Anonymization of Centralized and Distributed Social Networks by Sequential Clustering

A.Divya
M.Tech Student,
Srinivasa Institute of Engineering and Technology.

Dr.R.John Mathew
M.Tech, Ph.D
Srinivasa Institute Of Engineering and Technology.

Abstract:
Knowledge engineers interpret and organize information on how to make systems decisions. The social media has grown very vastly in the earlier years known think for all. There are different social media sites like Face book, Twitter, LinkedIn, Google+ and many more that holds public and confidential/ personal information about their users. We work the problem of privacy-preservation in social networks.

We consider the distributed setting in which the network data is split between several data holders. The goal is to arrive at an anonymized view of the unified network without revealing to any of the data holders information about links between nodes that are controlled by other data holders. To that end, we start with the centralized setting and offer two variants of an anonymization algorithm which is based on sequential clustering.

Our algorithms significantly outperform the SaNGreeA algorithm due to Campan and Truta which is the leading algorithm for achieving anonymity in networks by means of clustering. We then devise secure distributed versions of our algorithms. To the best of our knowledge, this is the first study of privacy preservation in distributed social networks. We conclude by outlining future research proposals in that direction.

Index Terms:
Social networks; clustering; privacy preserving data mining; distributed computation; Sequential Clustering.

INTRODUCTION:
Knowledge engineers interpret and organize information on how to make systems decisions. KDEG investigates the fundamental challenges and practical applications of knowledge-driven systems. Its research combines knowledge discovery, representation and reasoning with web-based data management and intelligent systems engineering. This research is advanced through application in a number of areas. However, there were significant issues with trying to apply these conventional methods to the development of expert systems. Many of the design modeling techniques (e.g. flow charts, dataflow diagrams) that were essential to these methods were of minimal value to designing expert systems.

For example, an inference engine (the tool used by expert systems to represent and utilize expert knowledge coded as rules) attempts to abstract away from things like explicit flow of control. The control flow for an expert system can be very hard to predict because for each example the system will be driven by the particular rules that have fired.

This can be a very powerful mechanism, allowing engineers to define knowledge via rules that are independent of specific programs. However, trying to specify such rules and their control flow via diagrams that must specify predefined flow of control will be very difficult.

Indeed one of the goals of expert systems was to abstract away from specific programming. It was often claimed that expert system shells allowed the experts to become programmers and to do away with professional programmers. This was seldom true in reality but it reflects the core idea that expert system shells tried to move the definition of the system to a higher level of abstraction than conventional code.

Thus, the whole need for detailed design before programming was mostly ameliorated. Note that while design diagrams could often be minimized for the expert system itself as expert...
systems began to take off the requirement for them to integrate with existing systems and legacy databases was significant and design of such integration was a critical part of the complete knowledge engineering process. Another issue with using conventional methods to develop expert systems was that due to the unprecedented nature of expert systems they were one of the first applications to adopt rapid application development methods that feature iteration and prototyping as well as or instead of detailed analysis and design.

In the 1980’s few conventional software methods supported this type of approach. Networks are structures that describe a set of entities and the relations between them. A naïve anonymization of the network, in the sense of removing identifying attributes like names or social security numbers from the data, is insufficient. As shown in [2], the mere structure of the released graph may reveal the identity of the individuals behind some of the nodes. The idea behind the attack described in [2] is to inject a group of nodes with a distinctive pattern of edges among them into the network.

The adversary then may link this distinctive structure to some set of targeted individuals. When the naïve anonymized network is published, the adversary traces his injected subgraph in the graph; if successful (namely, there is only one such subgraph in the graph, an event of probability that can be made sufficiently high), the targets who are connected to this subgraph are re-identified and the edges between them are disclosed. Even less sophisticated adversaries may use prior knowledge of some property of their target nodes (say, the number of their neighbors and their interrelations) in order to identify them in the published graph and then extract additional information on them.

**EXISTING SYSTEM:**

A social network, for example, provides information on individuals in some population and the links between them, which may describe relations of friendship, collaboration, correspondence, and so forth. An information network, as another example, may describe scientific publications and their citation links. In their most basic form, networks are modeled by a graph, where the nodes of the graph correspond to the entities, while edges denote relations between them.

