Prediction Promotes Privacy In Dynamic Social Networks

Smriti Bhagat
Rutgers University
smbhagat@cs.rutgers.edu

Graham Cormode, Balachander Krishnamurthy, Divesh Srivastava
AT&T Labs–Research
{*graham, bala, divesh*}*@research.att.com*

Abstract

Recent work on anonymizing online social networks (OSNs) has looked at privacy preserving techniques for publishing a single instance of the network. However, OSNs evolve and a single instance is inadequate for analyzing their evolution or performing longitudinal data analysis. We study the problem of repeatedly publishing OSN data as the network evolves while preserving privacy of users. Publishing multiple instances independently has privacy risks, since stitching the information together may allow an adversary to identify users. We provide methods to anonymize a dynamic network when new nodes and edges are added to the published network. These methods use link prediction algorithms to model the evolution. Using this predicted graph to perform group-based anonymization, the loss in privacy caused by new edges can be eliminated almost entirely. We propose metrics for privacy loss, and evaluate them for publishing multiple OSN instances.

1 Introduction

OSNs are a ubiquitous feature of modern life. A key feature of current OSNs, exemplified by Facebook, is that a user's detailed information is not visible without their explicit permission. This leaves interested parties—network researchers, sociologists, app designers—to scrape away at the edges. Full release of snapshots of the network would address this need. But the default settings are private for a reason: OSNs contain sensitive personal information of their users. Principled *anonymization* of OSN data allows sharing with 3rd-parties without revealing private information. After simplistic anonymization methods were shown to be vulnerable [3, 15] more sophisticated anonymizations have been proposed [20].

Prior work focused primarily on *static* networks: the dataset is a single instance of the network, represented as a graph, failing to capture the highly dynamic nature of social network data. We would like to repeatedly release anonymized snapshots reflecting the current state of the network. Ensuring sufficient privacy while keeping output relevant for its intended uses is more challenging in the dynamic case. Anonymizing each version of the network independently is easily shown to leak information by comparing the different versions of the data [21]. Instead, we ensure that subsequent releases are consistent with the initial release. Bad decisions made for an initial anonymization mean that subsequent releases may lead to an undesirable amount of information (measured in terms of probabilities) that can be extracted about the users in the data, and may require that some information is suppressed from the subsequent releases. Without knowing how the network will grow, how do we choose proper anonymizations early on so that the information that can be extracted about individuals from later releases is minimized?

We propose a solution based on *link prediction algorithms*, that use the current state of the network to predict future structure. The prediction is used to choose an anonymization which is expected to remain safe and useful for future releases. Existing prediction methods tend to over-predict edges, i.e., they suggest many more edges than actually arrive. Thus, we cannot treat the predicted edges equally to observed edges, and must define how to integrate predicted edges with anonymization algorithms. We present a variety of methods to select a subset of predicted edges to find a usable anonymization.

Outline and Contributions. Section 2 defines the anonymization problem for dynamic graphs, and describes four requirements of the output. Section 3 provides metrics for evaluating privacy preservation of anonymizations based on prediction. Section 4 discusses how different prediction models can be incorporated into our framework, and how the results of the prediction can be fine-tuned by adoption of conditions for anonymization. Section 5 presents experiments over temporal data representing social network activity from three different sources, and empirically evaluates privacy guarantees

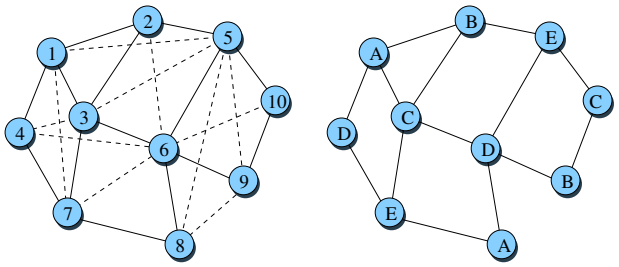
and utility resulting from our anonymization methods. Our study shows that with the correct choice of prediction method and anonymization properties, it is possible to provide useful data on dynamic social networks while retaining sufficient privacy. We conclude in Section 6 after reviewing related work.

2 Problem Definition

Graph Model. A time-varying social network can be represented with a graph $G_t = (V_t, E_t)$. Here V_t is the set of vertices that represent users (or, entities) U_t that are a part of the network at time t, and E_t is the set of all edges (interactions between users) created up to time t. Each user is associated with a set of attributes. Let $\mathcal{G} = \{G_1, G_2, \ldots, G_T\}$ be the sequence of T graphs representing the network observed at timesteps $t = 1, 2, \ldots, T$ respectively. We assume edges and nodes are only added to the graph, not deleted (our model can be extended to allow deletions, but we do not discuss this issue in this presentation) Thus, we have $V_t \subseteq V_{t+1}$ and $E_t \subseteq E_{t+1}$, i.e. the graph at time t represents the complete history of events recorded on the graph. New edges created between time t and $t+1$ form the set $E_{t+1} \backslash E_t$. Accordingly, any edge created at time $t+1$ is one of three kinds (i) "old-old": between nodes $v, w \in V_t$ (ii) "old-new": between node $v \in V_t$ and $w \in V_{t+1} \backslash V_t$ (iii) "new-new": between nodes $v, w \in V_{t+1} \backslash V_t$. Let T be the current timestamp so that all prior graphs G_i for $i \leq T$ are observed and known. The graph continues to evolve so that the graph G_{T+i} for $i > 0$ represents the (unknown) future state of the network.

