International Journal ofComputing Communication and Information System(IJCCIS)
Vol 6. No.1 – Jan-March 2014 Pp. 14-28
©gopalax Journals, Singapore
available at : www.ijcns.com
ISSN: 0976–1349

CLASSIFIED INFORMATION PREVENTING FROM INTERFERENCE ATTACKS ON SOCIAL NETWORKS

G.Saraswathy, K.Kanchana
A.R.Engineering College,Villupuram
saraswathygurusamy@gmail.com klkanchammu@gmail.com

ABSTRACT

Online social networks, such as Facebook, are increasingly utilized by many people. These networks allow users to publish details about themselves and to connect to their friends. Some of the information revealed inside these networks is meant to be private. Yet it is possible to use learning algorithms on released data to predict private information. In this paper, we explore how to launch inference attacks using released social networking data to predict private information. We then devise three possible sanitization techniques that could be used in various situations. Then, we explore the effectiveness of these techniques and attempt to use methods of collective inference to discover sensitive attributes of the data set. We show that we can decrease the effectiveness of both local and relational classification algorithms by using the sanitization methods we described.

1. INTRODUCTION

SOCIAL networks are online applications that is used to users to connect by means of various link types. As part of their offerings, these networks allow people to list detailsabout themselves that are relevant to the nature of the network. For instance, Facebook is a general-use social network, so individual users list their favorite activities, books, and movies. Conversely, LinkedIn is a professional network; because of this, users specify details which are related to their professional life (i.e., reference letters, previous employment, and so on.)

Because these sites gather extensive personal information, social network application providers have a rare opportunity: direct use of this information could be useful to advertisers for direct marketing. However, in practice, privacy concerns can prevent these efforts. This conflict between the desired use of data and individual privacy presents an opportunity for privacy-preserving social network data mining that is, the discovery of information and relationships from social network data without violating privacy.

Privacy concerns of individuals in a social network can be classified into two categories: privacy after data release, and private information leakage. Instances of privacy after data release involve the identification of specific individuals in a data set subsequent to its release to the general public or to paying customers for a specific usage. Perhaps the most illustrative example of this type of privacy breach (and the repercussions thereof) is the AOL search data scandal.

In 2006, AOL released the search results from 650,000 users for research purposes. However, these results had a significant number of “vanity” searches—searches on an individual’s name, social security number, or address—that could then be tied back to a specific individual

Private information leakage, conversely, is related to details about an individual that are not explicitly stated, but, rather, are inferred through other details released and/ or relationships to individuals who may express that detail. A trivial example ofthis type of information leakage is a scenario where a user, say John, does not enter his political affiliation because of privacy concerns. However, it is publicly available that he is a member of the “legalize the same sex marriage.” Using this publicly available information regarding a general group membership, it is easily guessable what John’s political affiliation is. Somewhat less obvious is the favorite movie “The End of the Spear.1” Wenote that this is an issue both in live data (i.e., currently on the server) and in any released data.This paper focuses on the problem of private information leakage for individuals as a direct result of their actions as being part of an online social network. We model an attack scenario as follows: Suppose Facebook wishes to release data to electronic arts for their use in advertising games to interested people. However, once electronic arts has this data, they want to identify the political affiliation of users in their data for lobbying efforts. Because they would not only use the names of those individuals who explicitly list their affiliation, but also—through inference—could determine the affiliation of other users in their data, this would obviously be a privacy violation of hidden details.2 We explore how the online social network data could be used to predict some individual private detail that a user is notwilling to disclose (e.g., political or religious affiliation, sexual orientation) and explore the effect of possible data sanitization approaches on preventing such private infor-mation leakage, while allowing the recipient of the sanitized data to do inference on nonprivate details.

This problem of private information leakage could be an important issue in some cases. Recently, both ABC News [3] and the Boston Globe [4] published reports indicating that it is possible to determine a user’s sexual orientation by obtaining a relatively small subgraph from Facebook that includes only the user’s gender, the gender they are interested in, and their friends in that subgraph. Predicting an individual’s sexual orientation or some other personal detail may seem like inconsequential, but in some cases, it may create negative repercussions (e.g., discrimi-nation, and so on.). For example, using the disclosed social network data (e.g., family history, life style habits, and so on.), predicting an individual’s likelihood of getting Alzheimer disease for health insurance and employment purposes could be problematic.

Our Contributions

To the best of our knowledge, this is the first paper that discusses the problem of sanitizing a social network to prevent inference of social network data and then examines the effectiveness of those approaches on a real-world data set. In order to protect privacy, we sanitize both details and the underlying link structure of the graph. That is, we delete some information from a user’s profile and remove some links between friends. We also examine the effects of generalizing detail values to more generic values. We then study the effect these methods have on combating possible inference attacks and how they may be used to guide sanitization. We further show that this sanitization still allows the use of other data in the system for further tasks.

In addition, we discuss the notion of “perfect privacy” in social networks and give a formal privacy definition that is applicable to inference attacks discussed in this paper.

Overview

The remainder of this paper is organized as follows: we describe previous work in the area of social network anonymization, we describe the real-world data set that is used in our experiments. we describe, in detail, the learning methods that are used in our anonymization and classification tasks. In Section 4, we present our definition for privacy as well as describe the methods that we developed to anonymize social network data. In Section 5, we describe our experiments and the results we obtained. In Section 6, we suggest some possible future work in this area.

2. RELATED WORK

In this paper, we touch on many areas of research that have been heavily studied. The area of privacy inside a social network encompasses a large breadth, based on how privacy is defined. In Backstrom et al. consider an attack against an anonymized network. In their model, the network consists of only nodes and edges. Detail values are not included. The goal of the attacker is simply to identifypeople. Further, their problem is very different than the one considered in this paper because they ignore details and do not consider the effect of the existence of details on privacy.

Hay et al. and Liu and Terzi consider several ways of anonymizing social networks. However, our work focuses on inferring details from nodes in the network, not individually identifying individuals.

Other papers have tried to infer private information inside social networks.consider ways to infer private information via friendship links by creating a Bayesian network from the links inside a social network. While they crawl a real social network, LiveJournal, they use hypothetical attributes to analyze their learning algo-rithm. Also, compared to we provide techniques that can help with choosing the most effective details or links that need to be removed for protecting privacy. Finally, we explore the effect of collective inference techniques in possible inference attacks.