Real social networks may be more complex or contain additional information. For example, in networks where the described interaction is asymmetric (e.g., a financial transaction network), the graph would be directed; if the interaction involves more than two parties (e.g., a social network that describes co-membership in social clubs) then the network would be modeled as a hypergraph; in case where there are several types of interaction, the edges would be labeled; or the nodes in the graph could be accompanied by attributes that provide demographic information such as age, gender, location, or occupation which could enrich and shed light on the structure of the network. Such social networks are of interest to researchers from many disciplines, be it sociology, psychology, market research, or epidemiology. However, the data in such social networks cannot be released as is, since it might contain sensitive information. Therefore, it is needed to anonymize the data prior to its publication in order to address the need to respect the privacy of the individuals whose sensitive information is included in the data. Data anonymization typically trades off with utility.

**DISADVANTAGES OF EXISTING SYSTEM:**

In an existing system, the complexity in communication increases and security level is low in social network system. Real social networks may be more complex or contain additional information.

**PROPOSED SYSTEM:**

In this study, we deal with social networks where the nodes could be accompanied by descriptive data, and propose two novel anonymization methods of the third category (namely, by clustering the nodes). Our algorithms issue anonymized views of the graph with significantly smaller information losses than anonymizations issued by the existing algorithms.

The study of anonymizing social networks has concentrated so far on centralized networks, i.e., networks that are held by one data holder. However, in some settings, the network data is split between several data holders, or players. For example, the data in a network of email accounts where two nodes are connected if the number of email messages that they exchanged was greater than some given threshold, might be split between several email service providers.
As another example, consider a transaction network where an edge denotes a financial transaction between two individuals; such a network would be split between several banks. In such settings, each player controls some of the nodes (his clients) and he knows only the edges that are adjacent to the nodes under his control. It is needed to devise secure distributed protocols that would allow the players to arrive at an anonymized version of the unified network.

Namely, protocols that would not disclose to any of the interacting players more information than that which is implied by its own input (being the structure of edges adjacent to the nodes under the control of that player) and the final output (the anonymized view of the entire unified network). The recent survey by Wu et al. about privacy-preservation in graphs and social networks concludes by recommendations for future research in this emerging area. One of the proposed directions is distributed privacy-preserving social network analysis, which “has not been well reported in literature.”

**ADVANTAGES OF PROPOSED SYSTEM:**

We deal with social networks where the nodes could be accompanied by descriptive data, and propose two novel anonymization methods of the third category (namely, by clustering the nodes).

**SYSTEM ARCHITECTURE:**

**MODULES DESCRIPTION:**

1. **Input Dataset:**

   In this module, the user input the dataset for Centralized and Distributed Networks. The Dataset concept allows you to define sets of data from different tables and export this data in one step.

   Datasets are especially useful for managing reference data for a module, for example tax rates or default data in new tables added by a module.

   The reference data is packaged, distributed and installed together with the program code implementing the module. The content of a Dataset is defined by its Dataset Tables and Dataset Columns.

2. **Extraction:**

   Datasets can be defined at System, Organization, or Client/Organization levels. System-level datasets are applied when the module containing them is installed in the system.

   The user extracts the particular reference value to get particular information from the input dataset. And the Extraction Pattern makes the user to refer easily whenever or whatever they are in need of it. Extraction module performs action with the feature classification.

3. **Sequential Clustering:**

   Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). In this dataset Sequential Clustering is done. Sequential Clustering is a topic of data mining concerned with finding statistically relevant patterns between data examples where the values are delivered in a sequence.

   It is usually presumed that the values are discrete, and thus time series mining is closely related, but usually considered a different activity. Sequential pattern mining is a special case of structured data mining.
(4) Anonymization algorithm:

Random samples were drawn from data set at nine different sampling fractions. Any identifying variables were removed and each sample was k-anonymized. An existing global optimization algorithm was implemented to k-anonymize the samples. This algorithm uses a cost function to guide the k-anonymization process (the objective is to minimize this cost). A commonly used cost function to achieve baseline k-anonymity is the discernability metric. Finally, Graph Evaluation is also made to show the performance analysis.

This Project, focus on privacy-preservation in social networks. The goal is to arrive at an anonymized view of the unified network without revealing to any of the data holders information about links between nodes that are controlled by other data holders. To that end, we start with the centralized setting and offer two variants of an anonymization algorithm which is based on sequential clustering.

