Social Network Privacy for Attribute Disclosure Attacks

Sean Chester, Gautam Srivastava
CS Department, University of Victoria, CANADA, V8W 3P6
{schester,gsrivast}@uvic.ca

***Abstract*—Increasing research on social networks stresses the urgency for producing effective means of ensuring user privacy. Represented ubiquitously as graphs, social networks have a myriad of recently developed techniques to prevent identity disclosure, but the equally important attribute disclosure attacks have been neglected.**

To address this gap, we introduce an approach to anonymize social networks that have labeled nodes, α-nearness, which requires that the label distribution in every neighbourhood of the graph be close to that throughout the entire network. We present an effective greedy algorithm to achieve α-nearness and experimentally validate the quality of the solutions it derives.

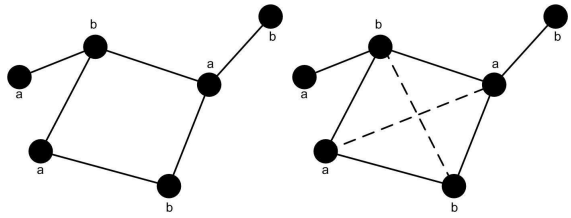
I. Introduction

Online communities and social networking sites are on the rise. With so many different communities (Facebook, Twitter, mySpace, PatientsLikeMe, etc.), each with so many users, the latent information opportunities for data miners are immense. Scientists, online marketing firms, and retailers can all gain invaluable knowledge about their target demographics from the analysis of these social networks. But with the quest for this knowledge comes the struggle to also protect the privacy of users. Is there a way to allow data miners access to our online lives while still protecting information we wish to keep private?

To date there has been copious research on protecting the identity of users from adversaries who exploit structural properties of social network graphs. However, another very important class of attacks—that of identifying just the sensitive information rather than the identity of users—has been mostly ignored. In this paper we look to answer the following question: Given a social network, can we protect against an adversary who uses certain targeted nodes within the network to gain sensitive information about the friends of those nodes? Consider the example of Facebook, where this attack model can be quite effective. User u may often get many unknown *friend requests*, which, if accepted, create friendship links to his account. By establishing these links, an adversary then gains access to the network of u and to any sensitive information that a friend of u has made accessible to *friends of friends*, which now includes the adversary.

We introduce a new measure of anonymization called α-nearness to capture the susceptibility of a network to this type of attribute disclosure attack. A danger exists when particular neighbourhoods have a vastly different distribution of a given sensitive attribute than does the network as a whole, because an adversary can then conclude with greater confidence the values for that attribute of the neighbours of a targeted node. To protect against this attack on a given vertex-labeled social network G, we require that the distribution of labels within the neighbourhood of each node in the graph be within α of the overall probability distribution of the labels in the graph.

Consider the following example:



(a) A small social network

(b) The network transformed to be $(.1)$-near

Fig. 1. Alpha-Nearness Example

Example 1: Figure 1(a) depicts a labeled graph with 6 nodes. The probability distribution of label ***a*** in the graph is $p_a = 0.5$. The label sequences of the 3 nodes labeled with ***b*** are $(b, a, a, a), (b, a, a), (b, a)$. If an adversary were to connect to one of these 3 nodes, in such exposing their neighbours to him, he could conclude that the neighbors have label ***a*** with $p_a = 0.75$, 0.67, and 0.5, respectively. Only the third node's neighbourhood has the same distribution of **a** labels as does the overall graph.

In Figure 1(b), 2 dotted edges have been added to the graph, which changes the label distributions of each neighbourhood. Now, instead, the label sequences are $(b, b, a, a, a), (b, b, a, a), (b, a)$, and label distribution in each neighbourhood of label ***a*** is $p_a = 0.6$, 0.5, and 0.5, very closely matching the overall distribution of the labels in the graph. No matter which of the neighbourhoods the adversary can identify, he cannot refine his prediction of the label of any node in that neighbourhood by more than 0.1.