In Zheleva and Getoor propose several methods of social graph anonymization, focusing mainly on the idea that by anonymizing both the nodes in the group and the link structure, that one thereby anonymizes the graph as a whole. However, their methods all focus on anonymity in the structure itself. For example, through the use of k-anonymity or t-closeness, depending on the quasi-identi-fiers which are chosen, much of the uniqueness in the data may be lost. Through our method of anonymity preserva-tion, we maintain the full uniqueness in each node, which allows more information in the data postrelease.

In Gross et al. examine specific usage instances at Carnegie Mellon. They also note potential attacks, such as node reidentification or stalking, that easily accessible data on Facebook could assist with. They further note that while privacy controls may exist on the user’s end of the social networking site, many individuals do not take advantage of this tool. This finding coincides very well with the amount of data that we were able to crawl using a very simple crawler on a Facebook network. We extend on their work by experimentally examining the accuracy of some types of the demographic reidentification that they propose before and after sanitization.

The Facebook platform’s data has been considered in some other research as well. In Jones and Soltren crawl Facebook’s data and analyze usage trends among Facebook users, employing both profile postings and survey informa-tion. However, their paper focuses mostly on faults inside the Facebook platform. They do not discuss attempting to learn unrevealed details of Facebook users, and do no analysis of the details of Facebook users. Their crawl consisted of around 70,000 Facebook accounts.

The area of link-based classification is well studied. In Sen and Getoor compare various methods of link-based classification including loopy belief propagation, mean field relaxation labeling, and iterative classification. However, their comparisons do not consider. In Tasker et al. present an alternative classification method where they build on Markov networks. However, none of these papers consider ways to combat their classification methods.

In Menon and Elkan use dyadic data methods to predict class labels. We show later that while we do notexamine the effects of this type of analysis, the choice of technique is arbitrary for anonymization and utility.

In Zheleva and Getoor attempt to predict the private attributes of users in four real-world data sets: Facebook, Flickr, Dogster, and BibSonomy. They do not attempt to actually anonymize or sanitize any graph data. Instead, their focus is on how specific types of data, namely, that of declared and inferred group membership, may be used as a way to boost local and relational classification accuracy. Their defined method of group-based (as opposed to details-based or link-based) classification is an inherent part of our details-based classification, as we treat the group membership data as another detail, as we do favorite books or movies. In fact, Zheleva and Getoor work provides a substantial motivation for the need of the solution proposed in our work.

In Talukder et al. propose a method of measuring the amount of information that a user reveals to the outside world and which automatically determines which informa-tion (on a per-user basis) should be removed to increase the privacy of an individual.

Finally, in we do preliminary work on the effectiveness of our Details, Links, and Average classifiers and examine their effectiveness after removing some details from the graph. Here, we expand further by evaluating their effectiveness after removing details and links.

3 . LEARNING METHODS ON SOCIAL NETWORKS

3.1 Social Network Description

We begin by describing the specific composition of a social network for the purposes of our study. Definition 1. A social network is represented as a graph, G ¼ fV; E; Dg, where V is the set of nodes in the graph, where each node ni represents a unique user of the social network. E represents the set of edges in the graph, which are the links defined in the social network. For any friendship link Fi;j between user ni and user nj, we assume that both Fi;j 2 E and Fj;i2 E. D is the set of details from the social network.

Definition 2. A detail type is a string defined over an alphabet that represents a specific category name within the social network details set. The set of all detail types is represented by H. A detail value is a string defined over an alphabet that represents a user's input for a detail type. A detail is a (detail type, detail value) pair, represented uniquely by an identifier Jk. Dji is the jth (detail type, detail value) pair specified by the user ni. Di is the set of all Dji for a node ni. D is the set of Di for all i.

It is important to note that for any detail type, the expected response can either be single or multivalued, and that a user has the option of listing no detail values for any given detail. For example, consider Facebook's "home town" and "activities" detail type. A user can only have one home town, but can list multiple activities (for instance, soccer, reading, video games). However, a user also has the option of listing no detail values for these. For example, the detail value of "video games" for the detail type "activities" will be listed as (activities, video games), to distinguish it from other details that may have the same detail value, such as System architecture (groups, video games). Further, even if a user lists multiple activities, we store each independently in a detail with the corresponding detail name. That is, a user who enters "jogging" and "swimming" as his favorite activities will have the corresponding details (favorite activity, jogging) and (favorite activity, swimming).

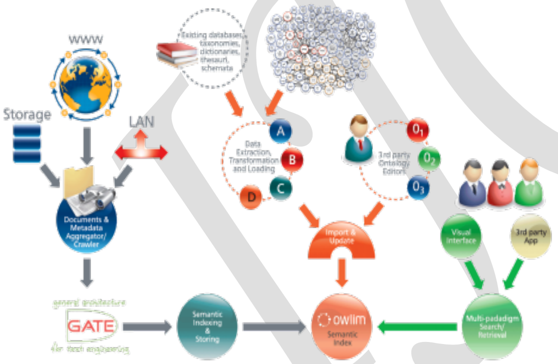
WWW
Storage
LAN
Existing databases, taxonomies, dictionaries, thesauri, schemata
Data Extraction, Transformation and Loading
A
B
C
D
3rd party Ontology Editors
O_1
O_2
O_3
Documents & Metadata Aggregator/ Crawler
Import & Update
Visual Interface
3rd party App
general architecture
GATE
for text engineering
Semantic Indexing & Storing
owlim
Semantic Index
Multi-padadigm Search/ Retrieval

Figure.1 System architecture

That is, from among four possible detail types (1), we define two detail types to be private, a person's political affiliation and their religion (2). Then, say we have two people, named Jane Doe and John Smith, respectively, (3) and (4). John Smith has specified that one of the activities he enjoys is fishing (6), which is also recorded as the fourth possible (detail type, detail value) pair. Also, John and Jane are friends (7). we have a reference for many frequently used notations found in the remainder of this paper.

