, G_a)}^{L\_SP(v_a^c, G_a)} Prob(SP^D(v_a^c, G_a) = \beta) \times \beta & v_i or v_j = v_a^c \\ E(G_a) & else \end{cases}$$

Experiment and discussion

Practically, there isn't any intelligence unit has a complete terrorist social network but each of them has a partial terrorist social network. The objective of this work is to support these intelligence units to share their social networks while preserving the sensitive information. In this section, we investigated our proposed techniques on a real-world dataset of terrorists. We extracted several social networks from the terrorist dataset to simulate the real-world problem. Intensive experiment was conducted under different settings to evaluate our proposed techniques.

Dataset

In this work, we employed the Global Salafi Jihad terrorist social network, denoted as **G**, in our experiment. The Global Salafi Jihad terrorist social network consists of 366 nodes (terrorists) and 1,275 edges (connection between terrorists)[23]. These terrorists come from four major groups, including Central Staff of al Qaeda (CSQ), Core Arab (CA), Southeast Asia (SA), and Maghreb Arab (MA). We randomly sample α percent of nodes from the Global Salafi Jihad terrorist social network as insensitive nodes, that their identities are known by all organizations. Suppose there are two independent organizations O_P and O_Q, we simulate G_P for O_P by randomly removing β percent of edges from the Global Salafi Jihad terrorist social network. Similarly, we randomly remove β percent of edges from the Global Salafi Jihad terrorist social network to simulate G_Q for O_Q. As a result, both G_P and G_Q are partial graph of G. Moreover, G_P are different from G_Q in terms of their edges.

Evaluation

As discussed before, there is no generic approach for privacy preservation since sensitive information can be defined in various ways. Moreover, shareable useful information is also different in terms of different SNAM tasks. In this work, we treat node identity as sensitive information and consider distance between nodes as useful information that we want to maintain. To evaluate our proposed technique, we assume that the SNAM task conducted by G_P is to compute closeness centrality for each node. If G_P is close to G, then distances between any two nodes in G_P should be roughly equal to their

distance in G, leading to similar closeness centrality for each node. Otherwise, nodes in G_P should have different closeness centrality in G. In this work, closeness centrality for a node in G_P is computed as:

$$closeness\ centrality_{G_P}(v_i) = \frac{n-1}{\sum_{j=1, i\neq j}^{n} SP(v_i, v_j, G_P)}$$

where n is the total number of nodes in G_P.

Given a complete social network G and the integrated social network $G_{Integrated}$, the performance of our proposed technique is evaluated by the error function defined as:

$$Error(G_{Integrated}) = \sum_{i=1}^{n} |closeness\ centrality_{G_{Integrated}}(v_i) - closeness\ centrality_G(v_i)$$

Experiment

Figure 4 demonstrates the average closeness centrality of nodes of original graph (G), integrated graph using EBB method ($G_{Integrated}$ (EBB)), integrated graph using KNN method ($G_{Integrated}$ (KNN)) and incomplete graph (G_P). In Figure 4, the blue line represents the average closeness centrality computed from G, which is a gold standard, so that the closer to this blue line the better it is.

For each α from 0.05 to 0.95, we increased β (percentage of edges randomly removed from G) from 0.2 to 0.8. We observed that the performance of G_P ($G_{Integrated}$ (KNN)) and ($G_{Integrated}$ (EBB)) decreased consistently when more edges are removed from the complete graph, no matter what the value of α is. Although our proposed technique integrates networks and estimates the average closeness centrality, the performance will not be as good as the average closeness centrality computed from the