These labels can correspond to very sensitive information. Consider the example of the online social media site PatientsLikeMe. Members of this online

community get a chance to connect and share information with other patients suffering life-changing diseases. This information could be vital in the study of Disease Research. However, can we ensure that patients' sensitive information, in this case a disease, can be protected while allowing the study of such vital information?

In this paper, we make the following contributions:

- We propose a novel and advanced notion of data anonymization called alpha-nearness that protects against attribute disclosure attacks.
- We provide an algorithm that modifies a vertex-labeled graph G, so as to ensure it is α-near.
- We illustrate empirically that the algorithm can transform a graph into an α-near graph with a quite conservative number of additional edges.

We conclude the introduction with related work and give definitions in §II. We present an algorithm in §III and an experimental discussion in §IV. Finally, we prescribe future work in §V and conclude in §VI.

Related Work

Li et al. [4] state that there are two types of privacy attack for data, namely identity disclosure and attribute disclosure. Identity disclosure often leads to attribute disclosure. Identity disclosure occurs when an individual is identified within a dataset, whereas attribute disclosure occurs when information that an individual wished to keep private is identified. In this paper, we consider the latter category.

With its roots in tables, data privacy took form with anonymization schemes centered around the notion of k-anonymity, wherein each row in a table must be identical to and therefore indistinguishable from at least $k-1$ other rows. Work by Meyerson and Williams [7] and by Agarwal et al. [1] set the foundations of k-anonymity for tables showing the problem was NP-hard even for a reduced alphabet size. Table privacy beyond k-anonymity followed with a new notion called l-diversity [6], wherein each k-anonymous equivalence class required l different values for each sensitive attribute. In this way, l-diversity looked to not only protect identity disclosure, but was also the first attempt to protect against attribute disclosure. To address the shortcomings of l-diversity, Li et al. [4] introduced t-closeness, which requires that the distribution of attribute values within each k-anonymous equivalence class needs to be close to that of the attribute's distribution throughout the entire table. Our work here is similar in spirit, but t-closeness cannot be clearly applied to social networks.

For graph data, anonymization has followed a similar path, starting from a simple version of k-anonymization [5], leading to more sophisticated versions of anonymity [2], [3], [8], [9], [10]. Our work is complementary to much of this research, because we protect against attribute rather than identity disclosure.

Recent work by Zheleva and Getoor [11] illustrates how easily information about users of many well known social media sites can be recovered. Zhou and Pei consider attribute disclosure attacks by defining l-diversity for graphs [12]; however, this notion suffers the same weaknesses as does l-diversity for tabular data. Thus the urgent need for research like what we present here.

II. Preliminaries

Before detailing our notion of anonymity to protect against attribute disclosure, we formalize the attack. Specifically, we are assuming that the social network is an undirected, simple, vertex-labeled graph:

Definition 2.1 (labeled social network): A *labeled social network* is a graph $\mathrm{G} = (\mathrm{V}, \mathrm{E}, \mathcal{L}, \ell)$, where V is a vertex set, $\mathrm{E} \subseteq \mathrm{V} \times \mathrm{V}$ is the edge set, $\mathcal{L}$ is an alphabet of labels, and $\ell = \mathrm{V} \mapsto \mathcal{L}$ is a labeling function that assigns a label $l \in \mathcal{L}$ to each vertex in V For simplicity, $(u, v) \in \mathrm{E} \rightarrow (v, u) \in \mathrm{E}$. We assume for convenience that the elements of $\mathcal{L}$ are ordered.

For a particular node, its label sequence is the collection of labels of itself and all its friends:

Definition 2.2 (label sequence neighbourhood): The *label sequence neighbourhood* of a vertex $v \in \mathrm{V}$, denoted $\eta(v)$, is the set of vertices used to make up the *label sequence* of v. That is to say, $\eta(v) = \{v\} \cup \{u \in \mathrm{V} : (u, v) \in \mathrm{E}\}$.[1]

In terms of attribute sensitivity and information disclosure, it is important to consider the distribution of labels over a set of vertices:

Definition 2.3 (label distribution): For $\mathrm{W} \subseteq \mathrm{V}$, let $\mathrm{count}(l_i, \mathrm{W})$ denote the number of vertices in the set W which have label l_i. Then, the *label distribution* over W, denoted $\mathrm{distr}(\mathrm{W})$, is the vector $\frac{\langle \mathrm{count}(l_1,\mathrm{W}),\ldots,\mathrm{count}(l_{|\mathcal{L}|},\mathrm{W})\rangle}{|\mathrm{W}|}$. Also, the distribution of a particular label, l_i is one element of the vector, $\mathrm{count}(l_i, \mathrm{W})/|\mathrm{W}|$.