Obviously, the detail types of I are varied based on an individual's choice. Generally, however, we consider a user's I to be any details that they do not specify. We use these detail types as our C in all classification methods. Further, for political affiliation, we consider only Clib and Ccons as possible class values—that is, "liberal" and"conservative." For sexual orientation, we consider Cheterosexual and Chomosexual as the possible class values.

To evaluate the effect that changing a person's details has on their privacy, we needed to first create a learning method that could predict a person's private details (for the sake of example, we assume that political affiliation is unspecified for some subset of our population). Since our goal is to understand the feasibility of possible inference attacks and the effectiveness of various sanitization techniques combating against those attacks, we initially used a simple naı¨ve Bayes classifier. Using naı¨ve Bayes as our learning algorithm allowed us to easily scale our implementation to the large size and diverseness of the Facebook data set. It also has the added advantage of allowing simple selection techniques to remove detail and linkinformation when trying to hide the classification a network node.Finally, it has shown itself to be extremely effective in these classification tasks.

Naïve Bayes Classification

A naive Bayes classifier is a simple probabilistic classifierbased on applying Bayes' theorem with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model". An overview of statistical classifiers is given in the article on Pattern recognition.The idea behind a Bayesian classifier is that, if an agent knows the class, it can predict the values of the other features. If it does not know the class, Bayes' rule can be used to predict the class given (some of) the feature values. In a Bayesian classifier, the learning agent builds a probabilistic model of the features and uses that model to predict the classification of a new example.

Naı¨ve Bayes on Friendship Links

Consider the problem of determining the class detail value of person ni given their friendship links using a naı¨ve Bayes model. Because there are relatively few people in the training set that have a friendship link to ni, the calculations become extremely inaccurate. Instead, we choose to decompose this relationship. Rather than having a link from person ni to nj, we instead consider the probability of having a link from ni to someone with nj's details.

Weighing Friendships

There is one last step to calculating P ðCxijN iÞ. In the specific case of social networks, two friends can be anything from acquaintances to close friends or family members. While there are many ways to weigh friendship links, the method we used is very easy to calculate and is based on the assumption that the more public details two people share, the more private details they are likely to share. This gives the following formula for Wi;j, which represents the weight of a friendship link from ni to node nj:

Network Classification

Collective inference is a method of classifying social network data using a combination of node details and connecting links in the social graph. Each of these classifiers consists of three components: a local classifier, a relational classifier, and a collective inference algorithm.

Local Classifiers

Local classifiers are a type of learning method that are applied in the initial step of collective inference. Typically, it is a classification technique that examines details of a node and constructs a classification scheme based on the details that it finds there. For instance, the naı¨ve Bayes classifier we discussed previously is a standard example of Bayes classification. This classifier builds a model based on the details of nodes in the training set. It then applies this model to nodes in the testing set to classify them.

Relational Classifiers

The relational classifier is a separate type of learning algorithm that looks at the link structure of the graph, and uses the labels of nodes in the training set to develop a model which it uses to classify the nodes in the test set. Specifically, in Macskassy and Provost examine four relational classifiers: class-distribution relational neighbor (cdRN), weighted-vote relational neighbor (wvRN), network-only Bayes classifier (nBC), and network-only link-based classi-fication (nLB).

The cdRN classifier begins by determining a reference vector for each class. That is, for each class, Cx, cdRN develops a vector RVx which is a description of what a node that is of type Cx tends to connect to. Specifically, RVxðaÞ is an average value for how often a node of class Cx has a link to a node of class Ca. To classify node ni, the algorithm builds a class vector, CVi, where CViðaÞ is a count of how often ni has a link to a node of class Ca. The class probabilities are calculated by comparing CVi to RVx for all classes Cx.

Collective Inference Methods

Unfortunately, there are issues with each of the methods described above. Local classifiers consider only the details of the node it is classifying. Conversely, relational classifiers consider only the link structure of a node. Specifically, a major problem with relational classifiers is that while we may cleverly divide fully labeled test sets so that we ensure every node is connected to at least one node in the training set, real-world data may not satisfy this strict requirement. If this requirement is not met, then relational classification will be unable to classify nodes which have no neighbors in the training set. Collective inference attempts to make up for these deficiencies by using both local and relational classifiers in a precise manner to attempt to increase the classification accuracy of nodes in the network. By using a local classifier in the first iteration, collective inference ensures that every node will have an initial probabilistic classification, referred to as a prior. The algorithm then uses a relational classifier to reclassify nodes. At each of these steps i > 2, the relational classifier uses the fully labeled graph from step i _ 1 to classify each node in the graph.

The collective inference method also controls the length of time the algorithm runs. Some algorithms specify a number of iterations to run, while others converge after a general length of time. We choose to use relaxation labeling as described in a method that retains the uncertainty of our classified labels. That is, at each step i, the algorithm uses the probability estimates, not a single classified label, from step i 1 to calculate new probability estimates. Further, to account for the possibility that there may not be a convergence, there is a decay rate, called set to 0.99 that discounts the weight of each subsequent iteration compared to the previous iterations.

We chose to use relaxation labeling because in the experiments conducted by Macs-kassy and Provost relaxation labeling tended to be the best of the three collective inference methods.

Each of these classifiers, including a relaxation labeling implementation, is included in NetKit-SRL.3 As such, after we perform our sanitization techniques, we allow NetKit to classify the nodes to examine the effectiveness of our approaches.

4. HIDING PRIVATE INFORMATION

Existing privacy definitions such as k-anonymity l-diversity and so on are defined for relational data only. They provide syntactic guarantees and do not try to protect against inference attacks directly. For example, k-anonym-ity tries to make sure that an individual cannot be identified from the data but does not consider inference attacks that can be launched to infer private information.

Recently developed differential privacy definition provides interesting theoretical guarantees. Basically, it guarantees that the result of a differential private algorithm are very similar with or without the data of any single user. In other words, differentially privacy guarantees that the change in one record does not change the result too much. On the other hand, this definition does not protect against the building of an accurate data mining model that can predict sensitive information. Actually many differentially private data mining algorithms have been developed that has similar accuracy to nondifferentially private versions. Since our goal is to release rich social network data set while preventing sensitive detail disclosure through data mining techniques, differential privacy definition is not directly applicable in our scenario.