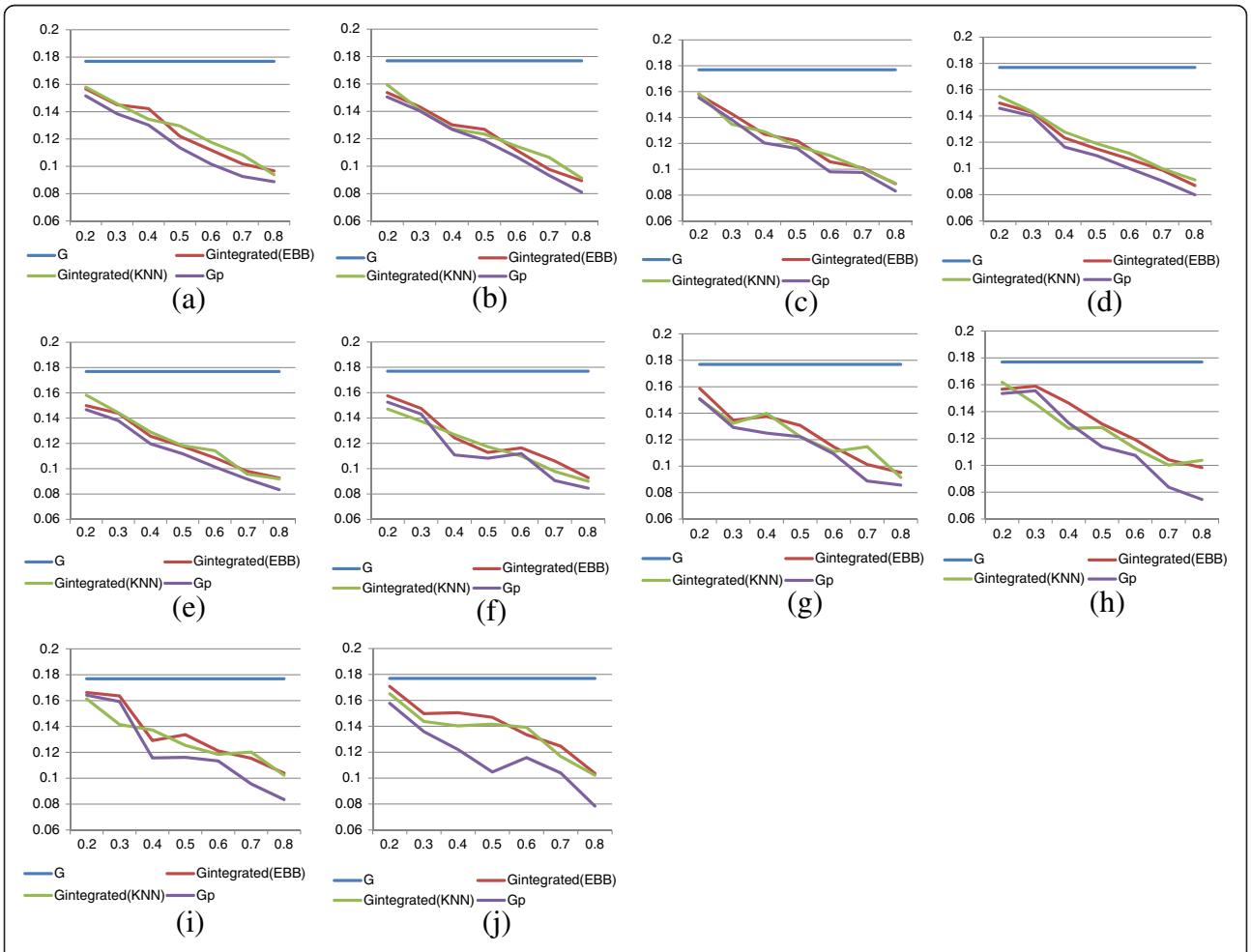


Figure 4 Average closeness centrality of complete graph (G), integrated graph using EBB method ($G_{integrated}(EBB)$), integrated graph using KNN method ($G_{integrated}(KNN)$) and incomplete graph G_P: (a) α = 0.05, (b) α = 0.15, (c) α = 0.25, (d) α = 0.35, (e) α = 0.45, (f) α = 0.55 (g) α = 0.65, (h) α = 0.75, (i) α = 0.85, (j) α = 0.95.

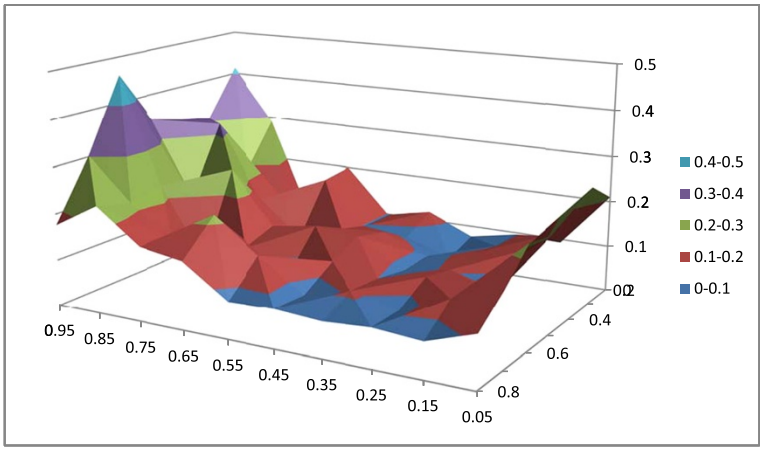
actual graph *G*. When more edges are removed before integration (β increase), the performance will degrade.

We further investigated the performance of our proposed technique by increasing the percentage of insensitive nodes from 0.05 (Figure 4(a)) to 0.95 (Figure 4(j)). Similar patterns are observed from 4(a) to 4(j). In terms of average closeness centrality, increasing or decreasing the percentage of insensitive nodes in network did not make substantial impact to the performance of our purposed technique. One plausible explanation is that: the average closeness centrality used in this experiment only reflects the performance of our approach in an abstract level. Some nodes in the integrated network may have higher closeness centrality than its original closeness centrality in the complete graph while some nodes may have lower closeness centrality in the integrated network than in the complete graph. As a result, when we consider the average closeness centrality, the differences may be offset by each other.