Then, given two distributions $\mathrm{distr}(\mathrm{W}_i)$, $\mathrm{distr}(\mathrm{W}_j)$, we define a distance measure:

Definition 2.4 (distance between two distributions): The distance between two distributions, $\mathrm{distr}(\mathrm{W}_i)$, $\mathrm{distr}(\mathrm{W}_j)$, denoted $\delta(\mathrm{distr}(\mathrm{W}_i), \mathrm{distr}(\mathrm{W}_j))$, is simply the sum of the differences between all but their last elements. For example, the distributions $\langle .7, .2, .1\rangle$ and $\langle .2, .4, .4\rangle$ have a distance of .5+.2=.7.

Given these definitions, we can now formally describe the adversary's attack. By identifying the label sequence of a node v in a labeled social network, an adversary gains knowledge about the labels of all the nodes in the neighbourhood of v. Whereas beforehand, he could only surmise the probability that their label is l_i to be $\mathrm{count}(l_i, \mathrm{V})/|\mathrm{V}|$, he now knows that the probability is closer to $\mathrm{count}(l_i, \eta(v))/|\eta(v)|$. That is:

[1]Note that this definition is equivalent to a traditional neighbourhood iff every vertex has a self-edge, an alternative formulation that we exploit for simplicity in our implementation.

Definition 2.5 (NAD attack): A *neighbourhood attribute disclosure attack* against node v is one in which the adversary discovers a more refined estimate of $\mathrm{distr}(\eta(v))$ than he had when he only knew $\mathrm{distr}(\mathrm{V})$. His knowledge gain is $\delta(\mathrm{distr}(\mathrm{V}), \mathrm{distr}(\eta(v)))$.

To combat this type of attack, we define a new measure of anonymity:

Definition 2.6 (α-Nearness): The neighbourhood of a vertex v is said to be *α-near* if $\delta(\mathrm{distr}(\eta(v)), \mathrm{distr}(\mathrm{V})) \leq \alpha$. A labeled social network G is *α-near* if the neighbourhood of every vertex in V is *α-near*.

III. An Algorithm to Produce α-Near Graphs

Here we describe a greedy algorithm to compute an α-near graph G' from an input graph G by augmenting the edge set with elements of $(\mathrm{V}\times\mathrm{V})\setminus\mathrm{E}$. For simplicity of discussion, we assume the labels are binary (i.e., $|\mathcal{L}| = 2$), but the ideas generalize quite readily.

Algorithm 1 Greedy α-Nearness

1: **while** graph is not α-near **do**
2: Partition V into $\mathrm{V}_{a->a}$, $\mathrm{V}_{a->b}$, $\mathrm{V}_{b->a}$, $\mathrm{V}_{b->b}$, and $\mathrm{V}_{\alpha-\mathrm{near}}$
3: Greedily add edges among vertices in $\mathrm{V}_{a->a}$ and among vertices in $\mathrm{V}_{b->b}$
4: Greedily add edges from vertices of $\mathrm{V}_{a->b}$ to vertices of $\mathrm{V}_{b->a}$
5: **if** graph is not α-near **then**
6: Add a random new edge from $(\mathrm{V}\times\mathrm{V})\setminus\mathrm{E}$
7: **end if**
8: **end while**

The algorithm relies on two intuitions. First, $\mathrm{K}_{|\mathrm{V}|}$ is α-near. So, given that the algorithm adds an edge on every iteration, it always progresses towards a solution in at most $(\mathrm{V}\times\mathrm{V})\setminus\mathrm{E}$ steps. Second, by partitioning the vertex set based on the label of a vertex and the label to which it needs be connected to be α-near, we construct partitions of mutually beneficial vertices. That is to say, all vertices in $\mathrm{V}_{a->a}$ have label a and require more edges to other vertices of label a. Consequently, from a greedy perspective, it does not make sense to add edges from this partition to other partitions. On the other hand, vertices of label a which require more edges to vertices of label b (i.e., elements of $\mathrm{V}_{a->b}$) pair perfectly with elements of $\mathrm{V}_{b->a}$.