To be able to formalize a privacy definition in our context, we need to address two issues with respect to an inference attack. First, we need to have some understanding of the potential prior information (i.e., background knowledge) the adversary can use to launch an inference attack. For example, if an adversary already knows all the hidden and unhidden private information related to the social network, it will be useless to try to protect against such an adversary. Second, we need to analyze the potential success of inference attack given the adversary's background information. For example, if the adversary has only the disclosed social network data, what is the best classifier he can build to predict sexual orientation?

Ideally, to address the first issue mentioned above, we may try to come up with a privacy definition that is successful against all possible background information. Unfortunately, this goal is not realistic in many privacy settings. As shown in it is impossible to provide "absolute" privacy guarantees with respect to all back-ground knowledge. In other words, it may not be possible tostop inference attacks against all background information. For example, if adversary has a background information stating that John's political affiliation is the same as the majority of people in Texas than any reasonable data release that preserve utility (e.g., data release that preserves aggregate statistics) can be used for an inference attack. For this reason, Dwork states the following observation: "In order to sidestep this issue we change from absolute guarantees about disclosures to relative ones..."

To address the second issue listed above, we need to estimate the performance of the best classifier that can be built by using the released social network data and the adversary's background knowledge. This is equivalent to estimating the Bayes error .In Clifton uses statistical learning theory to provide Bayess error bounds for classifiers that try to predict sensitiveinformation. Unfortunately, such bounds are not tight enough to use in practice.

Even though estimating Bayes error accurately is hard in general, it has been shown that certain classifiers such as k-nn and carefully constructed classifier ensembles provide good estimations for Bayes error. Therefore, in our privacy definition, we try to limit the success of an adversary with respect to a given set of classifiers. We believe that such set of classifiers would give a reasonable approximation of the Bayes error and provide good indication with respect to potential disclosure.

Due to the above reasons, we develop a relative privacy definition based on the difference in classification accuracy possible with and without the released social network data for a given background definition. We would like to stress that our privacy definition focuses on preventing inference attacks only and could be used with other definitions that tries to protect against other privacy attacks.

Formal Privacy Definition

Problem 1. Given a graph, G, from a social network, where I is a subset of H, and jIj _ 1, is it possible to minimize the classification accuracy on I when using some set of classifiers C while preserving the utility of H - I?

Definition 3. Background knowledge, K, is some data that is not necessarily directly related to the social network, but that can be obtained through various means by an attacker.
Examples of background knowledge in terms of a social network such as Facebook include voter registration, election results, phone book listings, and so on.

Definition 4. Classifier accuracy,Jcy, is the accuracy of a specific classifier, c, when used to classify based on detail name Jy, on data set G.

Definition 5. A graph is ð_; C; G; KÞ-private if, for a given set of classifiers C.

That is, if we have any set of given classifiers, C, then the classification accuracy of any arbitrary classifier c0 2 C when trained on K and used to classify G to predict sensitive hidden data is denoted by Pc0ðKÞ. Similarly, denotes the prediction accuracy of the classifier that is trained on both G and K. Here, _ denotes the additional accuracy gained by theattacker using G. Ideally, if ¼ 0, this means that the attacker does not gain additional accuracy in predicting sensitive hidden data.

The above privacy definition could be applied to other domains. Consider the scenario where we want to decide whether to release some private information (e.g., eating habits, lifestyle), and combined with some public information (e.g., age, zip code, cause of death of ancestors) or not. We may be worried that whether the disclosed information could be used to build a data mining model to predict the likelihood of an individual getting an Alzheimer's disease. Most individuals would consider such information to be sensitive for example, when applying for health insurance or employment. Our privacy definition could be used to decide whether to disclose the data set or not due to potential inference issues.

Manipulating Details

Clearly, details can be manipulated in three ways: adding details to nodes, modifying existing details and removing details from nodes. However, we can broadly classify these three methods into two categories: perturbation and anonymization. Adding and modifying details can both be considered methods of perturbation that is, introducing various types of "noise" into D to decrease.

Classification accuracies. Removing nodes, however, can be considered an anonymization method. Consider, for instance, the difference in two graphs, G0 and G00, which are sanitized versions of G by perturbation and anonymization methods, respectively. In G0, there are artificial details within D0. That is, suppose that there is a node ni 2 G, G0, G00 which has a listed detail of (favorite activities, sports) in our two sanitized data sets. When we consider this instance in G0, we are uncertain about its authenticity. Depending on the perturbation method used, the original node could have had no favorite activities, or had an entry of (favorite activities, dallas cowboys) which was altered to contain the aforementioned detail.

Choosing Details

We must now choose which details to remove. Our choice is guided by the following problem statement
Problem 2. Given G and a nonzero set of sensitive details I, determine the set of detail has the most reduction in classification accuracy for some set of classifiers C on the sensitive attributes I for the given number of removals m.
This allows us to find the single detail that is the most highly indicative of a class and remove it. Experimentally, we later show that this method of determining which details to remove provides a good method of detail selection.

Manipulating Link Information

The other option for anonymizing social networks is altering links. Unlike details, there are only two methods of altering the link structure: adding or removing links. we choose to evaluate the effects of privacy on removing friendship links instead of adding fake links.

Consider for determining detail type using friend-ship links. Also assume that there are two possible classes for a node, and the true class is C1. We want to remove links that increase the likelihood of the node being in class C1. Please note that we define a node to be in class.Therefore, we would like to maximize the value of as much as possible by removing links.

Detail Generalization

To combat inference attacks on privacy, we attempt to provide detail anonymization for social networks. By doing this, we believe that we will be able to reduce the value to an acceptablethreshold value that matches the desired utility/privacy tradeoff for a release of data.

Definition 6. A detail generalization hierarchy (DGH) is an anonymization technique that generates a hierarchical ordering of the details expressed within a given category. The resulting hierarchy is structured as a tree, but the generalization scheme guarantees that all values substituted will be an ancestor, and thus at a maximum may be only as specific as the detail the user initially defined.