Problem Statement. Given $\mathcal{G}$ as input, our objective at any time T is to publish an anonymized version of graph G_t as G'_t. The output graph G'_t should have the following properties based on privacy parameters p_n and p_e:
1. *entity privacy:* any $u \in U_t$ cannot be identified with a node in G'_t with probability $> p_n$.
2. *privacy of observed edges:* for any two entities $u_1, u_2 \in U_t$, where $t \leq T$, without background information it should not be possible to determine the existence of an edge between them with probability $> p_e$.
3. *privacy of future edges:* when G'_{T+i} is later published, it should not be possible to identify the presence of an edge between $u_1, u_2 \in U_{T+i}$ with probability $> p_e$.
4. *utility:* the anonymized graphs should be usable to obtain accurate answers to queries involving longitudinal analysis (e.g., how does the interaction between users from NJ change between two releases G'_t and G'_{t+1}).

Prior work in graph anonymization focused on publishing a single graph instance, with requirements similar to goals 1 and 2 above. When publishing information about network evolution, new events impact what has already been published, which motivates the third goal. If the anonymization has any value (i.e. it meets the fourth goal of utility), then we must balance the extra utility from publishing the new information with the potential threat to the privacy of the previously published data.



(a) Example graph G at time t. Solid lines are existing edges, dotted are predicted edges

(b) Full list anonymized G

Figure 1: Anonymization of a single snapshot of a graph

3 Understanding Dynamic Privacy

3.1 Anonymizing a single graph

The (full) list-based scheme for anonymizing a single graph was proposed in [4] (Section 6 identifies other methods). It masks the mapping between nodes in the graph V and entities U such that each $v \in V$ is associated with a *list* of possible labels $l(v) \subset U$. The original label of a node must appear within that node's list. Using the full list anonymization scheme, $|l(v)| \geq k$ and $|l(v)|$ nodes are assigned the same label list. The underlying graph structure is published, with a label list at each node instead of the user identifier. The lists can be generated by partitioning the nodes into *groups* of size k, so that each node in the group is given the same list, which consists of all (true) labels of nodes in the group.

If the links between nodes in a group, or between nodes in two groups, are *dense*, then an observer will conclude a high probability of certain edges. This contradicts the "privacy of observed edges" requirement, even while the privacy of entities requirement may be met. Hence, lists are generated by dividing nodes into g groups $S_1, S_2 \ldots S_g$ so that they satisfy a *Safety Condition*. This condition states (informally) that each node must interact with at most one node in any group and so ensures sparsity of interactions between nodes of any two groups. The resulting grouping guarantees the privacy of entities with parameter $p_n = 1/k$, and the privacy of observed edges with $p_e = 1/k$. Our focus in this paper is on maintaining this safety condition in the presence of arriving nodes and edges. For more context, and details of the strength it provides, see [4].

Example 1. *Figure 1(a) shows a sample snapshot of a graph at time t with node-set* $V_t = \{1, 2, \ldots 10\}$. *In Figure 1(b), the graph has been anonymized using the full*

list method: the nodes are partitioned into groups with $k = 2$ as $A = \{1, 8\}, B = \{2, 9\}, C = \{3, 10\}, D = \{4, 6\}$, and $E = \{5, 7\}$. The anonymized version satisfies the safety condition and shows the same graph with each node label replaced by the name of its group.

3.2 Naïve Approach to Dynamic Privacy

A natural first approach to anonymizing an evolving network $\mathcal{G} = \{G_1, G_2, \ldots, G_T\}$ is to individually anonymize each $G_t \in \mathcal{G}$ and publish each corresponding anonymized graph G'_t in turn as it is produced. For timestep t, G_t is grouped and each node $v \in V_t$ is assigned a list $l_t(v)$. However, since we treat each G_t separately, v's list is potentially different at different times t. Publishing the resulting set of lists is likely to reveal the identity of v. For instance, let the lists assigned to v at two timesteps be $l_1(v) = \{u_1, u_2, u_3\}$ and $l_2(v) = \{u_1, u_4, u_5\}$. The identity of v must be u_1, violating entity privacy.

This style of attack is possible because the partitioning of the nodes into groups varies across different releases of data, which allows linking attacks across the releases [9, 15]. A natural fix keeps the partitioning consistent over timesteps; i.e., partitioning of the vertices present in previous steps are kept the same. New vertices are grouped together, and the resulting grouping published using the lists implied. This approach (the "naïve method"), clearly has the same guarantees on node-privacy. We can modify existing algorithms to build groupings that respect the safety conditions for new-new edges and the new-old edges (as defined above). But if we fix the grouping, new edges between existing nodes that arrive at time t (i.e. the old-old edges) may violate the safety condition, and thus break the privacy guarantees. If too many edges arrive between a pair of existing groups, the (future) edge privacy is broken, as these edges can be identified with higher probability. We can then either publish these edges, with an associated elevated probability of revelation, or suppress them, which distorts the published graph and alters its basic properties. Our goal is to minimize the impact of these future edges compared to the naïve method, without knowing where they will arise.

3.3 Privacy Metrics

We partition the nodes into groups to minimize loss of edge privacy. As we must commit to a grouping before new edges arise, subsequent edges could raise the probability of re-identifying an edge. The first metric to quantify the effect of adding new edges on privacy is *Edge Identification* which measures the likelihood of identifying an interaction.