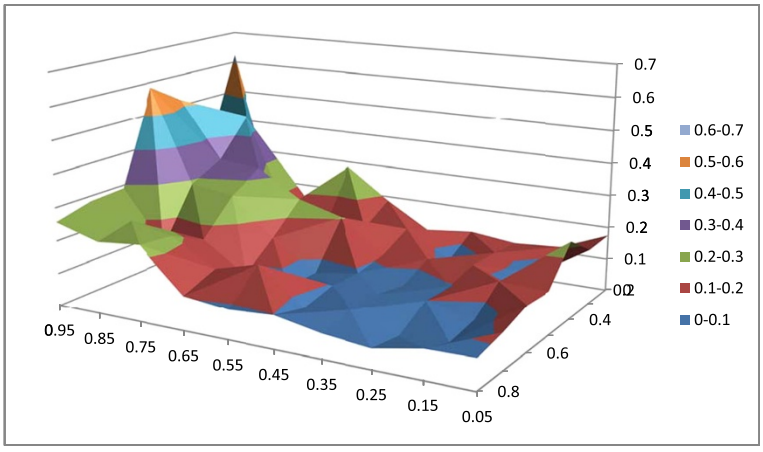
Figure 5 (a) presents the error ratio of ($G_{Integrated}$ (*KNN*)) with different α and β. Similarly, Figure 5 (b) presents the error ratio of $G_{Integrated}$ (*EBB*) with different α and β. We compute the errors in closeness centrality obtained from the networks with and without integration ($Error(G_{integrated})$ and $Error(G_p)$) using the error function defined in 5.2. and the error ratio is defined as:

$$\frac{Error(G_P) - Error(G_{Integrated})}{Error(G_P)}$$

Different from the average closeness centrality which we used as a measurement in Figure 4, the error function accumulates the closeness centrality difference for each



(a)



(b)

Figure 5 (a) error ratio of $G_{integrated}(KNN)$ with different settings of β and α; (b) error ratio of $G_{integrated}(EBB)$ with different settings of β and α.

individual node, so that the offset effect of average closeness will not occur. The experiment results of Figure 5 can be used to verify our explanation to the Figure 4 in the last paragraph.

The experiment results demonstrate that when α is high (means more insensitive nodes), the improvement of our proposed technique comparing to the partial graph is also higher. The highest improvement was achieved when α equals to 0.95. The improvement decreased slowly along with the decrease of α. This observation indicated that our explanation of Figure 4 is correct. With more insensitive nodes, the integrated network will be closer to the original network so that the improvement of our technique will be higher.

Last but not least, from both Figures 4 and 5, we do not observe any significant differences of the performance between using KNN or EBB to produce generalized network. However, as it is shown in section 4.2.2, the EBB algorithm is dominated by the step of calculating the edge betweenness which has time complexity $O(N^3)$. On the other hand, KNN is much more efficient which is only O(N). As a result, when the network size is huge, KNN is preferred. Moreover, in a fully connected network where several edges have the same betweenness weight, EBB will take longer to produce the generalized network. However, KNN also has its limitation. For example, KNN starts from each insensitive node to look for sensitive nodes in its neighborhood to form a sub-graph step by step. However, the search process is not fully simultaneous, but is controlled by a FOR loop. As a result, the sequence in the FOR loop is matter, especially for some nodes in the middle of two insensitive nodes. As a result, the division of sub-graph by using KNN is less natural than EBB method.

Conclusion

In this paper, we investigate the privacy preservation techniques for social network integration. We introduce a research framework which consists of three major steps. First of all, we propose the K-Nearest Neighborhood method and the Edge Betweenness Based method to decompose a social network into multiple sub-graphs. Secondly, we propose techniques to generalize a social network by sharing the probabilistic model of the generalized information. At third, we introduced the techniques of social network integration and distance estimation.

Using the Global Salafi Jihad terrorist social network as test bed, we thoroughly evaluated our proposed technique with different parameters and settings. The experiment results demonstrated that an organization can improve the accuracy of computing closeness centrality by sharing and integrating generalized information. Our proposed techniques were able to preserve the privacy as well as increase the utility of the shared social networks. We observed that KNN performed better than EBB but did not have substantial difference. Moreover, our proposed techniques were not sensitive to the number of insensitive node but relatively sensitive to the number of removed edges.