To clarify, this means that if a user inputs a favorite activity as the Boston Celtics, we could have, as an example, the following DGH

Boston Celtics ! NBA ! Basketball:

This means that to completely anonymize the entry of “Boston Celtics” in a user’s details, we replace it with “basketball.” However, notice that we also have the option of maintaining a bit more specificity by replacing it instead with “NBA.” This hierarchical nature will allow us to programmatically determine a more efficient release anonymization, which hopefully ensures that we have a generalized network that is as near-optimal as possible. Our scheme’s guarantee, however, ensures that at no time will the value “Boston Celtics” be replaced with the value “Los Angeles Lakers.”

We obtain the DGH by referring to a domain authority who specializes in categorizing the specific detail value. For books and activities, we use Google directories. For groups, we use Facebook.

Alternately, we have some details, such as “Favorite Music” which do not easily allow themselves to be placed in a hierarchy. Instead, we perform detail value decomposition (DVD) on these details.

Definition 7. DVD is a process by which an attribute is divided into a series of representative tags. These tags do not necessarily reassemble into a unique match to the original attribute.

Thus, we can decompose a group such as “Enya” into {ambient, alternative, irish, new age, celtic} to describe the group. To obtain the tags, we refer to Last.fm for music and IMDb for movies.

We provide a general outline of the generalization process in Algorithm 1. At each step, we generalize each detail type by one level [Lines 3-5] by determining which attributes can be further generalized without complete removal and keep a list of the accuracy of this generalization. At the end of each round, we “permanently” store the individual detail type that provides the greatest privacy savings [Line 4]. When the changed graph, G00, meets the chosen privacy requirement, we consider it ready for release.

Algorithm 1. Generalize(_; G)

```
1: G0    G
2: while Classify(G) - Classify(G0Þ _ _ do
3: S    all details that can be further generalized
4: s    getHighestInfoGainAttrib(S)
5: Gen(s; G0)
6: end while
7: return G’
```

5. EXPERIMENTS

Data Gathering

We wrote a program to crawl the Facebook network to gather data for our experiments. Written in Java 1.6, the crawler loadeda profile, parsed the details out of the HTML, and stored the details inside a MySQL database. Then, the crawler loaded all friends of the current profile and stored the friends inside the database both as friend-ship links and as possible profiles to later crawl.

Because of the sheer size of Facebook’s social network, the crawler was limited to only crawling profiles inside the Dallas/Forth Worth (DFW) network. This means that if two people share a common friend that is outside the DFW network, this is not reflected inside the database. Also, some people have enabled privacy restrictions on their profile which prevented the crawler from seeing their profile details. The total time for the crawl was seven days.

Because the data inside a Facebook profile is free form text, it is critical that the input be normalized. For example, favorite books of "Bible" and "The Bible" should be considered the same detail. Further, there are often spelling mistakes or variations on the same noun.

The normalization method we use is based upon a Porter stemmer presented in . To normalize a detail, it was broken into words and each word was stemmed with a Porter stemmer then recombined. Two details that normalized to the same value were considered the same for the purposes of the learning algorithm.

Our total crawl resulted in over 167,000 profiles, almost 4.5 million profile details, and over 3 million friendship links. In the graph representation, we had one large central group of connected nodes that had a maximum path length of 16. Only 22 of the collected users were not inside this group.

We provide some general statistics of our Facebook data set, including the diameter mentioned above. Common knowledge leads us to expect a small diameter in social networks. Note that, although popular, not every person in society has a Facebook account and even those who do still do not have friendship links to every person they know. Additionally, given the limited scope of our crawl, it is possible that some connecting individuals may be outside the Dallas/Fort Worth area. This consideration allows us to reconcile the information presented in and our observed network diameter.we show the original class likelihood for those details which will be used as experimental class values.

Experimental Setup

In our experiments, we implemented four algorithms to predict the political affiliation of each user. The firstalgorithm is called "Details Only." This algorithm uses (9) to predict political affiliation and ignores friendship links. The second algorithm is called "Links Only." This algorithm uses (12) to predict political affiliation using friendship links and does not consider the details of a person. The third algorithm is called "Average."

We define two classification tasks. The first is that we wish to determine whether an individual is politically "conservative" or "liberal." The second classification task is to determine whether an individual is "heterosexual" or "homosexual." It is important to note that we consider individuals who would also be considered "bisexual" as "homosexual" for this experiment. We begin by pruning the total graph of 160,000 nodes down to only those nodes for which we have a recorded political affiliation or sexual orientation to have reasonable tests for the accuracy of our classifiers and the impact of our sanitization. This reduces our overall set size to 35,000 nodes for our political affiliation tests and to 69,000 nodes for our sexual orientation tests.

We then conduct a series of experiments where we remove a number of details and a separate series of experiments where we remove a number of links. We conduct these removing up to 20 details and links, respectively.

Local Classification Results

We show the details that most indicate the "homosexual" classification. In contrast to political affiliation, there are no single details which are very highly correlated with that classification. For example, the three details we have selected here are more highly indicative ofbeing "Liberal" than of being "homosexual" that there are a few categories that are very highly representative of the "heterosexual" classification.

Detail Removal

As can be seen from the results, our methods are generally successful at reducing the accuracy of classification tasks. Fig. 1 shows that removing the details most highly connected with a class is accurate across the details and average classifiers. Counter-intuitively, perhaps, is that the accuracy of our links classifier is also decreased as we remove details. However, as discussed in Section 4.4, the details of two nodes are compared to find a similarity. As we remove details from the network, the set of "similar" nodes to any given node will also change. This can account for the decrease in accuracy of the links classifier.

Additionally, we see that in there is a severe drop in the classification accuracy after the removal of a single detail. However, when looking at the data, this can be explained by the removal of a detail that is very indicative of the “conservative” class value. When we remove this detail, the probability of being “conservative” drastically decreases, which leads to a higher number of incorrect classifications. When we remove the second detail, which has a similar likelihood for the “Liberal” classification, then the class value probabilities begin to trend downward at a much smoother rate.

While we do not see this behavior in we do see a much more volatile classification accuracy. This appears to be as a result of the wider class size disparity in the underlying data. Because approximately 95 percent of the available nodes are “heterosexual” and there are not details that are as highly indicative of sexual orientation as there are of political affiliation, even minor changes can affect the classification accuracy in unpredictable.
when we remove five details, we have lowered the classification accuracy, but for the sixth and seventh details, we see an increase in classification accuracy. Then, we again see another decrease in accuracy when we remove the eighth detail.