Definition 1. *Given a pair of groups of nodes at time t, $S_1 \subset V'_t$ and $S_2 \subset V'_t$, their* Edge Identification *$EI(S_1, S_2)$ is the ratio of the number of edges between the two groups to the maximum number of such edges:*

$$EI(S_1, S_2) = \frac{|(S_1 \times S_2) \cap E_t|}{|S_1| \cdot |S_2|}$$

Applying it to a grouping of G_t counts the pairs of groups S_1, S_2 with $EI(S_1, S_2) \geq \alpha$, for $1/k^2 \leq \alpha \leq 1$.

Note that EI is the probability an attacker can attach to a particular pair of users in groups S_1, S_2: over all the possibilities encoded by the grouping, only an $EI(S_1, S_2)$ fraction connect any pair of users with an edge. When the grouping satisfies the safety condition, $EI(S_1, S_2)$ is at most $1/k$ for any pair S_1, S_2, since each of the k nodes in a group may be connected to one node in the other group. Even when the safety condition is violated, this metric may still be less than $1/k$. Consider a node $v_1 \in S_1$ that interacts with nodes $w_1, w_2 \in S_2$, and these are the only links between the two groups. Here, the grouping does not meet the safety condition, but $EI(S_1, S_2) = 2/|S_1||S_2|$, which is no more than than $1/k$. Conversely though, if the number of interactions between a pair of groups of size k is more than k, it is not possible to meet the safety condition: some node in one group must connect to more than one node in the other. At the same time, the EI value must be greater than $1/k$. Hence, we can think of the safety condition as guaranteeing an EI of at most $1/k$, but not vice-versa. For full list anonymization to provide guarantees on the privacy of edges (observed or future), any pair of groups S_1, S_2, must have $EI(S_1, S_2) \leq p_e$.

The second privacy metric concerns the density of interactions between a given node and other groups.

Definition 2. *For a node $v \in V'_t$, the* Node-Group Density *with respect to a given group S is defined as*

$$NG(v, S) = |w \in S : (v, w) \in E|/|S|.$$

The overall Node-Group density of node v is defined as the maximum node group density of v over all groups i.e., $NG(v) = \max_S(NG(v, S))$. Applying the metric to a grouping of a graph counts the number of nodes v with $NG(v) \geq \beta$, for $1/k \leq \beta \leq 1$.

The condition $NG(v) > 1/k$ for any node v is a witness for a safety condition violation. A single violation has only limited local implications for edge privacy. Measuring the node-group density this way quantifies the extent to which the grouping is at risk. Even if $NG(v) = 1$, the maximum possible value, privacy of the node's interactions is not lost. But, if v is re-identified, then so are at least k interactions of v. Our goal is to anonymize each G_t so that EI and NG are small as new nodes and edges are added.

4 Dynamic Graph Anonymization

Our approach chooses a grouping of the nodes in V_1 and publishes G'_1 using this grouping. That is, we publish the

full graph structure along with a list of entities and their attributes, at each node. This is useful for answering a variety of queries on subpopulations within the network. We publish subsequent graphs by extending the grouping to the new nodes, i.e. $V_2 \backslash V_1, V_3 \backslash V_2$ and so on. We measure the quality of our initial choices by tracking the two privacy metrics for the published graphs as new edges arrive, i.e. $E_2 \backslash E_1, E_3 \backslash E_2$ etc. Our techniques to choose the initial grouping are based on predicting which new edges are most likely, via *link prediction*.

Link prediction has been heavily studied in the link analysis and mining literature [12]. We want to predict which links are likely to arise with the new nodes in $V_t \backslash V_{t-1}$ and hence choose how to group these new nodes so existing edges meet a safety condition, and future edges are unlikely to violate it. If the model predicts most future edges, the number of links between nodes in any pair of groups will remain small, and privacy guarantees should remain intact (future edge privacy).

More precisely, at time t we use link prediction model M to predict $\tilde{G}_t = (V_t, E_t \cup \tilde{E}_t)$, which includes a set of predicted edges, $\tilde{E}_t$. We use $\tilde{G}_t$ to generate the groups for $V_t \backslash V_{t-1}$, (nodes which have not yet been assigned to groups). In doing so, the grouping respects the safety condition for the combination of the previously observed edges (E_t) and additionally uses information about the predicted edges ($\tilde{E}_t$) to further guide the grouping process. For now, we assume M is given, and focus on using it for anonymization. We do not predict the arrival of new nodes from outside V_t: these can be grouped when they do arrive.

4.1 Grouping Conditions

The formal safety condition for a static graph in [4] is $\forall v_1 \in S_1, w_1, w_2 \in S_2, (v_1, w_1), (v_1, w_2) \in E \Rightarrow w_1 = w_2$. This ensures sparsity of interaction between nodes in any pair of groups, and also holds recursively if groups are partitioned. A natural approach to extend this to $\tilde{G}_t$ would be to apply the above condition with E replaced by $E_t \cup \tilde{E}_t$. However, prediction models tend to predict a very large number of edges. So there may be no safe grouping that satisfies the additional constraints introduced by these edges. Instead, we propose a subtly different condition.