In the future, we will continue to examine our techniques in more datasets. We will explore other graph partition models and integration techniques to improve the performance of our technique. Moreover, we will also extend our work to maintain other useful information besides distance.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

CCY proposed the research framework. CCY and XT developed the algorithms together. XT implemented the algorithms and conducted experiments. CCY and XT together drafted the manuscript. All authors read and approved the final manuscript.

Received: 23 February 2011 Accepted: 30 May 2012
Published: 12 July 2012

References

1. Yang CC, Sageman M: **Analysis of terrorist social networks with fractal views.** *J Inf Sci* 2009, **35**(3):299–320.
2. Yang CC, Ng T: **"Terrorism and crime related weblog social network: Link, content analysis and information visualization,".** *IEEE Int Conf Intell Secur Inform IEEE*, 23-24 May 2007**:**55–58.
3. Yang CC, Liu N, Sageman M: **"Analyzing the terrorist social networks with visualization tools,".** *IEEE Inte Conf Intell Secur Inform*, 23-24 May 2007**:**331–342.
4. Yang CC, Tang X: **Social networks integration and privacy preservation using subgraph generalization**. In *Proceedings of the ACM SIGKDD Workshop on CyberSecurity and Intelligence Informatics*. Edited by. : ACM; :53–61.
5. Yang CC, Tang X: **Information Integration for Terrorist or Criminal Social Networks.** *Ann Inform Syst* 2010, **9:**41–57.
6. Tang X, Yang CC: **"Generalizing terrorist social networks with K-nearest neighbor and edge betweeness for social network integration and privacy preservation,".** *IEEE Int Conf Intell Secur Inform IEEE*, 23-24 May 2007**:**49–54.
7. Yang CC, Thuraisingham B: **Privacy-Preserved Social Network Integration and Analysis for Security Informatics.** *IEEE Intell Syst* 2010, **25**(3):88–90.
8. Campan A, Truta T: **"A clustering approach for data and structural anonymity in social networks,"**. In *Proceeding of the second ACM SIGKDD International Workshop on Privacy, Security, and Trust in KDD; 2008.*
9. Hay M, Miklau G, Jensen D, Weis P, Srivastava S: *"Anonymizing social networks,".*: University of Massachusetts Technical Report; 2007:07–19.
10. K. Liu, and E. Terzi, "Towards identity anonymization on graphs", Proceedings of the: *ACM SIGMOD international conference on Management of data*. NY, USA: ACM New York; 2008:93–106.
11. Hay M, Miklau G, Jensen D, Towsley D, Weis P: **Resisting structural re-identification in anonymized social networks.** *Proc VLDB Endowment Arch* 2008, **1**(1):102–114.
12. Zhou B, Pei J: **"Preserving privacy in social networks against neighborhood attacks,"**. In *IEEE 24th International Conference on Data Engineering.:* ICDE; 2008:506–515.
13. Zheleva E, Getoor L: **Preserving the privacy of sensitive relationships in graph data.** *Lecture Notes Comput Sci* 2008, **4980:**153.
14. Backstrom L, Dwork C, Kleinberg J: **"Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography,"**. In *Proceedings of the 16th international conference on World Wide Web*. Edited by. NY, USA: ACM New York; :181–190.
15. Cormode G, Srivastava D, Yu T, Zhang Q: **Anonymizing bipartite graph data using safe groupings.** *Proc VLDB Endowment Arch* 2008, **1**(1):833–844.
16. Ying X, Wu X: *"Randomizing social networks: a spectrum preserving approach," SIAM Conf. on Data Mining* August 2008.
17. Sweeney L: **k-anonymity: A model for protecting privacy.** *Int J Uncertainty Fuzziness Knowledge Based Syst* 2002, **10**(5):557–570.

18. Samarati P: **"Protecting respondents' identities in microdata release,".** *IEEE Transac Knowledge Data Eng* 2001, **:**1010–1027.
19. Machanavajjhala A, Kifer D, Gehrke J, Venkitasubramaniam M: **"l-diversity: Privacy beyond k-anonymity".** *ACM Trans Knowledge Discov Data (TKDD)* 2007, **1**(1):3.
20. X. Xiao, and Y. Tao, "Personalized privacy preservation", Proceedings of the: *ACM SIGMOD international conference on Management of data*. NY, USA: ACM New York; 2006:229–240.
21. Wong R, Li J, Fu A, Wang K: *"(alpha, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing,".*: ACM Press; :754–759.
22. Frikken K, Golle P: *"Private social network analysis: How to assemble pieces of a graph privately,".* NY, USA: ACM New York; :89–98.
23. Sageman M: *Understanding Terror Networks.*: University of Pennsylvania Press; 2004.

doi:10.1186/2190-8532-1-7
Cite this article as: Tang and Yang: **Social network integration and analysis using a generalization and probabilistic approach for privacy preservation.** *Security Informatics* 2012 **1**:7.