Link Removal

When we remove links, we have a generally more stable downward trend, with only a few exceptions in the “political affiliation” experiments.

Combined Removal

While each measure provides a decrease in classification accuracy, we also test what happens in our data set if we remove both details and links. To do this, we conduct further experiments where we test classification accuracy after removing 0 details and 0 links (the baseline accuracy), 0 details and 10 links, 10 details and 0 links, and 10 details and 10 links. We choose these numbers because after removing 12 links, we found that we were beginning to create a number of isolated groups of few nodes or single, disconnected nodes. Additionally, when we removed 13 details, 44 percent of our “political affiliation” data set and 33 percent of our “sexual orientation” data set had fewer than four details remaining. Since part of our goal was to maintain utility after a potential data release, we chose to remove fewer details and links to support this.
We refer to these sets as 0 details, 0 links; 10 details, 0 links; 0 details, 10 links; 10 details, 10 links removed, respectively. Following this, we want to gauge the accuracy of the classifiers for various ratios of labeled versus unlabeled graphs. To do this, we collect a list of all of the available nodes, as discussed above. We then obtain a random permutation of this list using the Java function built-in to the collections class. Next, we divide the list into a test set and a training set, based on the desired ratio.

We focus on multiples of 10 for the accuracy percentages, so we generate sets of 10=90; 20=80; . . . ; 90=10. Additionally, when creating training sets for our “sexual orientation” data set, because of the wide difference in the group size for “heterosexual” and “homosexual,” we make sure that weseparate out the chosen percentage from the known “heterosexual” and “homosexual” groups independently to make sure that we have a diversity in both our training and test sets. For example, in a test where we will have only 10 percent labeled data, we select 10 percent of heterosexual individuals and 10 percent of homosexual individuals independently to be in our training set.

We refer to each set by the percentage of data in the test set. We generate five test sets of each ratio, and run each experiment independently. We then take the average of each of these runs as the overall accuracy for that ratio.

It show the results of our classification methods for various labeled node ratios. These results indicate that the average and details classifiers generally perform at approximately the same accuracy level. The Links Only classifier, however, generally performs significantly worse except in the case where 10 details and no links are removed. In this situation, all three classifiers perform similarly. We see that the greatest variance occurs when we remove details alone. It may be unexpected that the Links Only classifier has such varied accuracies as a result of removing details, but since our calculation of probabilities for that classifier uses a measure of similarity between people, the removal of details may affect that classifier.

Using SVMs

Additionally it show the result of using SVMs as a classification technique. To obtain these results, we use the SVM classifier packaged in WEKA, after representing details as a bitstring. We see here that when we remove no details, the classification accuracy of the SVM has a classification accuracy between our Links Only and Average/Details Only classifiers, with the exception of sets where the graph has a large percentage of unknowns (80 and 90 percent of the graph is unknown) where the SVM classifier can actually outperform the Details Only/Average classifier. However, once we remove details (see Figs. 2d and 3d), the classification accuracy of the SVM drops much further than the Average/Details Only classifier, and even performs worse than the Links Only classification method.

Next, we examine the effects of removing the links. We remove K links from each node, where K 2 ½0; 10&, and again partition the nodes into a test set and training set of equal size. We then test the accuracy of the local classifier on this test set. We repeat this five times and then take the average of each accuracy for the overall accuracy of each classifier after K links are removed. For K 2 ½1; 6&, each link removal steadily decreases the accuracy of the classifier. Removing the seventh link has no noticeable effect, and subsequent removals only slightly increase the accuracy of the links only classifier. Also, due to space limitations, for the remainder of experiments we show only the results of the average classifier, as it is generally the best of the three classifiers.

When we again examine the performance of the SVM, we see similar results to what was seen with details only and average. Since the SVM does not include the link structure in its classification, there is no real affect from removing links on this classification method.

It is important to note that the sexual orientation classifier seems to be more susceptible to problems of incomplete knowledge. We can see in each subfigure, to a far greater degree than in Fig. 2, that as we decrease the amount of information available to the training method the sexual orientation classifier accuracy decreases considerably. Once again, we believe that this may be explained simply by the fact that there is far less support of the “homosexual” classification, and as such, is consider-ably harder to classify on without adequate data. Specifically, since there are so few instances of the “homosexual” classification in our data set, when you combine this with the fact that there are no absolute predictors of homosexuality and that the indicators for homosexuality have a very low increased likelihood, if most of the examples ofhomosexuals are unknown, then classifiers are going to be unable to create an accurate model for prediction.

However, we show that by applying our technique, we routinely restrict classification accuracy to some arbitrary value below 95 percent. As we mentioned this means that graph is effectively private because an attacker would be forced to use only K to determine classification labels.

Generalization Experiments

Each detail falls into one of several categories: religion, political affiliation, activities, books, music, quotations, shows/movies, and groups. Due to the lack of a reliable subject authority, that is, a source who could definitively categorize a given quotation without additional human input, quotations were discarded from all experiments. To generate the DGH for each activity, book, and show/movie, we used Google directories. To generate the DVD for Music, we used the Last.fm tagging system. To generate the hierarchy for Groups, we used the classification criteria from the Facebook page of that group.

To account for the free-form tagging that Last.fm allows, we also store the popularity for each tag that a particular detail has. Last.fm indicates this through the presentation of tags on the page. The font size for a tag is representative of how many users across the system have defined that particular tag for the music type. We then keep a list of tag recurrence (weighted by strength) for each user. For Music anonymization, we eliminate the lowest scoring tags.
In our experiments, we assume that the trait “political affiliation” is a sensitive attribute that the data owner prefers to hide. Our C includes a naive Bayes classifier and the implementation of SVM from Weka.

We present findings from our domain generalization. We present a comparison of simply using K to guess the most populated class from background knowledge, the result of generalizing all trait types, generalizing no trait types, and when we generalize the best single performing trait type (activities).