Definition 3. *A grouping of nodes in graph $\tilde{G}_t$, satisfies the* Prediction-based Condition *if*

$\forall v_1 \in S_1, w_1, w_2 \in S_2 :$

$(v_1, w_1) \in E_t \wedge (v_1, w_2) \in (E_t \cup \tilde{E}_t) \Rightarrow w_1 = w_2$

i.e. there is no path of length two between two nodes in a group with at most one *predicted edge in the path.*

Example 2. *Figure 1(a) shows existing edges as solid lines and predicted edges as dotted lines. The safety condition (without prediction) allows nodes 1 and 10 to be in the same group, but the prediction-based condition does not. Under prediction grouping 4 and 10* can *be in the same group, as all 2 hop paths have two predicted edges.*

The prediction-based condition is stronger than the previous safety condition: the set of groupings satisfying the prediction condition is a subset of those satisfying safety. Next, we propose an alternate, more liberal density based condition which restricts the number of interactions between pairs of groups instead of at the node-level.

Definition 4. *A grouping satisfies the* Group Density *condition if, for every pair of groups (S_1, S_2), the density of links from $E_t \cup \tilde{E}_t$ between nodes in S_1 and nodes in S_2 is less than η, with $0 \leq \eta \leq 1$. This is achieved by insisting that $EI(S_1, S_2)$ is bounded by η.*

A grouping that violates the prediction-based condition may still be allowed by the group density condition. Here, EI is upper bounded by η, which can be smaller than the $1/k$ of the prediction-based condition. So enforcing a tighter group density condition may allow more edges to be added between a pair of groups before EI exceeds $1/k$.

A natural greedy heuristic finds a grouping which respects a given condition. The algorithm takes each new node $v \in (V_t \backslash V_{t-1})$ in turn, and inserts v into the first group of size less than k created at time t, so that the newly formed group satisfies the grouping condition (either the prediction-based or group density condition). If no such group can be found, then a new group of size k is created, initially containing v alone, which by definition satisfies either grouping condition. At the end of the procedure, any nodes in groups of size less than k can be merged into other groups (created at t) to form a few groups of size $k+1$. When using anonymized graphs, it is often preferable to have entities with similar attributes grouped together, provided the conditions are still met. This is achieved, for instance, by considering the nodes in an order which respects this clustering.

4.2 Modeling graph evolution

It is important to distinguish between the class of *generative* models, which create a synthetic instance with the aim to match the observed properties of real networks; and *predictive* models which take an existing instance of a network and predict which links are likely to occur in future. Here, we summarize the most relevant predictive models (see [12] for more background). In our setting, there is less concern over false-positives given by a link-prediction algorithm: the privacy provided is the same even if it turns out not to exist. Hence, we can adopt models which predict more edges than actually arrive.

Friend-of-a-friend (FOAF). For a given node, the FOAF model predicts edges to all nodes that are within

two hops in G_t. Formally this model predicts edges $\tilde{E}_t$ as, $\forall u, v, z \in V_t : (u, v), (v, z) \in E_t \Rightarrow (u, z) \in \tilde{E}_t$. The model treats each such edge equally likely to appear.

Common Neighbors (CN). The CN model assumes that when there are many common neighbors of two nodes they are more likely to become linked. It predicts the same links as FOAF, but attaches higher weight to those with more common neighbors. More precisely, it gives weights $\tilde{W}_t$ as, $\tilde{W}_t(u, z) = |v \in V_t : (u, v), (v, z) \in E_t|$. These weights are mostly small, typically 1 or 2.

Adamic-Adar (AA). The AA model [1] extends the previous models by arguing that all neighbors are not equal: a common neighbor with a low degree is more significant than one with a very high degree. The weight on an edge between u and z is:

$$\tilde{W}_t(u, z) = \sum_{v \in V_t:(u,v),(v,z) \in E_t} \frac{1}{\log(\deg(v))}$$

This predicts the same *set* of edges as FOAF and CN, but applies a finer gradation of weights.

Preferential Attachment (PA). The PA model is based on global properties of the graph. It assumes that links are more likely to nodes with high degree than to nodes with low degree [2]. For any two nodes $u, z \in V_t$ the model predicts an edge between them with weight $\tilde{W}_t(u, z) = \deg(u) \cdot \deg(z)$, where $\deg(u)$ is the degree of node u. PA implicitly predicts *all* possible edges, so thresholding is needed to make this meaningful.

Exactly which model is most suitable depends also on features of the social network itself. For instance, Facebook suggests friends drawn from the user's local structure, which tends to lead to a denser local graph (i.e. adding FOAF links). By contrast, networks such as Flickr and Twitter allow directed links that don't have to be reciprocated, and so promote popular users (i.e. a more PA-style growth model).

Several other features of nodes affect link creation, including, homophily (similarity of node attributes) and temporal state of a node to capture whether it is actively adding links or has reached a stable phase. These insights can be incorporated into the adopted model to adjust weights on predicted edges accordingly. Our anonymization approach is not strongly model dependent, so if new models are proposed to modify weights, they can naturally be incorporated into our framework. Clearly though, the resulting privacy and utility will depend on how well the predictions match reality. In our experimental evaluation we compare the relatively simple PA, FOAF, CN and AA models, and show that they are sufficient to show clear gains over no prediction.

4.3 Choosing from Predicted Links

The unfiltered FOAF model predicts a number of edges close to the sum of squares of the degrees, while PA effectively predicts an edge between *all* pairs of nodes. Finding a grouping that meets a condition (which enforces local sparsity) can be impossible on a dense graph, so it is imperative to select a subset of the predicted edges. We propose several alternatives:

Global Threshold (GT): pick the top τ most heavily weighted edges. This might predict many more edges incident on some nodes than others. In the PA model, almost all predicted edges will be incident on the highest degree nodes.