Our algorithms significantly outperform the SaNGreeA algorithm due to Campan and Truta which is the leading algorithm for achieving anonymity in networks by means of clustering. We then devise secure distributed versions of our algorithms. To the best of our knowledge, this is the first study of privacy preservation in distributed social networks. Scope:

The main scope is to arrive at an anonymized view of the unified network without revealing to any of the data holders information about links between nodes that are controlled by other data holders. The study of anonymizing social networks has concentrated so far on centralized networks, i.e., networks that are held by one data holder. However, in some settings, the network data is split between several data holders, or players. For example, the data in a network of email accounts where two nodes are connected if the number of email messages that they exchanged was greater than some given threshold, might be split between several email service providers. As another example, consider a transaction network where an edge denotes a financial transaction between two individuals; such a network would be split between several banks. In such settings, each player controls some of the nodes (his clients) and he knows only the edges that are adjacent to the nodes under his control.

Based on these observations we focus on improving the performance of information collection from the neighborhood of a user in a social network and make the following contributions: We introduce sampling-based algorithms that given a user in a social network quickly obtain a near-uniform random sample of nodes in its neighborhood. We employ these algorithms to quickly approximate the number of users in a user’s neighborhood that have endorsed an item.

We introduce and analyze variants of these basic sampling schemes in which we aim to minimize the total number of nodes in the network visited by exploring correlations across samples. We evaluate our sampling-based algorithms in terms of accuracy and efficiency using real and synthetic data and demonstrate the utility of our approach. We show that our basic sampling schemes can be utilized for a variety of strategies aiming to rank items in a network, assuming that information for each user in the network is available.

ANONYMIZATION BY SEQUENTIAL CLUSTERING:

The sequential clustering algorithm for k-anonymizing tables was presented in [7]. It was shown there to be a very efficient algorithm in terms of runtime as well as in terms of the utility of the output anonymization. We proceed to describe an adaptation of it for anonymizing social networks. Algorithm 1 starts with a random partitioning of the network nodes into clusters.

The initial number of clusters in the random partition is set to N/k0 and the initial clusters are chosen so that all of them are of size k0 or k0+1, where k0 = αk is an integer and α is some parameter that needs to be determined. The algorithm then starts its main loop (Steps 2-4). In that loop, the algorithm goes over the N nodes in a cyclic manner and for each node it checks whether that node may be moved from its current cluster to another one while decreasing the information loss of the induced anonymization.

Algorithm:

Algorithm 1.

• Input: A social network SN, an integer k.
• Output: A clustering of SN into clusters of size ≥ k.

1) Choose a random partition C = {C1,...,CT} of V into T := N/k0 clusters of sizes either k0 or k0+1.
network data is split among M sites (or players) in the
Here we consider the distributed setting, in which the

THE DISTRIBUTED SETTING:

5) Player 1 computes \( a = \sum_{\ell=1}^{M} s_{\ell} \mod p \) and broadcasts
4) Players \( \ell, 2 \leq \ell \leq M \), send \( s_{\ell} \) to the player 1.
3) Player \( \ell, 1 \leq \ell \leq M \), computes \( s_{\ell} = \sum_{m=1}^{M} a_{m, \ell} \mod p \).
2) Player m selects \( M \) random share vectors \( a_{m, \ell} \in \mathbb{Z}_{d}^{p} \).
1) Player m selects \( M \) random share vectors \( a_{m, \ell} \in \mathbb{Z}_{d}^{p} \).

• Output: \( a = \sum_{m=1}^{M} a_{m} \).

Computing the sum of private integers has well known simple SMPs. The components of the vectors are rational numbers. The denominators of those numbers are common and known to all, but their numerators depend on private integers (those are the private integers that appear in the numerator). Hence, that problem reduces to computing sums of private vectors over the integers. Moreover, it is possible to compute upfront an upper bound \( p \) on the size of those integers and of their sum.