The addition of the random edge allows the algorithm to continue in the event that adding edges within compatible partitions is itself insufficient.

IV. Experimental Evaluation

Despite the simplicity of the algorithm in §III, it is very effective. The predominant concern that it evokes is whether the quality of the solutions it produces degrades towards the trivial solution $\mathrm{K}_{|\mathrm{V}|}$ for every input graph. To assess the reality of this concern, we conducted an experimental investigation.

Experimental Setup

We implemented[2] the algorithm in C and ran it against first the toy example from §I, for which it produces the same solution we present, and then against a series of randomly generated graphs.

The dependent variable we evaluate is *occupancy change*: out of the $\binom{n}{2}$ possible edges in an undirected graph on n vertices, what percentage do we add to the graph G by running the algorithm. We generate fifteen synthetic, random graphs by selecting uniformly from pairs of vertices until we have the desired number of edges. We vary the number of nodes through $\{10, 20, 30, 40, 50\}$ and the initial occupancy through $\{25\%, 50\%, 75\%\}$. We fix the label set to be binary, since this is the most common type of label, and distribute the two labels evenly among the vertices.

The results of the experiments are illustrated in Figure 2. In Figure 2(a), we give the occupancy change in terms of varying α with the initial occupancy held fixed at 25%, and in Figure 2(b), we present the occupancy change in terms of varying initial occupancy with $\alpha = .1$. Note that the cases when the solution degrades to $\mathrm{K}_{|\mathrm{V}|}$ are those when the sum of the initial occupancy and the occupancy change is 1.

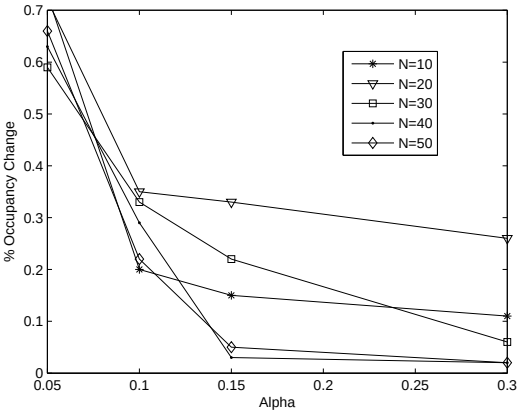
Discussion of Results

The results of the experiment in §IV are quite promising. Out of the sixty trials, only two instances, both with very small α, resulted in $\mathrm{K}_{|\mathrm{V}|}$ and the majority of trials produced graphs with $\leq 60\%$ occupancy. This is very important, because if the output is, indeed, $\mathrm{K}_{|\mathrm{V}|}$, then the analytical value of the network is lost. Rather, the closer that G' is to G, the better the analytical value is retained.

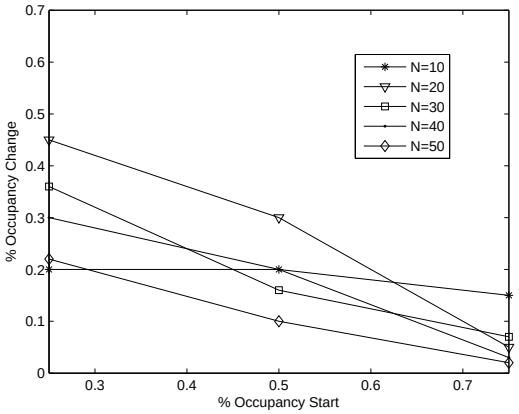
In terms of the variables that we investigated, no clear pattern emerges with respect to the number of vertices. As is clear from its definition, increases in α require adding fewer edges. More interestingly, the dropoff is quite abrupt at very small values of α and relatively minimal thereafter. This implies that requiring better anonymization than $\alpha = .1$ is perhaps unrealistic. The results with respect to initial occupancy reveal that for fixed n and α, one can expect similar total occupancy in the resultant graph, independent of initial occupancy.