Local Threshold (LT): pick the τ' edges at each node which have the highest weight. This implicitly assumes a uniform level of activity across nodes, whereas in reality activity varies over the network.

Adaptive-Local Threshold (ALT): pick the top $f(\deg(v))$ edges, where $f(\deg(v))$ is a function of the the current degree $\deg(v)$ of node v. f can be e.g., linear or logarithmic in $\deg(v)$; it can also be set based on observed historic behavior of node growth.

	Facebook	FriendFeed	Flickr
Nodes at t=0	54K	99K	1.5M
Edges at t=0	887K	1.4M	19.5M
Nodes added	8.8K	100K	717K
Edge added	658K	2.48M	13.6M

Table 1: Summary of datasets

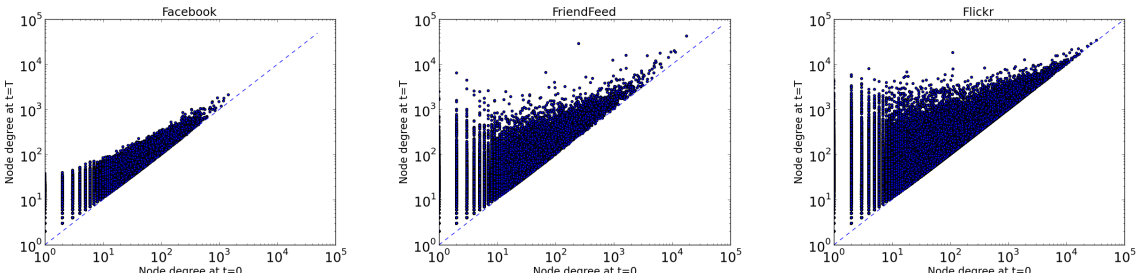
t	Timestamp	Nodes added	Edges added
0	Sept 14 '08	99K	1.4M
1	Jan 6 '09	33K	714K
2	Feb 26 '09	32K	1M
3	Mar 2 '09	2K	33K
4	Mar 27 '09	12K	257K
5	Apr 26 '09	20K	435K

Table 2: FriendFeed data at 6 timestamps

5 Experimental Analysis

We present results on the effectiveness of anonymization for data from Facebook, Flickr, and social network aggregator FriendFeed (summarized in Table 1). The data from the three networks was collected over 4-12 months. The first data collection is referred to as $t = 0$ and the last observed graph as $t = T$. Figure 2 shows log-log scale scatter plots for each dataset with node degrees at $t = 0$ on the x axis and at $t = T$ on the y axis, so a node lying on the $x = y$ line created no new links during this time period. Growth in Facebook is typically at most one order in magnitude, while in Flickr, some degrees increase by three orders of magnitude.

Facebook. The Facebook dataset is from the New Orleans region, collected between January 2008 and January 2009 [17]. Of the new links at $t = T$, 95% are



(a) Facebook (b) FriendFeed (c) Flickr

Figure 2: Log-log plots of node degree at t=0 versus degree at t=T for the three datasets

	Facebook	FFeed	Flickr
New edges at time T between existing (old) nodes	629K	803K	3.5M
# edges predicted by FOAF	42M	540M	900M
Sensitivity of FOAF predictions	53%	60%	60%
Sensitivity of PA predictions	4%	6.4%	0.4%
% of true positives *chosen* by ALT	98%	46%	8%
Naïve grouping for k=10	✓	×	✓
Fraction of celebrity nodes	0	0.01%	0.4%

Table 3: Summary of prediction and grouping

between nodes in V_0.

FriendFeed. The FriendFeed (FFeed) dataset has 99K nodes and 1.4M edges in September 2008, and within 7 months the network doubled in size [7]. The graph was collected at 6 timestamps (Table 2). The 12 highest degree nodes each have over 5000 links, and represent "celebrities".

Flickr. The Flickr datasets [14] were collected in November-December 2006, and February-May 2007. Flickr edges are directed with 62% of the links being reciprocated. 6284 nodes with degree over 1000 are considered celebrities. Of the remaining graph, 98% of the new links at $t = T$ are between existing nodes.