Algorithm 2. Secure computation of sums:

• Input: Each player \( m, 1 \leq m \leq M \), has a private input vector \( a_{m} \in \mathbb{Z}_{d}^{p} \).
• Output: \( a = \sum_{m=1}^{M} a_{m} \).
1) Player \( m \) selects \( M \) random share vectors \( a_{m, \ell} \in \mathbb{Z}_{d}^{p} \), \( 1 \leq \ell \leq M \), such that \( \sum_{\ell=1}^{M} a_{m, \ell} = \text{am} \mod p \).
2) Player \( m \) sends \( a_{m, \ell} \) to the \( \ell \)th player, for all \( 1 \leq \ell = m \leq M \).
3) Player \( \ell, 1 \leq \ell \leq M \), computes \( s_{\ell} = \sum_{m=1}^{M} a_{m, \ell} \mod p \).
4) Players \( \ell, 2 \leq \ell \leq M \), send \( s_{\ell} \) to the player 1.
5) Player 1 computes \( a = \sum_{\ell=1}^{M} s_{\ell} \mod p \) and broadcasts it.

2) For \( n = 1,\ldots,N \) do:
   a) Let \( C_{t} \) be the cluster to which \( v_{n} \) currently belongs.
   b) For each of the other clusters, \( C_{s} \), \( s = t \), compute the
difference in the information loss, \( \Delta n:ts \), if \( v_{n} \) would move
from \( C_{t} \) to \( C_{s} \).
   c) Let \( C_{s} \) be the cluster for which \( \Delta n:ts \) is minimal.
   d) If \( C_{t} \) is a singleton, move \( v_{n} \) from \( C_{t} \) to \( C_{s} \) and re-
move cluster \( C_{t} \).
   e) Else, if \( \Delta n:ts < 0 \), move \( v_{n} \) from \( C_{t} \) to \( C_{s} \).
3) If there exist clusters of size greater than \( k \), split
each of them randomly into two equally-sized clusters.
4) If at least one node was moved during the last loop,
go to Step 2.
5) While there exist clusters of size smaller than \( k \), se-
lect one of them and unify it with the cluster which is
closest.
6) Output the resulting clustering.

Explanation:

To illustrate the difference between the two scenarios,

1) Scenario A: Each player needs to protect the identi-
ties of the nodes under his control from other players,
as well as the existence or non-existence of edges ad-
Jacent to his nodes.
2) Scenario B: All players know the identities of all nodes
in \( V \); the information that each player needs to protect
from other players is the existence or non-existence of
edges adjacent to his nodes.

Assume that those players are banks, that the nodes
are accounts in those banks, and that the edges denote
financial transactions between the accounts. Here,
each node is identified by an account number, but the
bank is trusted to protect the identity of the clients
that hold those accounts. Hence, the square bank is
expected to hide the information that one of his clients
is a 62 year old female and the other is a 31 year old fe-
male (as indicated by the quasi identifier records (62,F)
and (31,F) next to his nodes in Figure 1) since that might
reveal the identity of the account holders. In addition,
the square bank is expected to hide from the circular
bank the internal transactions among his clients (there
is one such edge in the illustrated graph) or between
his clients and clients of the triangular bank (there
are two such edges in the graph). This is an example of
Scenario A. However, assume that the network is a cor-
respondence network between email addresses. Here
it is natural to assume that the identity of the nodes is
not confidential, since typical email addresses disclose
the name of the individual that holds them.
network data is split among M sites (or players) in the here we consider the distributed setting, in which the

4) Players ℓ, 2 ≤ ℓ ≤ M, send sℓto the player 1.

• Input: Each player m, 1 ≤ m ≤ M, has a private input

Algorithm 2. Secure computation of sums:

an upper bound p on the size of those integers and of

Integers. Moreover, it is possible to compute upfront

reduces to computing sums of private vectors over the

gers that appear in the numerator Hence, that problem

depend on private integers (those are the private inte

tional numbers The denominators of those numbers

simple SMPs. The components of the vectors are ra

lect one of them and unify it with the cluster which is

go to Step 2.

4) If at least one node was moved during the last loop,

eters.

each of them randomly into two equally-sized clus

2) For n = 1,...,N do:

ers: It requires a secure computation of the descrip

tive information loss (while in Scenario B such a com

putation can be made in a public manner); and the

players must hide from other players the allocation of

their nodes to clusters. Another research direction that

this study suggests is to devise distributed versions of

the k-anonymity algorithms in [15], [23], [31]; those al

gorithms might require different techniques than those

used here.

REFERENCES:


In ICDT, volume 3363 of LNCS, pages 246–258, 2005.