We see here that our method of generalization (seen through the All and Activities lines) does indeed decrease the accuracy of classification on the data set. Interestingly, while previous work indicates that group membership is the dominant detail in classification, we see the most benefit here from generalizing only the Activities detail. We believe that this is due to the fact that Activities generally have a far larger range of generalization values, because the trees for these detail types are taller than those of groups.

Next, we show that given a desired increase in , we are able to determine what level to anonymize the data set to. We see from that as we require less privacy from our anonymized graph, fewer categories are generalized to any degree. We also see that Groups is most consistently anonymized completely until the required privacy allowance is 20 percent. Further, we see that the most variable entry is music.
This may be because the nature of the music detail is that it allows us more easily to include or remove details to fit a required privacy value. Rather than, say, the activities detail type, which has a fixed hierarchy, music has a loosely collected group of tags, which we can more flexibly include.

Collective Inference Results

We note that in the Facebook data, there are a limited number of “groups” that are highly indicative of an individual’s political affiliation. When removing details, these are the first that are removed. We assume that conducting the collective inference classifiers after removing only one detail may generate results that are specific for the particular detail we classify for. For that reason, we continue to consider only the removal of 0 details and 10 details, the other lowest point on the classification accuracy. We also continue to consider the removal of 0 links and 10 links due to the marginal difference between the ½6; 7& region and removing 10 links.

For the experiments using relaxation labeling, we took the same varied ratio sets generated previously. For each, we store the predictions made by the details only, links only, and average classifiers and use those as the priors for the NetKit toolkit. For each of those priors, we test the final accuracy of the cdRN, wvRN, nLB, and nBC classifiers. We do this for each of the five sets generated for each of the four points of interest. We then take the average of their accuracies for the final accuracy.

Macskassy and Provost study the effects of collective inference on four real-world data sets: IMDB, CORA, WebKB, and SEC filings. While they do not discuss the difference in the local classifier and iterative classification steps of their experiments, their experiments indicate that Relaxation Labeling almost always performs better than merely predicting the most frequent class. Generally, it performs at near 80 percent accuracy, which is an increase of approximately 30 percent in their data sets. However, in our experiments, Relaxation Labeling typically performed no more than approximately 5 percent better than predicting the majority class for political affiliation. This is also substantially less accurate than using only our local classifier. We believe that this performance is at least partially because our data set is not densely connected.
Our results indicate that there is very little significant difference in the collective inference classifiersexcept for cdRN, which performs significantly worse on data sets where there is a small training set. These results also indicate that our Average classifier consistently out-performs relaxation labeling on the pre- and post anonymized data sets.

We see that while the local classifier’s accuracy is directly affected by the removal of details and/or links, this relationship is not shown by using relaxation labeling with the local classifiers as a prior. Relational classifier portion of the graph remains constant, only the local classifier accuracy changes. From these, we see that the most “anonymous” graph, meaning the graph structure that has the lowest predictive accuracy, is achieved when we remove both details and links from the graph.

Effect of Sanitization on Other Attack Techniques

We further test the removal of details as an anonymization technique by using a variety of different classification algorithms to test the effectiveness of our method. For each number of details removed, we began by removing the indicated number of details in accordance with the method as described in Section 4. We then performed tenfold cross validation on this set 100 times, and conduct this for 0-20 details removed. The results of these tests are shown in Figs. 6a and 6b. As can be seen from these figures, our technique is effective at reducing the classification of networks for those details which we have classified as sensitive.

While the specific accuracy reduction is varied by the number of details removed and by the specific algorithm used for classification, we see that we do in fact reduce the accuracy across a broad range of classifiers. We see that linear regression is affected the least, with approximately a 10 percent reduction in accuracy. Also that decision trees are affected the most, with a roughly 35 percent reduction in classification accuracy.

International Journal ofComputing Communication and Information System(IJCCIS)
Vol 6. No.1 – Jan-March 2014 Pp. 14-28
©gopalax Journals, Singapore
available at : www.ijcns.com
ISSN: 0976–1349

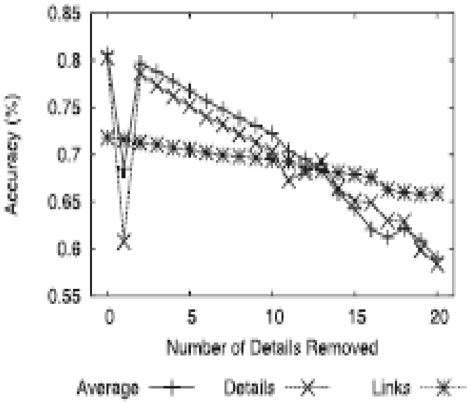

Figure. 2Political affiliation

This indicates that by using a Bayesian classifier to perform sanitization, which makes it easier to identify the individual details that make a class label more likely, we can decrease the accuracy of a far larger set of classifiers.We also see similar results with our generalization method in Figs. 6c and 6d. While the specific value of privacy which was defined for naive Bayes does not exactly hold, we still see that by performing generalization, we are able to decrease classification accuracy across multiple types of classifier.

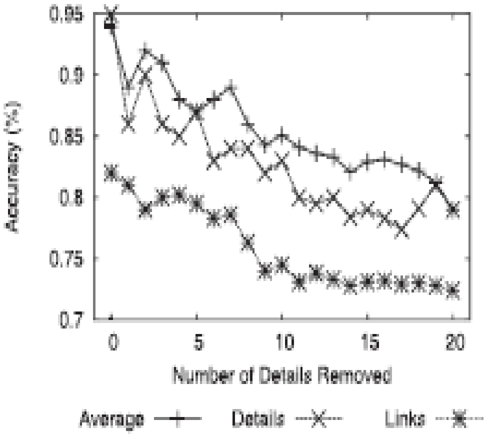

Figure.3 Sexual orientation

Effect of Sanitization on Utility

Of course, if the data is to be used by a company, then there must still be some value in the postsanitization graph. Of course, because utility in this system is difficult to know a priori, we show here, empirically, that we maintain the ability of the data to host various inference tasks on nonsensitive attributes after anonymization. We also show that by assuming the independence of details during removal, we maintain utility in later usage by minimizing the number of required deletions.