It is especially encouraging that there are a number of trials that required occupancy changes of $\leq 10\%$. Not only does this imply that the algorithm did well on those instances, but it also implies that our proposed metric, α-nearness, can be quite readily achieved on certain input graphs.

[2]The implementation is available on the author's webspace, http://webhome.csc.uvic.ca/~schester/.



(a) Change in edge occupancy when starting at 25%, as α varies.



(b) Change in edge occupancy with $\alpha = .1$, as starting occupancy varies.

Fig. 2. Experimental Results

V. Future Work

There is substantial opportunity to extend this research. A first direction is in determining whether we can produce guarantees on the optimality of the solutions produced by this algorithm, as a function of the variables we investigated in this experimental study and of the skew in the overall distribution of labels.

We are also investigating new algorithmic ideas based on dynamic programming and on modifying our greedy algorithm to eliminate the non-determinism in a sensible (as opposed to arbitrary) manner. Our approach with respect to the latter involves exhaustively exploring the solution spaces on small graphs to see whether there is a systematically reliable route towards good solutions and whether those conclusions generalise to larger graphs.

In a slightly different direction, our experimental study here motivates researching which characteristics of social networks make them more amenable to easy transformation into α-near graphs, given that several trials resulted in small occupancy changes.

Given the success of these first ideas, we plan to conduct more rigorous experimental evaluation, using larger datasets, real-life networks, and power law graphs. As part of this, we would analyse the effect on structural properties like clustering coefficient of transforming a graph G into an α-near graph G'.

VI. Conclusion

In this paper we introduced the first approach to providing protection within social networks against attribute disclosure attacks. We defined α-*Nearness*, a measure of the susceptibility of a network to an attribute disclosure attack against an adversary who can identify a particular node's identity.

We also offered a greedy algorithm to convert a graph G into an α-near graph G', by augmenting its edge set, an approach quite prevalent in the social network anonymity literature. We demonstrated empirically on some synthetic social networks that the greedy algorithm performs rather well in terms of minimizing the number of edges added to the original graph G.

Acknowledgements

The authors would like to thank Alex Thomo for the stimulating discussions early in this research and Venkatesh Srinivasan for the same and for his assistance with this manuscript.

References

[1] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Anonymizing tables. In *ICDT 2005*, pages 246–258, 2005.

[2] J. Cheng, A. W.-C. Fu, and J. Liu. K-isomorphism: privacy preserving network publication against structural attacks. In *SIGMOD 2010*, pages 459–470, 2010.

[3] G. Cormode, D. Srivastava, T. Yu, and Q. Zhang. Anonymizing bipartite graph data using safe groupings. *VLDB J.*, 19(1):115–139, 2010.

[4] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *ICDE*, pages 106–115, 2007.

[5] K. Liu and E. Terzi. Towards identity anonymization on graphs. In *SIGMOD 2008*, pages 93–106, 2008.

[6] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. -diversity: Privacy beyond -anonymity. *TKDD*, 1(1), 2007.

[7] A. Meyerson and R. Williams. General k-anonymization is hard. In *Principles of Database Systems*, 2004.

[8] B. Thompson and D. Yao. The union-split algorithm and cluster-based anonymization of social networks. In *ASIACCS 2009*, pages 218–227, 2009.

[9] B. K. Tripathy and G. K. Panda. A new approach to manage security against neighborhood attacks in social networks. In *ASONAM*, pages 264–269, 2010.

[10] W. Wu, Y. Xiao, W. Wang, Z. He, and Z. Wang. k-symmetry model for identity anonymization in social networks. In *EDBT 2010*, pages 111–122, 2010.

[11] E. Zheleva and L. Getoor. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *WWW*, pages 531–540, 2009.

[12] B. Zhou and J. Pei. Preserving privacy in social networks against neighborhood attacks. In *ICDE 2008*, pages 506–515, 2008.