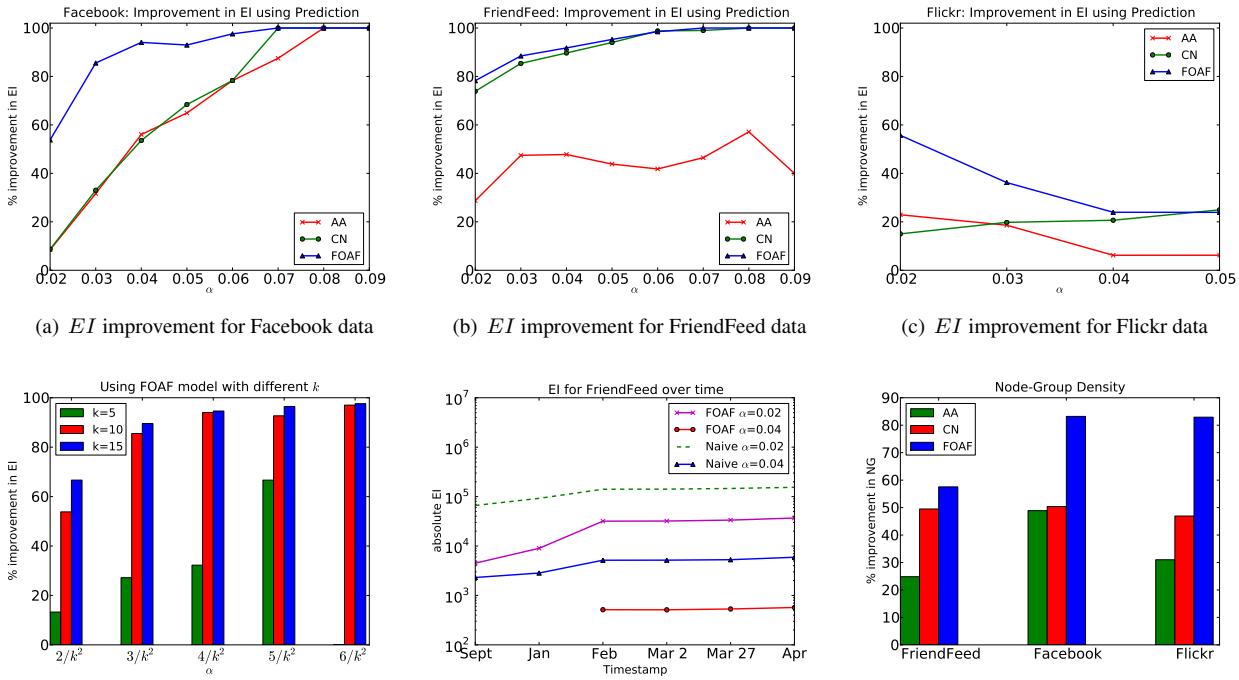
5.1 Anonymization

Link Prediction Models. We evaluate the prediction models on their *sensitivity*, which measures the fraction of new edges that are predicted by the model. Table 3 shows the sensitivity of prediction using the FOAF model as a percentage. These values strengthen our hypothesis that grouping with some knowledge of the anticipated edges helps to preserve privacy. Table 3 also shows the sensitivity of PA using ALT to choose a subset of edges. PA's sensitivity is sufficiently low that we focus on the local models for the rest of this analysis.

Choosing from Predicted Links. Table 3 shows that the number of edges predicted by FOAF is much larger than the new edges added by time T. If all predicted edges are included in the output of the model, the graph is no longer sparse enough to find a grouping satisfying the grouping conditions. We use thresholding to *choose* a subset of the predicted links based on their weights. A predicted edge is considered a "true positive" if it is observed in the graph at time T. We compare the threshold methods by computing the fraction of true positives chosen by each method for a fixed number of chosen edges, τ. We aim to choose a large τ value, provided that a grouping can be computed that respects the prediction-based condition.

For Facebook data with AA, we fix $\tau = 10$M, to allow sufficient over-prediction while still being able to group the graph. GT chooses 79% of the true positives. For LT, since there are 54K nodes, we pick the top $\tau' = 10\text{M}/54\text{K} = 185$ predicted edges by weight, for each node. This chooses 81% of the true positives. The threshold function for the ALT is chosen by first dividing the nodes into b bins ($b = 10$), based on their degree at $t = 0$. We set a local threshold by taking the distribution of final degrees of all nodes in each bin, and set the threshold $f(\deg(v))$ to be the 95th percentile of these. The parameter b and the percentile chosen may be tuned for each dataset based on node degree distribution. Of the thresholding methods, ALT best captures the true positives and picks 98% of them. For Facebook, all three threshold schemes work reasonably well. However, Figure 2 shows the number of edges added by nodes is not constant. Fixing one threshold for all nodes does not work as well as ALT. Since ALT outperforms the other methods for all datasets, we adopt this as the method of choice for all subsequent analysis. We show the percentage of true positives chosen using ALT in Table 3.

Partitioning Nodes into Groups. Depending on the sparsity G_t, there may be no grouping for a given k. For instance, if two nodes with high degrees are connected, then predicting all FOAF links means that no pair of their neighbors can be placed in the same group. This was the case in FriendFeed, where "celebrities" rendered it ungroupable for $k = 10$ even with the naïve grouping. Ta-



(a) EI improvement for Facebook data

(b) EI improvement for FriendFeed data

(c) EI improvement for Flickr data

(d) Different k values on Facebook data

(e) Multiple releases of FriendFeed data

(f) NG improvement for all datasets

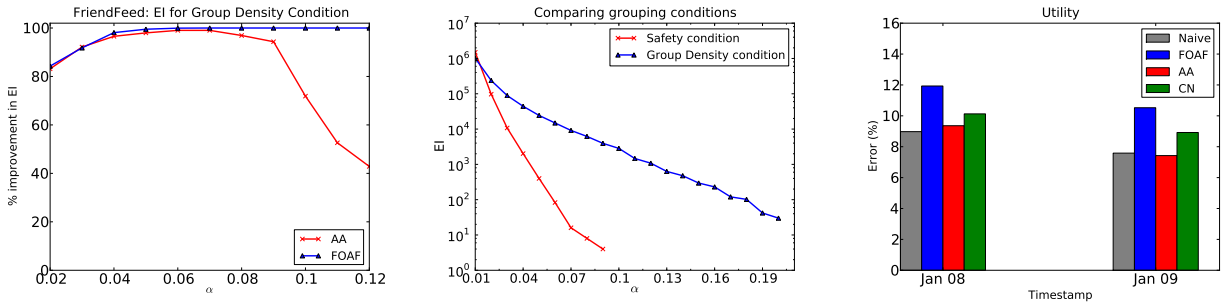
Figure 3: Evaluating EI and NG for different datasets

ble 3 shows the groupability of all datasets. In the rest of the section, we use $k = 10$ unless specified. For grouping FriendFeed, we ignore the 12 nodes with degree above 5000 to make the graph groupable with the naïve grouping. The graph is ungroupable using FOAF, but by modifying ALT to predict no links for the top 209 high degree nodes (degree > 1000), the graph becomes groupable with the prediction-based grouping condition. Celebrity nodes in Flickr are handled similarly.