To gauge the utility of the anonymized data set, we show the results of various inference tests performed on nonsensitive details. For these tests, we used an SVM and our Bayesian classifiers, as discussed earlier, to run inference and collective inference tasks on each selected detail. We perform each test on random subgraphs of diameter n. This is to determine whether the effects are observed when a data miner is able to obtain only smaller sections of the social graph. To create these subgraphs, we choose a random node and include all neighbors up to n degrees away. We performed each test with 50 percent of the data in a training set and 50 percent in a test set, randomly chosen. For each n, we repeat the experiment 100 times. Here, we report the accuracy from running the experiments from the best performing classifiers, which were the average Bayesian with nLB. Accuracies are presented as the ratio of correct predictions to incorrect predictions, averaged across all experiments for each n.

The postsanitization figures were performed after removing ten details and 10 links. Please note that this figure represents the accuracy of a classifier on these details, not what percentage of the graph has this detail. We tested a selection of details with multiple attributes. For example, the “like video games” detail value is specified specifically in the data set (in Favorite Activities). College-educated was specified as a level of education (scanned fortype of degree and school). “Like to read” and “like sports” were inferred from the existence of books in the “favorite book” category or the existence of a sports team in the “favorite activities” category.

It is important to note, obviously, that when we perform inference on details such as “likes to read” we do not consider any detail of the type “favorite book¼ .” These are discarded for the tests on that type of classification task.Further, the test sets had a wide variety of representative sizes. “Like to read” had

148,000 profiles, while "like video games" had only 30,000.

As we can see from these results, the sanitization has minimal impact on the accuracy of a classifier on non-sensitive details. In fact, for the "college educated" and "like video games" details, the sanitization method improved classification accuracies by a small percentage. The apparent reason for this is that the details that are representative of nonsensitive attributes and those that are representative of our sensitive attributes are very disjoint. Recall from Table 4 that the group "legalize same sex marriage" is highly indicative that a member is liberal. However, this does not translate to any of the tested details. Instead, groups like "1 pwn j00 n h4l0" are indicative of video game players, "i'm taking up money to buy SEC refs glasses" is indicative of sports fans, and so on.

We do see, however, that by considering only limited areas of the social network, we vastly decrease the performance of a classifier, regardless of the classification task. As such, an attacker will most likely attempt to gain as much information from within the network as possible.

It should be noted, however, that the attribute "favorite book ¼ the bible" was removed from this test set, as it is highly indicative of one being a conservative.

6. CONCLUSION AND FUTURE WORK

We addressed various issues related to private information leakage in social networks. We show that using both friendship links and details together gives better predict-ability than details alone. In addition, we explored the effect of removing details and links in preventing sensitive information leakage. In the process, we discovered situations in which collective inferencing does not improve on using a simple local classification method to identify nodes. When we combine the results from the collective inference implications with the individual results, we begin to see that removing details and friendship links together is the best way to reduce classifier accuracy. This is probably infeasible in maintaining the use of social networks. However, we also show that by removing only details, we greatly reduce the accuracy of local classifiers, which give us the maximum accuracy that we were able to achieve through any combination of classifiers.

We also assumed full use of the graph information when deciding which details to hide. Useful research could be done on how individuals with limited access to the network could pick which details to hide. Similarly, future work could be conducted in identifying key nodes of the graph structure to see if removing or altering these nodes can decrease information leakage.

REFERENCES

[1]. Facebook Beacon, 2007.

[2]. T. Zeller, "AOL Executive Quits After Posting of Search Data," The New York Times, no. 22, http://www.nytimes.com/2006/08/22/ technology/22iht- aol.2558731.html?pagewanted=all &_r=0, Aug. 2006.

[3]. K.M. Heussner, "'Gaydar' n Facebook: Can Your Friends Reveal Sexual Orientation?" ABC News, http://abcnews.go.com/Technology/gaydar-facebook-friends/story?id=8633224#. UZ939UqheOs, Sept. 2009.

[4]. C. Johnson, "Project Gaydar," The Boston Globe, Sept. 2009.

[5]. L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore Art Thou r3579x?: Anonymized Social Networks, Hidden Patterns, and Structural Steganography," Proc. 16th Int'l Conf. World Wide Web (WWW '07), pp. 181-190, 2007.

[6]. M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava, "Anonymizing Social Networks," Technical Report 07-19, Univ. of Massachusetts Amherst, 2007.

[7]. K. Liu and E. Terzi, "Towards Identity Anonymization on Graphs," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '08), pp. 93-106, 2008.

[8]. J. He, W. Chu, and V. Liu, “Inferring Privacy Information from Social Networks,” Proc. Intelligence and Security Informatics, 2006.

[9]. E. Zheleva and L. Getoor, “Preserving the Privacy of Sensitive research has been supported by grants from US NSF, AFOSR, ONR, Relationships in Graph Data,” Proc. First ACM SIGKDD Int’l Conf. NSA, and NIH. Some of his research work has been covered by media Privacy, Security, and Trust in KDD, pp. 153-171, 2008.

[10]. R. Gross, A. Acquisti, and J.H. Heinz, “Information Revelation received two best paper awards. He is a recipient of US NSF Career and Privacy in Online Social Networks,” Proc. ACM Workshop Award and Purdue CERIAS Diamond Award for academic excellence.

[11]. H. Jones and J.H. Soltren, “Facebook: Threats to Privacy,” technical report, Massachusetts Inst. of Technology, 2005.

[12]. P. Sen and L. Getoor, “Link-Based Classification,” Technical Report CS-TR-4858, Univ. of Maryland, Feb. 2007.

[13]. B. Tasker, P. Abbeel, and K. Daphne, “Discriminative Probabilistic Models for Relational Data,” Proc. 18th Ann. Conf. Uncertainty in Artificial Intelligence (UAI ’02), pp. 485-492, 2002.

[14]. A. Menon and C. Elkan, “Predicting Labels for Dyadic Data,” Data Mining and Knowledge Discovery, vol. 21, pp. 327-343, 2010.

[15]. E. Zheleva and L. Getoor, “To Join or Not to Join: The Illusion of papers, and 12 books. She has three US patents and has given Proc. IEEE 26th Int’l Conf. Data Eng. Workshops (ICDE ’10), pp. 266-269, 2010.