5.2 Evaluating Privacy

We evaluate the privacy metrics EI and NG (Section 3.3). We anonymize the graph at $t = 0$ using no prediction, G'_t, and $\tilde{G}'_t$ using FOAF, CN or AA models. We compute the EI and NG counts for each anonymized graph at $t = T$. For each dataset, we compare the difference in EI (and NG) values for G'_t, and $\tilde{G}'_t$. In our experiments, out of the pairs of groups that interact, over 85% have at most one interaction between them, that is, $\alpha = 1/k^2$ for such pairs. This case is interesting as the EI values for $\alpha = 1/k^2$ are larger when using prediction, signifying that new edges are added between pairs of nodes that did not already interact. For $\alpha \geq 2/k^2$ the EI values are smaller for prediction-based grouping than the naïve grouping. This reduction signifies that fewer groups have more than 2 edges between them. Both cases are desirable for maintaining sparsity in interactions between groups, which translates to improved privacy for a grouping-based anonymization.

Edge Identification. Figure 3 shows the percentage change in EI when using prediction as compared with no prediction for different settings. Figures 3(a), 3(b) and 3(c) show the improvement in EI for Facebook, FriendFeed and Flickr respectively. Figure 3(a) shows an improvement of about 90% in EI for publishing $\tilde{G}'_t$ using FOAF over publishing $\tilde{G}'_t$ without prediction. The gain in using the fewer predicted edges from AA or CN is still significant: over 50% for $\alpha = 0.04$ and above. High values of α correspond to more edges between a pair of groups. The larger gain for higher values of α is expected, since use of predicted edges prevents groupings that have many edges between them. Figure 3(b) shows a similar trend with an average improvement of 93% over different α for the FriendFeed dataset anonymized using FOAF. For this dataset, the CN model (average gain 90%) seems to better explain the formation of the new links as compared with AA (average gain 43%), due to the greater number of true positives chosen. We observe that AA chooses 46% of correctly predicted links, while CN chooses about 70% of the true positives. The FOAF model for Flickr shows about a 36% improvement over the naïve grouping as seen in Figure 3(c). The prediction models do not exhibit large gains for the Flickr dataset. The size of this dataset is much larger and the average



(a) Grouping Density Condition (b) Comparing grouping conditions (c) Utility on Facebook dataset

Figure 4: Evaluating Density Condition and Utility for different datasets

degree of nodes (ignoring celebrities) is just 5. For such sparse graphs, the absolute EI values are low, leaving less room for improvement.

Group Size. Next we study the privacy parameter k, and its effect on EI. Figure 3(d) shows the results on Facebook dataset for $k = 5, 10$ and 15 using FOAF. For $k = 5$, there are more groups than with larger values of k. Hence, the likelihood of a new edge being added to a given pair of groups is smaller resulting in a smaller EI, and a relatively small benefit in using prediction. Conversely, for large k, the improvement in EI from predicting links and grouping accordingly is greater.

Multiple Releases. Figure 3(e) shows the absolute values of EI for multiple releases of the FriendFeed network. For a given α, the absolute EI values of naïve approach are significantly higher than with using prediction. The EI values increase before Feb, when many links are added to the graph. Before that, no pairs of groups have 4 or more links with prediction-based grouping. With fewer links added between existing nodes after Feb, the EI values change little.

Node-Group Density. The prediction based anonymization shows significant improvement over the naïve anonymization for the Node-Group Density metric as well. Figure 3(f) shows the improvement in the NG metric for graph $\tilde{G}'_t$ over G'_t. We present results for $\beta = 0.2$ and $k = 10$, which is equivalent to computing the number of pairs of groups where the safety condition is violated. That is, there exists at least one node in the groups that connects with two or more nodes in any other group. As with EI, the trend is that FOAF performs best, then CN followed by AA.

Group Density Condition. This condition was proposed to allow grouping a graph that is "ungroupable" using stricter safety and prediction-based conditions. Here, we consider all links including those that involve the high degree nodes. Figure 4(a) shows the improvement in EI obtained using FOAF and AA with group density condition on FriendFeed. This dataset is groupable using the group density condition, but not the prediction-based condition. Prediction results in a significant improvement in EI for various η, or density of links between groups. The results in Figure 4(a) are for a grouping that allows at most $\eta = 4/k^2$. The small improvement in EI for large α with AA is due to the small number of pairs of groups (S_1, S_2) that have $EI(S_1, S_2) = \alpha$. For the plot shown, the difference in the number of pairs of groups between $\tilde{G}'_t$ and G'_t for $\alpha \leq 0.09$ is in thousands, while that for $\alpha > 0.09$ is less than 5.

Figure 4(b) compares absolute EI values (in log scale) under the safety condition and the more relaxed group density condition for anonymization without prediction. We use Facebook, as it is groupable for both conditions so we can directly compare the privacy provided. For a given $\alpha \geq 0.02$, the number of groups with $EI \geq \alpha$ is much larger for the group density condition, leading to weaker privacy. With the safety condition there are no groups with more than 9 edges between them, which is not the case with the group density condition. Thus, the group density condition allows us to group graphs that are ungroupable under the safety condition, but with lower privacy bounds.

Utility. We analyze the accuracy of results of querying multiple releases of an anonymized graph. We compute the relative error in the query result with respect to the true result obtained on querying the unanonymized graph. Figure 4(c) shows the relative error for the query: "How many Facebook users aged 15-20 interact with users aged 20-30 at the start and end of the measurement period?" Our results show that there is negligible loss of utility when using AA compared to naïve grouping. The result is slightly better on graphs grouped using the AA and CN compared to FOAF. Similar trends were observed for a workload of 20 other queries.

6 Related Work

Graph Anonymization. There has been much research on data anonymization since k-anonymity on tabular data [16] and efforts in statistics [6, 8]. There are two styles of approach to the graph anonymization problem [20]. *Graph modification* adds and removes links so that the same structure appears multiple times in the graph. This is intended to defeat attacks which try to link to known structure in the graph[3, 15]. The duplicated structure can be local, such as degree [13] or immediate neighborhood [19], or global, such as full reachability from each node [21]. However, a graph may need a lot of modification before it meets these requirements.

Clustering-based methods aggregate edge or node information, so that there are many possible mappings from the clustering back to graphs, of which the original graph is promised to be one. List anonymization (Section 3) falls in this space; other variations are found in the work of Campan *et al.* [5] and Zheleva *et al.* [18]. Zou *et al.* [21] discuss the case of dynamic social network data (as an extension of their graph modification method), arguing that it can handle multiple releases by simply adding more dummy edges to mirror the newly arriving edges in k places around the graph. Use of prediction to better prepare for those edges that are likely to arrive is not discussed.

Evolution of Social Networks. An early large scale study on the evolution of two social networks Flickr and Yahoo! 360 [10] proposed a generative model based on preferential attachment biased by the activity state of the node. Empirical studies on Flickr [14], Facebook [17] and FriendFeed [7] analyze the growth of these networks over time. In [14], the authors showed that 80% of new links in Flickr are between nodes that are two-hops away.

7 Concluding Remarks

Our methods for anonymization of social network data permit multiple releases of data. The published data meets privacy requirements while remaining useful for further analysis. Link prediction gives significant benefits in maintaining the privacy in the data. It remains to extend this approach to other techniques for graph anonymization, such as other clustering and modification based methods. The clustering methods we studied give anonymity guarantees against adversaries with a limited background knowledge. However, it is also desirable to defend against more powerful adversaries, such as those that control large numbers of entities and manipulate their link structure.

Acknowledgements. We thank the authors of [14], [17] and [7] for graciously providing us with their data.

References

[1] L. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25:211–230, 2001.

[2] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74:47–97, 2002.

[3] L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore are thou R3579X? Anonymized social networks, hidden patterns and structural steganography. In *WWW*, 2007.

[4] S. Bhagat, G. Cormode, B. Krishnamurthy, and D. Srivastava. Class based graph anonymization for social network data. In *VLDB*, 2009.

[5] A. Campan and T. M. Truta. A clustering approach for data and structural anonymity in social networks. In *PinKDD*, 2008.

[6] G. Cormode and D. Srivastava. Anonymized data: generation, models, usage. In *ACM SIGMOD*, 2009.

[7] S. Garg, T. Gupta, N. Carlsson, and A. Mahanti. Evolution of an online social aggregation network: An empirical study. In *IMC*, 2009.

[8] J. Gehrke and A. Macahanavajjhala. Privacy in data publishing. In *IEEE Symposium on Security and Privacy*, 2009.

[9] M. Hay, D. Jensen, G. Miklau, D. Towsley, and P. Weis. Resisting structural re-identification in anonymized social networks. In *VLDB*, 2008.

[10] R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *ACM SIGKDD*, 2006.

[11] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. Microscopic evolution of social networks. In *ACM SIGKDD*, 2008.

[12] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *CIKM*, 2003.

[13] K. Liu and E. Terzi. Towards identity anonymization on graphs. In *ACM SIGMOD*, 2008.

[14] A. Mislove, H. S. Koppula, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Growth of the flickr social network. In *WOSN*, 2008.

[15] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *IEEE Symposium on Security and Privacy*, 2009.

[16] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. SRI-CSL-98-04, SRI Intl., 1998.

[17] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi. On the evolution of user interaction in facebook. In *WOSN*, 2009.

[18] E. Zheleva and L. Getoor. Preserving the privacy of sensitive relationships in graph data. In *PinKDD*, 2007.

[19] B. Zhou and J. Pei. Preserving privacy in social networks against neighborhood attacks. In *IEEE ICDE*, 2008.

[20] B. Zhou, J. Pei, and W. Luk. A brief survey on anonymization techniques for privacy preserving publishing of social network data. *SIGKDD Explorations*, 2008.

[21] L. Zou, L. Chen, and M. T. Ozsu. K-automorphism: A general framework for privacy preserving network publication. In *VLDB*, 2009.